

QTI 3 Player Item Reference



VERSION: DRAFT 0.2.10

AUTHOR: Paul Grudnitski, Amp-up.io

Table of Contents

Table of Contents	2
Change Log	9
Introduction	10
Supported Item Element Reference	11
Assessment Item Element	11
QtiAssessmentItem	11
qti-assessment-item Supported Attributes	11
Miscellaneous Item Elements	13
QtiAssessmentStimulusRef	13
qti-assessment-stimulus-ref Supported Attributes	13
QtiStylesheet	13
qti-stylesheet Supported Attributes	13
Variable Declaration Elements	15
QtiContextDeclaration	15
qti-context-declaration Supported Attributes	15
QtiResponseDeclaration	15
qti-response-declaration Supported Attributes	15
QtiOutcomeDeclaration	16
qti-outcome-declaration Supported Attributes	16
QtiTemplateDeclaration	17
qti-template-declaration Supported Attributes	17
QtiCorrectResponse	18
qti-correct-response Supported Attributes	18
QtiDefaultValue	18
qti-default-value Supported Attributes	18
QtiInterpolationTable	18
qti-interpolation-table Supported Attributes	18
QtiInterpolationTableEntry	19
qti-interpolation-table-entry Supported Attributes	19
QtiMapping	19
qti-mapping Supported Attributes	19
QtiMapEntry	20
qti-map-entry Supported Attributes	20
QtiMatchTable	20
qti-match-table Supported Attributes	20
QtiMatchTableEntry	21
qti-match-table-entry Supported Attributes	21

QtiValue	21
qti-value Supported Attributes	21
Body Elements	23
QtiItemBody	23
qti-item-body Supported Attributes	23
QtiChoiceInteraction	23
qti-choice-interaction Supported Attributes	24
QtiExtendedTextInteraction	25
qti-extended-text-interaction Supported Attributes	26
QtiGapMatchInteraction	27
qti-gap-match-interaction Supported Attributes	28
QtiGraphicGapMatchInteraction	30
qti-gap-match-interaction Supported Attributes	32
QtiHotspotInteraction	34
qti-hotspot-interaction Supported Attributes	35
QtiHottextInteraction	36
qti-hottext-interaction Supported Attributes	36
QtiInlineChoiceInteraction	37
qti-inline-choice-interaction Supported Attributes	38
QtiMatchInteraction	39
qti-match-interaction Supported Attributes	41
QtiSimpleMatchSet	42
qti-simple-match-set Supported Attributes	42
QtiOrderInteraction	43
qti-order-interaction Supported Attributes	44
QtiTextEntryInteraction	46
qti-text-entry-interaction Supported Attributes	46
QtiCustomInteraction	47
qti-custom-interaction Supported Attributes	47
amp:equation-editor Configuration	48
QtiEndAttemptInteraction	52
qti-end-attempt-interaction Supported Attributes	53
QtiPrintedVariable	56
qti-printed-variable Supported Attributes	56
QtiPrompt	57
qti-prompt Supported Attributes	57
QtiRubricBlock	57
qti-rubric-block Supported Attributes	57
QtiAssociableHotspot	58

qti-associable-hotspot Supported Attributes	58
QtiGap	59
qti-gap Supported Attributes	60
QtiGapImg	61
qti-gap-img Supported Attributes	61
QtiGapText	62
qti-gap-text Supported Attributes	62
QtiSimpleChoice	63
qti-simple-choice Supported Attributes	63
QtiSimpleAssociableChoice	63
qti-simple-associable-choice Supported Attributes	63
QtiHotspotChoice	64
qti-hotspot-choice Supported Attributes	64
QtiHottext	65
qti-hottext Supported Attributes	65
QtiInlineChoice	66
qti-inline-choice Supported Attributes	66
Miscellaneous Special HTML Elements	67
Object	67
MathML	67
Response and Template Processing Elements	68
QtiResponseProcessing	68
qti-response-processing Supported Attributes	68
QtiTemplateProcessing	69
qti-template-processing Supported Attributes	69
Response and Template Processing Rule Elements	70
QtiExitResponse	70
qti-exit-response Supported Attributes	70
QtiExitTemplate	70
qti-exit-template Supported Attributes	70
QtiLookupOutcomeValue	70
qti-lookup-outcome-value Supported Attributes	70
QtiResponseCondition	70
qti-response-condition Supported Attributes	71
QtiResponseSelf	71
qti-response-if Supported Attributes	71
QtiResponseElseIf	71
qti-response-else-if Supported Attributes	71
QtiResponseElse	71

qti-response-else Supported Attributes	72
QtiResponseProcessingFragment	72
qti-response-processing-fragment Supported Attributes	72
QtiSetCorrectResponse	72
qti-set-correct-response Supported Attributes	72
QtiSetDefaultValue	72
qti-set-default-value Supported Attributes	72
QtiSetOutcomeValue	73
qti-set-outcome-value Supported Attributes	73
QtiSetTemplateValue	73
qti-set-template-value Supported Attributes	73
QtiTemplateCondition	74
qti-template-condition Supported Attributes	74
QtiTemplateIf	74
qti-template-if Supported Attributes	74
QtiTemplateElseIf	74
qti-template-else-if Supported Attributes	74
QtiTemplateElse	75
qti-template-else Supported Attributes	75
QtiTemplateConstraint	75
qti-template-constraint Supported Attributes	75
Processing Expression Elements	76
QtiAnd	76
qti-and Supported Attributes	76
QtiAnyN	76
qti-any-n Supported Attributes	76
QtiBaseValue	77
qti-base-value Supported Attributes	77
QtiCorrect	77
qti-correct Supported Attributes	77
QtiCustomOperator	77
amp:math.stringToFloat	78
amp:ee.isEquivalent	78
QtiDefault	79
qti-default Supported Attributes	79
QtiDelete	79
qti-delete Supported Attributes	79
QtiDivide	79
qti-divide Supported Attributes	80

QtiEqual	80
qti-equal Supported Attributes	80
QtiEqualRounded	81
qti-equal-rounded Supported Attributes	81
QtiFieldValue	82
qti-field-value Supported Attributes	82
QtiGcd	82
qti-gcd Supported Attributes	82
QtiGt	83
qti-gt Supported Attributes	83
QtiGte	83
qti-gte Supported Attributes	83
QtiIndex	83
qti-index Supported Attributes	83
QtiIntegerDivide	84
qti-integer-divide Supported Attributes	84
QtiIntegerModulus	84
qti-integer-modulus Supported Attributes	84
QtiIntegerToFloat	84
qti-integer-to-float Supported Attributes	84
QtiIsNull	85
qti-is-null Supported Attributes	85
QtiLcm	85
qti-lcm Supported Attributes	85
QtiLt	85
qti-lt Supported Attributes	85
QtiLte	86
qti-lte Supported Attributes	86
QtiMapResponse	86
qti-map-response Supported Attributes	86
QtiMatch	86
qti-match Supported Attributes	87
QtiMathConstant	87
qti-math-constant Supported Attributes	87
QtiMathOperator	87
qti-math-operator Supported Attributes	88
QtiMax	88
qti-max Supported Attributes	89
QtiMember	89

qti-member Supported Attributes	89
QtiMin	89
qti-min Supported Attributes	89
QtiMultiple	90
qti-multiple Supported Attributes	90
QtiNot	90
qti-not Supported Attributes	90
QtiOr	90
qti-or Supported Attributes	90
QtiOrdered	91
qti-ordered Supported Attributes	91
QtiPatternMatch	91
qti-pattern-match Supported Attributes	91
QtiPower	92
qti-power Supported Attributes	92
QtiProduct	92
qti-product Supported Attributes	92
QtiRandom	92
qti-random Supported Attributes	92
QtiRandomFloat	93
qti-random-float Supported Attributes	93
QtiRandomInteger	93
qti-random-integer Supported Attributes	93
QtiRepeat	94
qti-random-float Supported Attributes	94
QtiRound	94
qti-round Supported Attributes	95
QtiRoundTo	95
qti-round-to Supported Attributes	95
QtiStatsOperator	96
qti-stats-operator Supported Attributes	96
QtiStringMatch	97
qti-string-match Supported Attributes	97
QtiSubstring	97
qti-substring Supported Attributes	97
QtiSubtract	98
qti-subtract Supported Attributes	98
QtiSum	98
qti-sum Supported Attributes	98

QtiTruncate	98
qti-truncate Supported Attributes	99
QtiVariable	99
qti-variable Supported Attributes	99
Template and Feedback Elements	100
QtiTemplateBlock	100
qti-template-block Supported Attributes	100
QtiTemplateInline	101
qti-template-inline Supported Attributes	101
QtiFeedbackBlock	102
qti-feedback-block Supported Attributes	102
QtiFeedbackInline	103
qti-feedback-inline Supported Attributes	103
QtiModalFeedback	104
qti-modal-feedback Supported Attributes	104
QtiContentBody	105
qti-content-body Supported Attributes	105
Catalog Elements	106
QtiCatalogInfo	106
qti-catalog-info Supported Attributes	106
QtiCatalog	106
qti-catalog Supported Attributes	106
QtiCard	106
qti-card Supported Attributes	107
QtiCardEntry	107
qti-card-entry Supported Attributes	108
QtiHtmlContent	108
qti-html-content Supported Attributes	108
QtiFileRef	108
qti-file-ref Supported Attributes	108
Example of QtiCatalogInfo	110
Annotated Item XML Examples	111
TDX - baseline	111
TDX - showex	117
TDX - solve	126

Change Log

Version	Date	Description	Author
0.2.2	Feb 24, 2022	Add QtiExtendedTextInteraction support	Paul Grudnitski
0.2.3	Mar 15, 2022	Add Shared CSS to Annotated Examples	Paul Grudnitski
0.2.4	Apr 4, 2022	Add QtiCatalogInfo support	Paul Grudnitski
0.2.5	Apr 7, 2022	Add QtiRubricBlock support	Paul Grudnitski
0.2.6	May 31, 2022	Add QtiOrderInteraction support	Paul Grudnitski
0.2.7	Jan 4, 2023	Add QtiStylesheet support Add QtiAssessmentStimulusRef support	Paul Grudnitski
0.2.8	Jan 23, 2023	Add QtiHottextInteraction support Add QtiHotspotInteraction support	Paul Grudnitski
0.2.9	Sep 16, 2023	Add QtiMatchInteraction support	Paul Grudnitski
0.2.10	Sep 17, 2023	Add QtiGapMatchInteraction support Add QtiGraphicGapMatchInteraction support	Paul Grudnitski

Introduction

The QTI 3 Player SDK (“QTI 3 Player”) is a 100% JavaScript component that aims to encapsulate the best practices and behaviors of the IMS Global **QTI 3 Assessment Item** (not Assessment Test) specification which can be found here:

[IMS Global QTI v3 Best Practices and Implementation Guide](#)

This document is a reference of the QTI 3 Item XML elements and attributes supported by the QTI 3 Player. By inspecting this reference, a conforming QTI 3 authoring or exporting system can construct a QTI 3 Item XML solution that will “play” authentically and reliably (according to the Best Practices Implementation Guide) in the QTI 3 Player component.

THIS REFERENCE DOCUMENT (“REFERENCE”) IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER. ANY USE OF THIS REFERENCE SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER AMP-UP.IO, NOR ANY OF ITS MEMBERS, EMPLOYEES OR CONTRIBUTORS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS REFERENCE.

Supported Item Element Reference

Assessment Item Element

QtiAssessmentItem

An assessment item encompasses the information that is presented to a candidate and information about how to score the item. Scoring takes place when candidate responses are transformed into outcomes by response processing rules.

qti-assessment-item Supported Attributes

Name	Usage	Value	Default
identifier	Required	string Must be a valid QTI identifier string.	
title	Optional	string "My Awesome Item"	
time-dependent	Required	boolean Denotes if the Item must be answered within some defined time limit.	
adaptive	Optional	boolean Denotes the adaptive nature of the Item. Items are classified as either Adaptive or Non-adaptive.	false
xml:lang	Optional	string Used to inform presentation and/or assistive technology.	en

The qti-assessment-item element must contain an "identifier" attribute with a non-null string to identify the item. It is used to identify the item as a unique entity within the context of the exchange of assessment content.

An Item may contain the following child elements. The elements with an asterisk (*) are NOT supported by the QTI 3 Player component. Note: the development roadmap does include support for asterisked elements.

Context Declaration
Response Declaration
Outcome Declaration
Template Declaration
Template Processing
Stylesheet

AssessmentStimulusRef
Companion Materials *
Item Body
Catalog Resources
Response Processing
Modal Feedback

Miscellaneous Item Elements

QtiAssessmentStimulusRef

AssessmentStimulusRef (qti-assessment-stimulus-ref) elements enable references to external “qti-assessment-stimulus” instances. The stimulus must be contained within its own instance.

According to Content Packaging rules, each AssessmentStimulusRef identifierref attribute must resolve to some AssessmentStimulus object that has been associated with the item. Qti 3 Player *does not support runtime injection of stimuli* referenced by AssessmentStimulusRef’s. Instead, injection occurs during Content Package import processing.

qti-assessment-stimulus-ref Supported Attributes

Name	Usage	Value	Default
identifierref	Required	“Stimulus1” This is a QTI identifier which must match the identifier of a content package resource of type “imsqti_stimulus_xm1v3p0” in the imsmanifest.xml of the content package.	
href	Required	uri	
title	Optional	string	

QtiStylesheet

Stylesheet (qti-stylesheet) elements are the set of external style sheets that are associated with the Item or Stimulus. The order of definition is significant.

QTI 3 Player injects the contents of each qti-stylesheet element into the DOM as <style> elements within the encapsulating element; e.g., qti-assessment-item, qti-assessment-stimulus, qti-rubric-block, etc.

qti-stylesheet Supported Attributes

Name	Usage	Value	Default
href	Required	uri	

		This must resolve to a CSS stylesheet. The URI may be a data URI.	
type	Optional	A mime type; e.g., "text/css"	

Variable Declaration Elements

QtiContextDeclaration

Context declarations (qti-context-declaration) declare item variables that are global in scope. Context variables are initialized at the start of an item session. They can have their values set and retrieved during templateProcessing or responseProcessing.

qti-context-declaration Supported Attributes

Name	Usage	Value	Default
identifier	Required	"MY_CONTEXT_VARIABLE"	
base-type	Optional	Enumerated value set of: { boolean float identifier integer string }	
cardinality	Required	Enumerated value set of: { multiple ordered record single }	

The QTI 3 information model has one built-in context variable with an identifier of QTI_CONTEXT and cardinality of record. QTI_CONTEXT has the following built-in fields:

candidateIdentifier: a string used to map to a candidate object.

testIdentifier: a string used to map to a test object.

environmentIdentifier: a string used to map to an environment runtime in which template processing or response processing is being evaluated.

QtiResponseDeclaration

Response variables are declared by response declarations (qti-response-declaration). Interactions in the itemBody must be bound to one of these declared variables. It may indicate what the correct response is, or the partially correct response(s). The base-type and cardinality defined for a response variable must match the types and cardinality required by the interactions bound to that response variable.

qti-response-declaration Supported Attributes

Name	Usage	Value	Default
identifier	Required	"RESPONSE_1"	

base-type	Optional	Enumerated value set of: { boolean float identifier integer string }	
cardinality	Required	Enumerated value set of: { multiple ordered record single }	

QtiOutcomeDeclaration

Outcome variables are declared by outcome declarations (qti-outcome-declaration). Their value is set either from a default given in the declaration itself by a responseRule during responseProcessing. Items that declare a numeric outcome variable representing the candidate's overall performance on the item should use the outcome name "SCORE" for the variable. SCORE needs to be a float. Items that declare a maximum score (in multiple response choice interactions, for example) should do so by declaring the "MAXSCORE" variable. MAXSCORE needs to be a float. Items that want to make the fact that the candidate scored above a predefined threshold available as a variable should use the "PASSED" variable. PASSED needs to be a boolean. At runtime, outcome variables are instantiated as part of an item session. Their values may be initialized with a default value and/or set during responseProcessing. If no default value is given in the declaration then the outcome variable is initialized to NULL unless the outcome is of a numeric type (integer or float) in which case it is initialized to 0. Declared outcomes with numeric types should indicate their range of possible values using normal-maximum and normal-minimum, especially if this range differs from [0,1].

qti-outcome-declaration Supported Attributes

Name	Usage	Value	Default
identifier	Required	"MY_CONTEXT_VARIABLE"	
base-type	Optional	Enumerated value set of: { boolean float identifier integer string }	
cardinality	Required	Enumerated value set of: { multiple ordered record single }	
normal-maximum	Optional	float	
normal-minimum	Optional	float	
interpretation	Optional	string	

external-scored	Optional	Enumerated value set of: { externalMachine human }	
variable-identifier-ref	Optional	string	

QtiTemplateDeclaration

Template declarations (qti-template-declaration) declare item variables that are to be used specifically for the purposes of cloning items. They can have their value set only during templateProcessing. They are referred to within the itemBody in order to individualize the clone and possibly also within the responseProcessing rules if the cloning process affects the way the item is scored. Template variables are initialized at the start of an item session. Template processing may update template variables during each iteration of template processing. If a templateConstraint=true is encountered then Template variables are reinitialized.

qti-template-declaration Supported Attributes

Name	Usage	Value	Default
identifier	Required	"MY_VARIABLE"	
base-type	Optional	Enumerated value set of: { boolean float identifier integer string }	
cardinality	Required	Enumerated value set of: { multiple ordered record single }	
param-variable	Optional	boolean This characteristic determines whether or not the template variable's value should be substituted for object parameter values that match its name. Resolves to a boolean value.	false
math-variable	Optional	boolean This characteristic determines whether or not the template variable's value should be substituted for identifiers that match its name in MathML expressions.	false

QtiCorrectResponse

A response declaration may assign an optional CorrectResponse (qti-correct-response) value. This value may indicate the only possible value of the response variable to be considered correct or merely just a correct value. For responses that are being measured against a more complex scale than correct/incorrect this value should be set to the (or an) optimal value.

Finally, for responses for which no such optimal value is defined the correctResponse must be omitted.

qti-correct-response Supported Attributes

Name	Usage	Value	Default
No Supported Attributes			

QtiDefaultValue

A DefaultValue (qti-default-value) is an optional default value for a variable. The point at which a variable is set to its default value varies depending on the type of item variable.

qti-default-value Supported Attributes

Name	Usage	Value	Default
No Supported Attributes			

QtiInterpolationTable

An InterpolationTable (qti-interpolation-table) transforms a source float (or integer) by finding the first interpolationTableEntry with a sourceValue that is less than or equal to (subject to includeBoundary) the source value. For example, an interpolation table can be used to map a raw numeric score onto an identifier representing a grade. It may also be used to implement numeric transformations such as those from a simple raw score to a value on a calibrated scale.

qti-interpolation-table Supported Attributes

Name	Usage	Value	Default
default-value	Optional	string The default outcome value to be used when no matching table entry is found. If omitted, the null value is used.	null

QtiInterpolationTableEntry

Provides an interpolation table entry (qti-interpolation-table-entry) in the associated interpolation table.

qti-interpolation-table-entry Supported Attributes

Name	Usage	Value	Default
source-value	Required	float The lower bound for the source value to match this entry.	
include-boundary	Optional	boolean Determines if an exact match of sourceValue matches this entry. If 'true', the default, then an exact match of the value is considered a match of this entry.	false
target-value	Required	string The target value that is used to set the outcome when a match is found.	

QtiMapping

A Mapping element (qti-mapping) is used to create a mapping (a Map) from a source set of any baseType (except file and duration) to a single float. Note that mappings from values of base type float should be avoided due to the difficulty of matching floating point values, see the match operator for more details. When mapping containers the result is the sum of the mapped values from the target set. See the qti-map-response expression for further details.

Note: A qti-mapping must contain at least one qti-map-entry.

qti-mapping Supported Attributes

Name	Usage	Value	Default
lower-bound	Optional	float The lower bound for the result of mapping a container. If unspecified there is no lower bound.	
upper-bound	Optional	float	

		The upper bound for the result of mapping a container. If unspecified there is no upper bound.	
default-value	Optional	float The default value from the target set to be used when no explicit mapping for a source value is given.	

QtiMapEntry

The map is defined by a set of mapEntries (qti-map-entry), each of which maps a single value from a source set onto a single float.

qti-map-entry Supported Attributes

Name	Usage	Value	Default
map-key	Required	string The source value for the mapEntry.	
mapped-value	Required	float The mapped value	
case-sensitive	Optional	boolean Used to control whether or not a mapEntry string is matched case sensitively. The default value is false.	false

QtiMatchTable

A MatchTable element (qti-match-table) transforms a source integer by finding the first matchTableEntry with an exact match to the source.

qti-match-table Supported Attributes

Name	Usage	Value	Default
default-value	Optional	string	null

		The default outcome value to be used when no matching table entry is found. If omitted, the null value is used.	
--	--	---	--

QtiMatchTableEntry

Provides a match table entry (qti-match-table-entry) in the associated match table.

qti-match-table-entry Supported Attributes

Name	Usage	Value	Default
source-value	Required	integer The lower bound for the source value to match this entry. Must be an integer.	
target-value	Required	Enumerated value set of: { boolean float identifier integer string } The target value that is used to set the outcome when a match is found.	

QtiValue

The Value element (qti-value) represents a single value of any base-type in variable declarations and result reports. The base-type is defined by the 'base-type' attribute of the declaration except in the case of variables with record cardinality. Parents can be qti-correct-response or qti-default-value.

qti-value Supported Attributes

Name	Usage	Value	Default
base-type	Optional	Enumerated value set of: { boolean float identifier integer string }	
field-identifier	Optional	identifier This is only used for specifying the field identifier for a value that forms part of a record.	

Body Elements

QtiItemBody

The ItemBody for the item. The ItemBody (qti-item-body) contains the text, graphics, media objects and interactions that describe the item's content and information about how it is structured. The body is presented to the candidate when the associated item session is in the interacting state.

qti-item-body Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiChoiceInteraction

The Choice Interaction (qti-choice-interaction) is a block interaction which presents a collection of choices to the candidate. The candidate's task is to select one or more of the choices, up to a maximum of max-choices. The interaction is always initialized with no choices selected.

Depending on the cardinality of the response variable declaration that is bound to a Choice Interaction, the QTI 3 Player presents the choices with (single cardinality) Radio Group UX or (multiple cardinality) Choice Group UX.

The QTI 3 Player supports ALL QTI 3 Choice Interaction shared vocabularies. In the absence of any explicit shared vocabulary, the QTI 3 Player displays the choices with a vertical orientation, with radio or checkbox controls, and with labels of upper alpha ("A", "B", "C", etc.) and suffix of period (".")

Example of Default (no shared vocabulary) Single Cardinality Display

What does it say?

☒ A. You must stay with your luggage at all times.
 ☐ B. Do not let someone else look after your luggage.
 ☐ C. Remember your luggage when you leave.

Example of Default (no shared vocabulary) Multiple Cardinality Display

Select 0 to 6 SimpleChoices below and trigger Response Processing by ending the attempt.

☐ A. Mapped Value = 1.0

☒ B. Mapped Value = 2.0

☐ C. Mapped Value = 5.0

☐ D. Mapped Value = -1.0

☐ E. Mapped Value = -2.0

☐ F. Mapped Value = -5.0

qti-choice-interaction Supported Attributes

Name	Usage	Value	Default
response-identifier	Required	identifier Must resolve to a response variable identifier.	
class	Optional	string Use class to specify shared vocabulary.	
min-choices	Optional	integer	0
max-choices	Optional	integer	1
shuffle	Optional	boolean If the shuffle characteristic is true then the QTI 3 Player randomizes the order in which the choices are initially presented, subject to the value of the fixed attribute of each choice.	false
data-max-selections-message	Optional	string	"You are permitted a maximum of {max-choices} choices"

			for this question. Please unselect one of your choices before making another choice.”
data-min-selections-message	Optional	string	“You must make at least {min-choices} choices for this question.”

For a complete description of the QTI 3 Shared Vocabularies for the Choice Interaction, please refer to the QTI 3 Best Practices Implementation Guide. You may also refer to this [QTI 3 Player TestRunner Test \(Single cardinality\)](#) and [QTI 3 Player TestRunner Test \(Multiple cardinality\)](#), which demonstrate implementation and usage of all QTI 3 Choice Interaction (single and multiple cardinality) conformance items..

QtiExtendedTextInteraction

The Extended Text Interaction (qti-extended-text-interaction) is a block interaction that allows the candidate to enter an extended amount of text.

The response variable declaration that is bound to an Extended Text Interaction must be of single cardinality, with a string base-type.

The QTI 3 Player supports all Extended Text Interaction shared vocabulary which provides more precision when describing:

1. Interaction height
2. Character counter behavior
3. Plain or Rich Text options
4. A custom pattern-mask message

In accordance with responsive design best practices, QTI 3 Player renders an Extended Text Interaction with a width equal to 100% of the interaction's encapsulating container. In the absence of any explicit shared vocabulary, the QTI 3 Player displays a plain textarea with a width of 100% of its container and a height of 5.15rem (tall enough for 3 lines).

Example of Default (no shared vocabulary) Display

The quick brown fox
jumps over the lazy dog.

Example format="xhtml" Display (rich text editor)

↺ ↻ **B** *I* U “ ” ☰ ☷

 The **quick** brown fox
jumps over the
lazy dog.

qti-extended-text-interaction Supported Attributes

Name	Usage	Value	Default
response-identifier	Required	identifier Must resolve to a response variable identifier.	
class	Optional	string Use class to specify shared vocabulary.	
format	Optional	string Use format to specify the control that QTI 3 Player renders. Enumerated value set of: { “plain” “xhtml” }	“plain”
pattern-mask	Optional	string If given, the pattern mask specifies a regular expression that the candidate's response must match in order to be considered	

		valid. The regular expression language used is defined here: https://www.w3.org/TR/xmlschema-2/#regexs Note: This is an XML Regex expression.	
placeholder-text	Optional	string By default, QTI 3 Player extended text interactions display no placeholder text. Use this attribute to override the default behavior.	
data-patternmask-message	Optional	string Override QTI 3 Player's default pattern-mask error message.	"Invalid Input"

For a complete description of the QTI 3 Shared Vocabularies for the Extended Text Interaction, please refer to the QTI 3 Best Practices Implementation Guide. You may also refer to this [QTI 3 Player TestRunner Test](#).

QtiGapMatchInteraction

The GapMatch Interaction (qti-gap-match-interaction) is a block interaction that contains a number of gaps that the candidate can fill from an associated set of choices. The candidate must be able to review the content with the gaps filled in context, as indicated by their choices.

The GapMatch Interaction must be bound to a response variable with base-type 'directedPair' and either single or multiple cardinality, depending on the number of gaps. The choices represent the source of the pairing and gaps the targets. Each gap can have at most one choice associated with it.

The QTI 3 Player supports ALL QTI 3 Gap Match Interaction shared vocabularies. In the absence of any explicit shared vocabulary, the QTI 3 Player displays the set of choices above the content containing the gaps.

Example of Default (no shared vocabulary) Display

Identify the missing words in this famous quote from Shakespeare's Richard III.

winter

spring

summer

autumn

Now is the of our discontent

Made glorious by this sun of York;

And all the clouds that lour'd upon our house

In the deep bosom of the ocean buried.

Example of Display with “qti-choices-left” shared vocabulary

winter

spring

summer

autumn

Now is the of our discontent

Made glorious by this sun of York;

And all the clouds that lour'd upon our house

In the deep bosom of the ocean buried.

Example of Display with “qti-choices-left” shared vocabulary and data-choices-container-width=“100”

winter

spring

summer

autumn

Now is the of our discontent

Made glorious by this sun of York;

And all the clouds that lour'd upon our house

In the deep bosom of the ocean buried.

qti-gap-match-interaction Supported Attributes

Name	Usage	Value	Default
response-identifier	Required	identifier Must resolve to a response variable of single or multiple	

		cardinality and base-type of directedPair	
class	Optional	string Use class to specify shared vocabulary.	
min-associations	Optional	integer The minimum number of associations that the candidate is required to make to form a valid response. If min-associations is 0 then the candidate is not required to make any associations. min-associations must be less than or equal to the limit imposed by max-associations.	0 (no constraint)
max-associations	Optional	integer The maximum number of associations that the candidate is allowed to make. If max-associations is 0 then there is no restriction. If max-associations is greater than 1 (or 0) then the interaction must be bound to a response variable with multiple cardinality.	1
shuffle	Optional	boolean If the shuffle characteristic is true then the QTI 3 Player randomizes the order in which the choices are initially presented, subject to the value of the fixed attribute of each choice.	false
data-max-selections-message	Optional	string	"You may make a maximum of {max-associations}"

			matches for this question.”
data-min-selections-message	Optional	string	“You must make at least {min-associations} matches for this question.”
data-choices-container-width	Optional	integer The width in pixels of the gap choices container. Example: data-choices-container-width=“100”	

For a complete description of the QTI 3 Shared Vocabularies for the GapMatch Interaction, please refer to the QTI 3 Best Practices Implementation Guide. You may also refer to this [QTI 3 Player TestRunner Test](#) which demonstrates implementation and usage of all QTI 3 Gap Match Interaction conformance items.

QtiGraphicGapMatchInteraction

The GraphicGapMatch Interaction (qti-graphic-gap-match-interaction) is a graphical interaction with a set of gaps that are defined as areas (hotspots) of the graphic image and an additional set of gap choices that are defined outside the image.

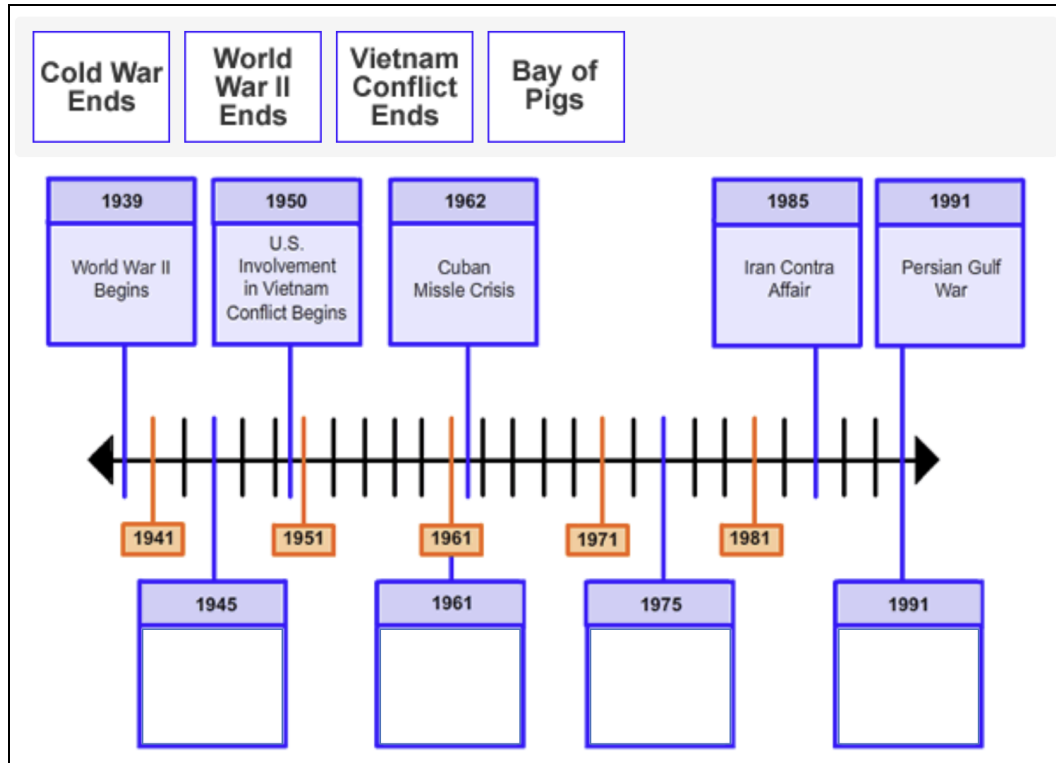
The candidate must associate the gap choices with the gaps in the image and be able to review the image with the gaps filled in context, as indicated by their choices. Care should be taken when designing these interactions to ensure that the gaps in the image are a suitable size to receive the required gap choices. It must be clear to the candidate which hotspot each choice has been associated with. When associated, choices must appear wholly inside the gaps if at all possible and, where overlaps are required, should not hide each other completely.

If the candidate indicates the association by positioning the choice over the gap (e.g. drag and drop) the system should 'snap' it to the nearest position that satisfies these requirements. The GraphicGapMatch Interaction must be bound to a response variable with base-type 'directedPair' and multiple cardinality.

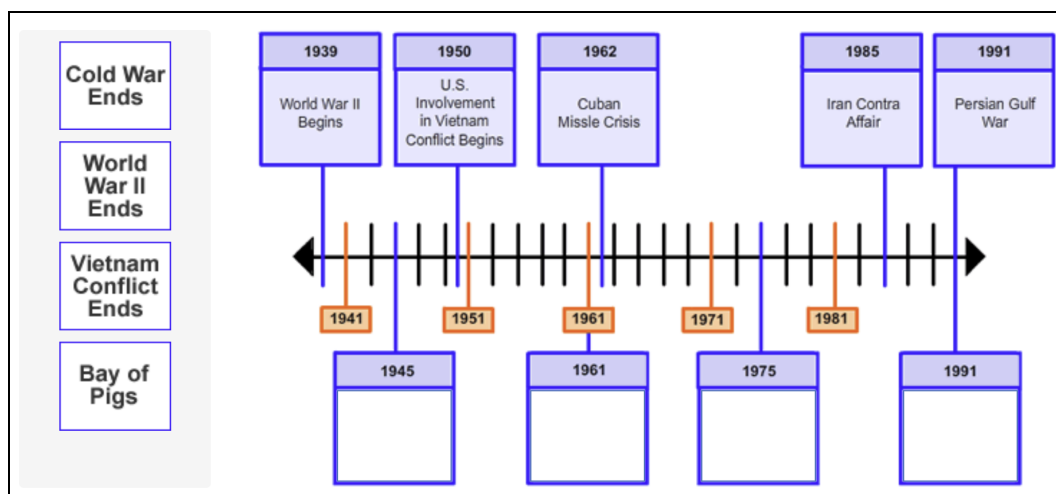
The choices represent the source of the pairing and the gaps in the image (the hotspots) representing the targets. Unlike the simple GapMatch Interaction, each gap can have several choices associated with it if desired, furthermore, the same choice may be associated with an AssociableHotspot multiple times, in which case the corresponding directed pair appears multiple times in the value of the response variable.

The QTI 3 Player supports ALL QTI 3 Graphic Gap Match Interaction shared vocabularies and themes. In the absence of any explicit shared vocabulary, the QTI 3 Player displays the set of choices above the image containing the Associable Hotspots.

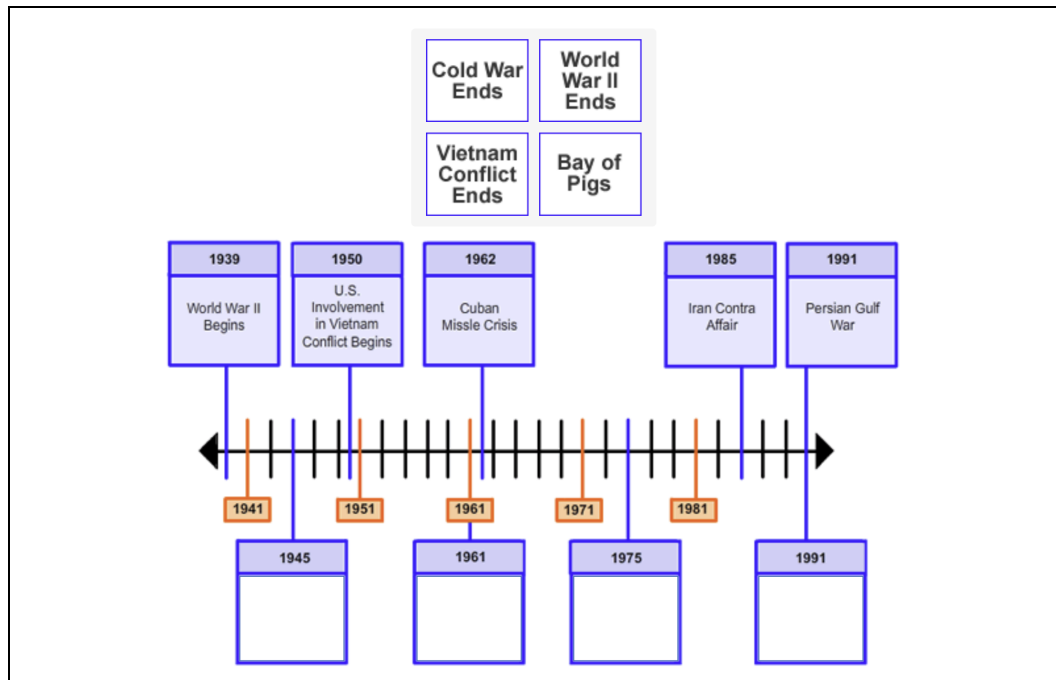
Example of Default (no shared vocabulary) Display



Example of Display with “qti-choices-left” shared vocabulary



Example of Display with “qti-choices-top” shared vocabulary and data-choices-container-width=“188”



qti-gap-match-interaction Supported Attributes

Name	Usage	Value	Default
response-identifier	Required	identifier Must resolve to a response variable of single or multiple cardinality and base-type of directedPair	
class	Optional	string Use class to specify shared vocabulary.	
min-associations	Optional	integer The minimum number of associations that the candidate is required to make to form a valid response. If min-associations is 0	0 (no constraint)

		then the candidate is not required to make any associations. min-associations must be less than or equal to the limit imposed by max-associations.	
max-associations	Optional	integer The maximum number of associations that the candidate is allowed to make. If max-associations is 0 then there is no restriction. If max-associations is greater than 1 (or 0) then the interaction must be bound to a response variable with multiple cardinality.	1
shuffle	Optional	boolean If the shuffle characteristic is true then the QTI 3 Player randomizes the order in which the choices are initially presented, subject to the value of the fixed attribute of each choice.	false
data-max-selections-message	Optional	string	"You may make a maximum of {max-associations} matches for this question."
data-min-selections-message	Optional	string	"You must make at least {min-associations} matches for this question."
data-choices-container-width	Optional	integer The width in pixels of the gap choices container.	

		Example: data-choices-container-width="188"	
--	--	--	--

For a complete description of the QTI 3 Shared Vocabularies for the GraphicGapMatch Interaction, please refer to the QTI 3 Best Practices Implementation Guide. You may also refer to this [QTI 3 Player TestRunner Test](#) which demonstrates implementation and usage of all QTI 3 Graphic Gap Match Interaction conformance items.

QtiHotspotInteraction

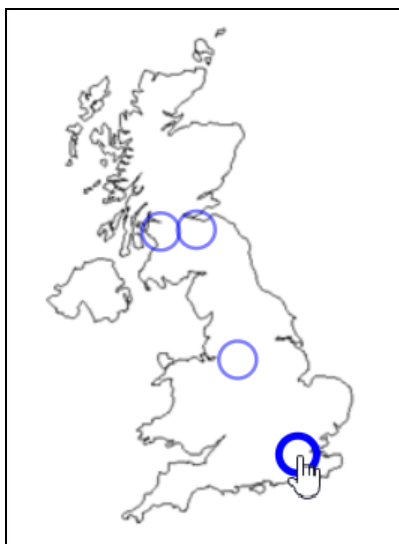
The Hotspot Interaction (qti-hotspot-interaction) is a graphical interaction with a corresponding set of choices that are defined as areas of a background graphic image. The candidate's task is to select one or more of the areas (hotspots). The hotspot interaction should only be used when the spacial relationship of the choices with respect to each other (as represented by the graphic image) is important to the needs of the item.

The hotspot interaction must be bound to a response variable with a base-type of identifier and single or multiple cardinality.

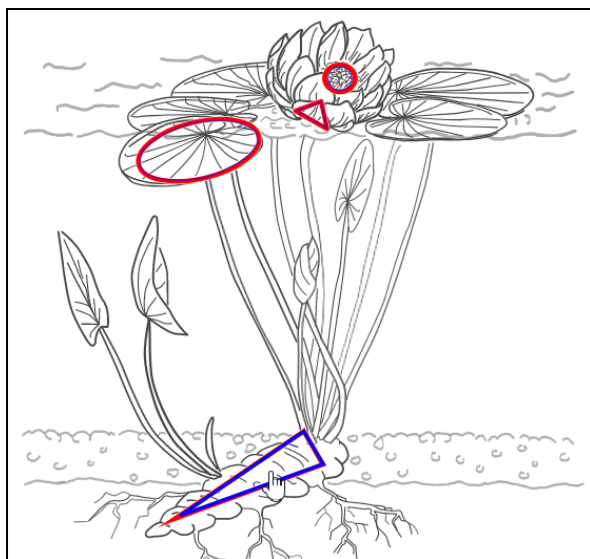
The QTI 3 Player supports the most common QTI 3 Hotspot Interaction HotSpot Choice shapes which are defined by the enumeration of { circle | rect | poly | default | ellipse }. **Note:** the "default" shape is not currently implemented by QTI 3 Player.

The QTI 3 Player supports ALL QTI 3 Hotspot Interaction shared vocabularies and themes. In the absence of any explicit shared vocabulary, the QTI 3 Player displays hotspots with the "qti-selections-dark" theme.

Example of Default (no shared vocabulary) Display and Circle Shapes



Example of Default (no shared vocabulary) Display and Polygon Shapes



qti-hotspot-interaction Supported Attributes

Name	Usage	Value	Default
response-identifier	Required	identifier Must resolve to a response variable identifier.	
class	Optional	string Use class to specify shared vocabulary..	
min-choices	Optional	integer	0
max-choices	Optional	integer	1
data-max-selections-message	Optional	string	"You are permitted a maximum of {max-choices} selections for this question. Please unselect one of your selections before"

			making another selection.”
data-min-selections-message	Optional	string	“You must make at least {min-choices} selections for this question.”

For a complete description of the QTI 3 Shared Vocabularies for the HotSpot Interaction, please refer to the QTI 3 Best Practices Implementation Guide. You may also refer to this [QTI 3 Player TestRunner Test](#) which demonstrates implementation and usage of all QTI 3 HotSpot Interaction conformance items.

QtiHottextInteraction

The HotText Interaction (qti-hottext-interaction) presents a set of choices to the candidate represented as selectable runs of text embedded within a surrounding context, such as a simple passage of text. Like ChoiceInteraction, the candidate's task is to select one or more of the choices, up to a maximum of max-choices.

The interaction is initialized from the qti-default-value of the associated response variable, a NULL value indicating that no choices are selected (the usual case). The qti-hottext-interaction must be bound to a response variable with a base-type of identifier and single or multiple cardinality.

The QTI 3 Player supports all QTI 3 HotText Interaction shared vocabulary.

Example of Default Display

Select the error in the following passage of text (or *No Error* if there is none).

Sponsors of the Olympic Games **who bought** advertising time on United States television **includes** **at least** a dozen international firms **whose** names are familiar to American consumers. **No error.**

qti-hottext-interaction Supported Attributes

Name	Usage	Value	Default
response-identifier	Required	identifier Must resolve to a response variable identifier.	
class	Optional	string	

		Use class to specify shared vocabulary..	
min-choices	Optional	integer	0
max-choices	Optional	integer	1
data-max-selections-message	Optional	string	“You are permitted a maximum of {max-choices} selections for this question. Please unselect one of your selections before making another selection.”
data-min-selections-message	Optional	string	“You must make at least {min-choices} selections for this question.”

For a complete description of the QTI 3 Shared Vocabularies for the HotText Interaction, please refer to the QTI 3 Best Practices Implementation Guide. You may also refer to this [QTI 3 Player TestRunner Test](#) which demonstrates implementation and usage of all QTI 3 HotText Interaction conformance items.

QtiInlineChoiceInteraction

The Inline Choice Interaction (qti-inline-choice-interaction) is an inline interaction which presents a collection of simple text and/or MathML choices to the candidate in the context of a surrounding text. The interaction is always initialized with no choices selected.

The response variable declaration that is bound to an Inline Choice Interaction must be of single cardinality, with identifier base-type. The QTI 3 Player renders the choices with a select box control.

The QTI 3 Player supports ALL QTI 3 Inline Choice Interaction shared vocabularies. In the absence of any explicit shared vocabulary, the QTI 3 Player displays a select box with a width of 9rem and the text “Choose...” displayed inside the select box container.

Example of Default (no shared vocabulary) Display

Now is the winter of our discontent
 Made glorious summer by this sun of
 And all the clouds that lour'd upon ou
 In the deep bosom of the ocean buried

Choose...
 ✓ Choose...
 Gloucester
 Lancaster
 York

Example of Default (no shared vocabulary) Display with MathML Choices

If $a = b$ and $b = c$ then $a = c$

Choose...
 $x = y$
 ✓ $a = c$
 $b = d$

qti-inline-choice-interaction Supported Attributes

Name	Usage	Value	Default
response-identifier	Required	identifier Must resolve to a response variable identifier.	
class	Optional	string Use class to specify shared vocabulary.	
min-choices	Optional	integer	0
max-choices	Optional	integer Setting max-choices greater than 1, or equal to 0, has no effect on the candidate's ability to select more than one choice.	1
shuffle	Optional	boolean	false

		If the shuffle characteristic is true then the QTI 3 Player randomizes the order in which the choices are initially presented, subject to the value of the fixed attribute of each choice.	
data-prompt	Optional	string Custom text to be rendered when the selection element is in an unselected state.	“You are permitted a maximum of {max-choices} choices for this question. Please unselect one of your choices before making another choice.”
data-min-selections-message	Optional	string	“You must make at least 1 choice for this question.”

For a complete description of the QTI 3 Shared Vocabularies for the Inline Choice Interaction, please refer to the QTI 3 Best Practices Implementation Guide. You may also refer to this [QTI 3 Player TestRunner Test](#) which demonstrates implementation and usage of all QTI 3 Inline Choice Interaction conformance items.

QtiMatchInteraction

The Match Interaction (qti-match-interaction) is a block interaction which presents candidates with two sets ([qti-simple-match-set](#)) of choices and allows them to create associations between pairs of choices in the two sets, but not between pairs of choices in the same set. Further restrictions can still be placed on the allowable associations using the match-max characteristic of the choices.

The Match Interaction must be bound to a response variable with base-type 'directedPair' and either single or multiple cardinality.

The QTI 3 Player supports ALL QTI 3 Match Interaction shared vocabularies. In the absence of any explicit shared vocabulary, the QTI 3 Player displays the first set of choices above the second set of choices.

Example of Default (no shared vocabulary) Display

Match the following characters to the Shakespeare play they appeared in:

Capulet

Demetrius

Lysander

Prospero

A Midsummer-Night's Dream

Romeo and Juliet

The Tempest

Example of Default Display with “qti-choices-left” shared vocabulary

Capulet

Demetrius

Lysander

Prospero

A Midsummer-Night's Dream

Romeo and Juliet

The Tempest

Example of “qti-match-tabular” Display

Demonstrates qti-match-tabular

	A Midsummer-Night's Dream	Romeo and Juliet	The Tempest
Capulet	☐	☒	☐
Demetrius	☐	☐	☐
Lysander	☒	☐	☐
Prospero	☒	☐	☐

qti-match-interaction Supported Attributes

Name	Usage	Value	Default
response-identifier	Required	identifier Must resolve to a response variable of single or multiple cardinality and base-type of directedPair	
class	Optional	string Use class to specify shared vocabulary.	
min-associations	Optional	integer The minimum number of associations that the candidate is required to make to form a valid response. If min-associations is 0 then the candidate is not required to make any associations. min-associations must be less than or equal to the limit imposed by max-associations.	0 (no constraint)
max-associations	Optional	integer The maximum number of associations that the candidate is allowed to make. If max-associations is 0 then there is no restriction. If max-associations is greater than 1 (or 0) then the interaction must be bound to a response variable with multiple cardinality.	1
shuffle	Optional	boolean If the shuffle characteristic is true then the QTI 3 Player randomizes	false

		the order in which the choices are initially presented, subject to the value of the fixed attribute of each choice.	
data-max-selections-message	Optional	string	"You may make a maximum of {max-associations} matches for this question."
data-min-selections-message	Optional	string	"You must make at least {min-associations} matches for this question."
data-first-column-header	Optional	string When qti-match-tabular is specified as the presentation, the custom text to be rendered in the top-left header cell of the table (headings must be visible).	null

For a complete description of the QTI 3 Shared Vocabularies for the Match Interaction, please refer to the QTI 3 Best Practices Implementation Guide. You may also refer to this [QTI 3 Player TestRunner Test](#) which demonstrates implementation and usage of all QTI 3 Match Interaction conformance items.

QtiSimpleMatchSet

The SimpleMatchSet (qti-simple-match-set) element is found in the Match Interaction. A Match Interaction must contain two SimpleMatchSet elements. The order of the SimpleMatchSets is significant. The first SimpleMatchSet in the order contains the first identifiers in all directedPairs produced by the encapsulating MatchInteraction. The second SimpleMatchSet in the order contains the last identifiers in all directedPairs produced by the encapsulating MatchInteraction.

qti-simple-match-set Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiOrderInteraction

The Order Interaction (qti-order-interaction) is a block interaction which presents a collection of simple choices to the candidate with the candidate's task to reorder the choices. The order in which the choices are displayed initially is significant. By default the candidate's task is to order all of the choices but a subset of the choices can be requested using the max-choices and min-choices attributes. When specified the candidate must select a subset of the choices and impose an ordering on them.

The response variable declaration that is bound to an Order Interaction must be of ordered cardinality, with identifier base-type.

The QTI 3 Player supports ALL QTI 3 Order Interaction shared vocabularies. In the absence of any explicit shared vocabulary, the QTI 3 Player displays the orderable simple choices from left to right in a horizontal orientation.

Example of Default (no shared vocabulary) Display

The following F1 drivers finished on the podium in the first ever Grand Prix of Bahrain. Can you rearrange them into the correct finishing order from left to right, 1st, 2nd, and 3rd?

Rubens Barrichello Jenson Button Michael Schumacher

Note: This conformance example has no QTI 3 shared presentation vocabulary. Consequently, QTI 3 Player renders the *Order Interaction* above with QTI 3 Player defaults; i.e., as a series of choices with horizontal orientation.

Example of Default (no shared vocabulary) Display with orientation="horizontal"

The following F1 drivers finished on the podium in the first ever Grand Prix of Bahrain. Can you rearrange them into the correct finishing order from left to right, 1st, 2nd, and 3rd?

Rubens Barrichello Jenson Button Michael Schumacher

Note: The *orientation* of the layout of the drivers should be horizontal.

Example of Default (no shared vocabulary) Display with orientation="vertical"

The following F1 drivers finished on the podium in the first ever Grand Prix of Bahrain. Can you rearrange them into the correct finishing order - from top to bottom - 1st, 2nd, and 3rd?

Rubens Barrichello

Jenson Button

Michael Schumacher

Note: The *orientation* of the layout of the drivers should be vertical.

qti-order-interaction Supported Attributes

Name	Usage	Value	Default
response-identifier	Required	identifier Must resolve to a response variable of ordered cardinality and identifier.	
class	Optional	string Use class to specify shared vocabulary.	
min-choices	Optional	integer The minimum number of choices that the candidate must select and order to form a valid response to the interaction. If specified, min-choices must be '1' or greater but must not exceed the number of choices available. If unspecified, all of the choices must be ordered and max-choices is ignored.	0 (no constraint)
max-choices	Optional	integer The maximum number of choices that the candidate may select and order when responding to the	0 (no limit)

		interaction. Used in conjunction with minChoices, if specified, max-choices must be greater than or equal to min-choices and must not exceed the number of choices available. If unspecified, all of the choices may be ordered.	
shuffle	Optional	boolean If the shuffle characteristic is true then the QTI 3 Player randomizes the order in which the choices are initially presented, subject to the value of the fixed attribute of each choice.	false
orientation	Optional	string Use orientation to specify the orientation of the choices (and potentially the targets) that QTI 3 Player renders. Enumerated value set of: { "horizontal" "vertical" }	"horizontal"
data-max-selections-message	Optional	string	"You may set an order for a maximum of {max-choices} choices for this question."
data-min-selections-message	Optional	string	"You must set the order for at least {min-choices} choices for this question."

For a complete description of the QTI 3 Shared Vocabularies for the Order Interaction, please refer to the QTI 3 Best Practices Implementation Guide. You may also refer to this [QTI 3 Player TestRunner Test](#) which demonstrates implementation and usage of all QTI 3 Order Interaction conformance items.

QtiTextEntryInteraction

A Text Entry Interaction (qti-text-entry-interaction) is an inline Interaction that obtains a simple piece of text from the candidate in the context of a surrounding text. The Text Entry Interaction must be bound to a response variable with single cardinality only. Text Entry response variables of multiple, ordered, or record cardinality are not supported by the QTI 3 Player. The base-type must be one of string, integer or float.

The QTI 3 Player supports ALL QTI 3 Text Entry Interaction shared vocabularies. In the absence of any explicit shared vocabulary, the QTI 3 Player displays an input area with a width of 8.6rem, and no placeholder text displayed inside the input area.

Example of Default (no shared vocabulary) Display

Now is the winter of our discontent

Made glorious summer by this sun of ;

And all the clouds that lour'd upon our house

In the deep bosom of the ocean buried.

qti-text-entry-interaction Supported Attributes

Name	Usage	Value	Default
response-identifier	Required	identifier Must resolve to a response variable identifier.	
class	Optional	string Use class to specify shared vocabulary.	
pattern-mask	Optional	string If given, the pattern mask specifies a regular expression that the candidate's response must match in order to be considered valid. The regular expression language used is defined here:	

		https://www.w3.org/TR/xmlschema-2/#regexs Note: This is an XML Regex expression.	
placeholder-text	Optional	string By default, QTI 3 Player text entry interactions display no placeholder text. Use this attribute to override the default behavior.	
data-patternmask-message	Optional	string Override QTI 3 Player's default pattern-mask error message.	"Invalid Input"

For a complete description of the QTI 3 Shared Vocabularies for the Choice Interaction, please refer to the QTI 3 Best Practices Implementation Guide. You may also refer to the [QTI 3 Player Sandbox](#).

QtiCustomInteraction

The Custom Interaction (qti-custom-interaction) provides an extension point for delivery platforms such as QTI 3 Player to render interactions and behaviors that are not provided by the built-in interaction types in QTI 3.

One custom interaction (class="amp:equation-editor") is provided under a proprietary license and not available in the open source, MIT-licensed QTI 3 Player. Contact support@amp-up.io for licensing information.

qti-custom-interaction Supported Attributes

Name	Usage	Value	Default
response-identifier	Required	identifier Must resolve to a response variable identifier.	
class	Optional	string	

		Use class to specify the custom interaction; e.g., class="amp:equation-editor"	
--	--	---	--

amp:equation-editor Configuration

The amp:equation-editor must be bound to a response variable of single cardinality and base-type of string. The amp:equation-editor is configured via a child XML element named "<custom-option>". The Custom Option element contains a JSON string with the following properties:

Name	Usage	Value	Default
restrictKeys	Optional	boolean Constrain user input to the key commands on the palette.	false
paletteConfig	Optional	Object containing a "palette" property; e.g., { "palette": "palette identifier" } Palette identifiers are from the enumerated value set of: { numeric-basic numeric-operators-basic numeric-fraction-basic all-palettes }	"numeric-basic"

@author Paul Grudnitski, amp-up.io

@license - The amp:equation-editor custom interaction software is proprietary software copyright amp-up.io and may not be shared, copied, or distributed without the explicit written consent of amp-up.io.

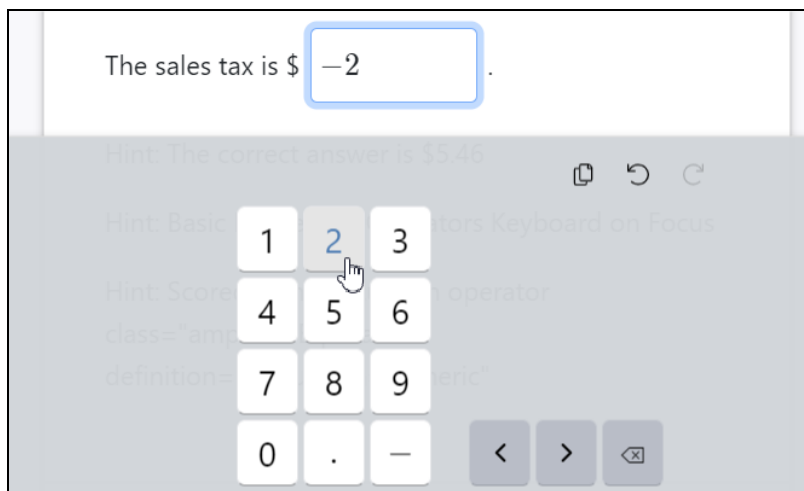
Example 1 of Equation Editor configuration

```

0001 <!-- Equation Editor must be bound to response variable of base-type string and
0002      cardinality of single -->
0003
0004 <qti-response-declaration
0005   base-type="string"
0006   cardinality="single"
0007   response-identifier="RESPONSE" />
0008
0009
0010 <!-- in the item body ... -->
0011 <qti-custom-interaction class="amp:equation-editor" response-identifier="RESPONSE">
0012   <custom-option>
0013     {
0014       "restrictKeys": true,
0015       "paletteConfig": {
0016         "palette": "numeric-basic"
0017       }
0018     }
0019   </custom-option>
0020 </qti-custom-interaction>
0021
0022 <!-- Custom Option configuration above specifies restrict keys to the keys in the palette.
0023      With the numeric basic palette, only the digits keys, a decimal point, and a minus sign are
0024      permitted -->

```

Example 1 of Equation Editor (numeric-basic palette) Rendering



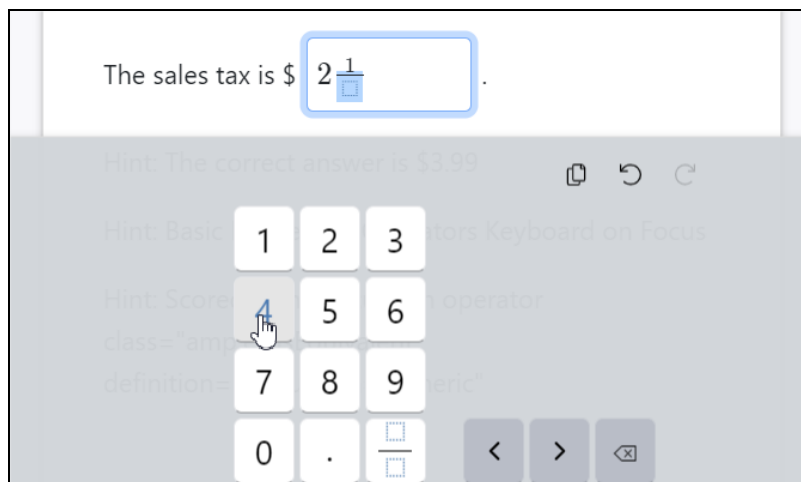
Example 2 of Equation Editor configuration

```

0001 <!-- Equation Editor must be bound to response variable of base-type string and
0002      cardinality of single -->
0003
0004 <qti-response-declaration
0005   base-type="string"
0006   cardinality="single"
0007   response-identifier="RESPONSE" />
0008
0009
0010 <!-- in the item body ... -->
0011 <qti-custom-interaction class="amp:equation-editor" response-identifier="RESPONSE">
0012   <custom-option>
0013     {
0014       "restrictKeys": true,
0015       "paletteConfig": {
0016         "palette": "numeric-fraction-basic"
0017       }
0018     }
0019   </custom-option>
0020 </qti-custom-interaction>
0021
0022 <!-- Custom Option configuration above specifies restrict keys to the keys in the palette.
0023      With the numeric fraction basic palette, only the digits keys, a decimal point, and a
0024      fraction symbol ("/") are permitted -->

```

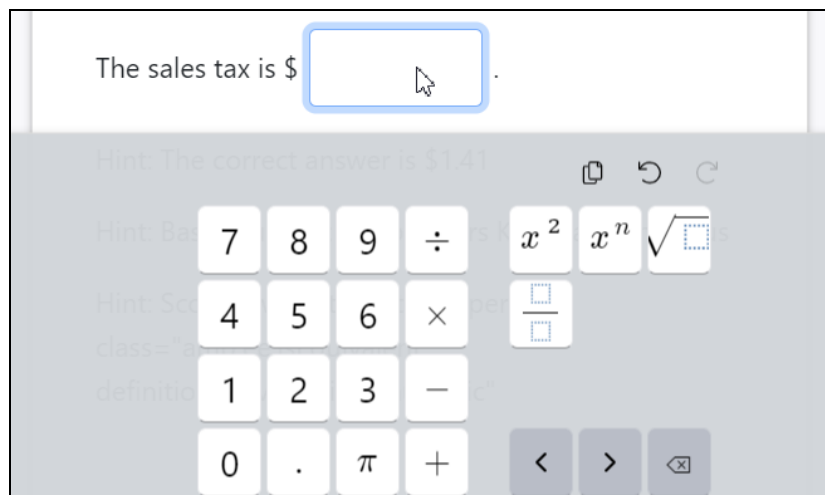
Example 2 of Equation Editor (numeric-fraction-basic palette) Rendering



Example 3 of Equation Editor configuration

0001	<!-- Equation Editor must be bound to response variable of base-type string and
0002	cardinality of single -->
0003	
0004	<qti-response-declaration
0005	base-type="string"
0006	cardinality="single"
0007	response-identifier="RESPONSE" />
0008	
0009	
0010	<!-- in the item body ... -->
0011	<qti-custom-interaction class="amp:equation-editor" response-identifier="RESPONSE">
0012	<custom-option>
0013	{
0014	"restrictKeys": true,
0015	"paletteConfig": {
0016	"palette": "numeric-operators-basic"
0017	}
0018	}
0019	</custom-option>
0020	</qti-custom-interaction>
0021	
0022	<!-- Custom Option configuration above specifies restrict keys to the keys in the palette.
0023	With the numeric operators basic palette, the digits keys, a decimal point, the symbol pi,
0024	add, subtract, multiply, divide, fraction symbol, power of 2, power of n, and a
	square root symbol are permitted -->

Example 3 of Equation Editor (numeric-operators-basic palette) Rendering



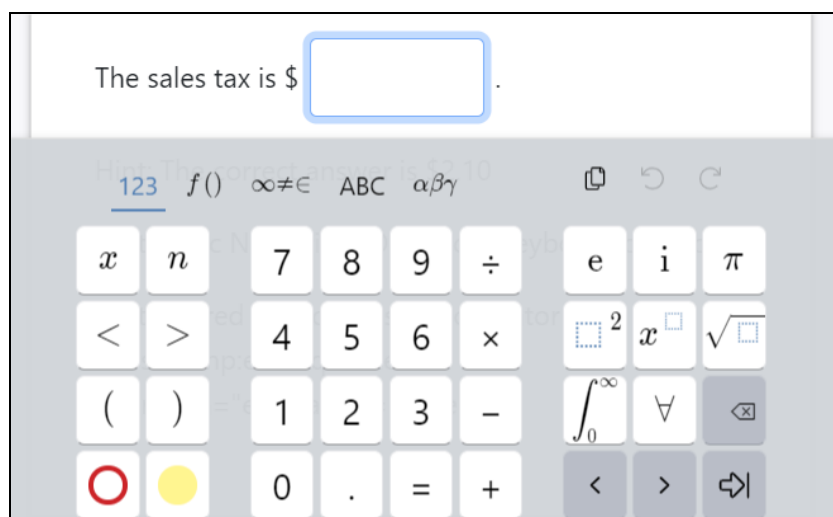
Example 4 of Equation Editor configuration

```

0001 <!-- Equation Editor must be bound to response variable of base-type string and
0002      cardinality of single -->
0003
0004 <qti-response-declaration
0005   base-type="string"
0006   cardinality="single"
0007   response-identifier="RESPONSE" />
0008
0009
0010 <!-- in the item body ... -->
0011 <qti-custom-interaction class="amp:equation-editor" response-identifier="RESPONSE">
0012   <custom-option>
0013     {
0014       "restrictKeys": false,
0015       "paletteConfig": {
0016         "palette": "all-palettes"
0017       }
0018     }
0019   </custom-option>
0020 </qti-custom-interaction>
0021
0022 <!-- Custom Option configuration above uses the all-palettes configuration. In this
0023      configuration the "restrictKeys" option has no effect because all keys are permitted. -->
0024

```

Example 4 of Equation Editor (all-palettes palette) Rendering



QtiEndAttemptInteraction

The End Attempt interaction (qti-end-attempt-interaction) is a special type of interaction which allows item authors to provide the candidate with control over the way in which the candidate terminates an attempt. The candidate can use the interaction to terminate the attempt (triggering response processing) immediately, typically - but not always - to request a hint. An End Attempt Interaction must be bound to a response variable with base-type boolean and single cardinality.

An End Attempt interaction can be combined with an item's `adaptive="true"` attribute to provide dynamic assessment experiences. The QTI 3 Player supports additional instrumentation of the End Attempt interaction via data- extension points/attributes.

qti-end-attempt-interaction Supported Attributes

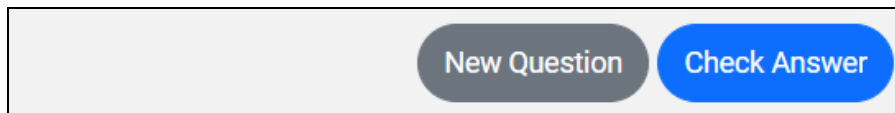
Name	Usage	Value	Default
response-identifier	Required	identifier Must resolve to a response variable identifier of base-type="boolean" and cardinality="single"	
class	Optional	string Use class to trigger the special behaviors of the "endattemptl-controller-bar"; e.g., class="endattempt-controller-bar" on TDX, TDX-ShowExample, and TDX-SolveThis items.	
title	Required	string The string that should be displayed to the candidate as a prompt for ending the attempt using this interaction. This should be short, preferably one word. A typical value would be "Hint" or "Check Answer". In the QTI 3 Player, this is presented as the label on a button that, when pressed, ends the attempt.	
data-steps	Optional	integer Extension point for specifying the max number of steps for an item. This should be a string that evaluates to a positive integer (1..n); e.g., data-steps="8" on an adaptive item with 8 steps.	

data-controllertype	Optional	string Extension point for specifying subtypes of endattempt-controller-bar: { "generic" "solve" "showexample" } It is <i>essential</i> to set this to “solve” or “showexample” on Solve This and Show Example items, respectively.	“generic”
data-hastemplates	Optional	boolean Extension point for specifying that this end attempt interaction of class=“end attempt-controller-bar” should display a New Question button, thus enabling the candidate to generate a new template - for example, on a Show Example or Solve This item).	false
data-hideprogress	Optional	boolean Extension point for specifying whether or not to display the controller bar progress-panel. Enumerated value set of { "true" "false" }	false Progress is shown by default.

Example 1 of End Attempt endattempt-controller-bar configuration

0001	<!-- The End Attempt Interaction MUST be bound to response variable of base-type boolean and
0002	cardinality of single -->
0003	
0004	<qti-response-declaration
0005	base-type="boolean"
0006	cardinality="single"
0007	response-identifier="EndAttempt" />
0008	
0009	
0010	<!-- in the item body ... -->
0011	<qti-end-attempt-interaction
0012	title="Check Answer"
0013	class="endattempt-controller-bar"
0014	data-steps="1"
0015	data-hideprogress="true"
0016	data-hastemplates="true"
0017	response-identifier="EndAttempt" />
0018	
0019	
0020	<!-- The endattempt-controller-bar does not display the progress stepper because
0021	data-hideprogress="true". The main button is labeled as "Check Answer" because
0022	the title="Check Answer". A "New Question" button is shown because
0023	data-hastemplates="true". -->

Example 1 Rendering



Example 2 of End Attempt endattempt-controller-bar configuration

0001	<!-- The End Attempt Interaction MUST be bound to response variable of base-type boolean and
0002	cardinality of single -->
0003	
0004	<qti-response-declaration
0005	base-type="boolean"
0006	cardinality="single"
0007	response-identifier="EndAttempt" />
0008	
0009	
0010	<!-- in the item body ... -->
0011	<qti-end-attempt-interaction
0012	title="Continue"
0013	class="endattempt-controller-bar"
0014	data-steps="8"
0015	data-controllertype="showexample"
0016	data-hastemplates="true"
0017	response-identifier="EndAttempt" />
0018	
0019	
0020	<!-- The endattempt-controller-bar shows a main button labeled "Continue" because
0021	title="Continue". A "New Question" button is shown because
0022	data-hastemplates="true". The stepper shows 8 steps because data-steps="8".

```
The controller bar has showexample behaviors because data-controllertype="showexample"
-->
```

Example 2 Rendering



QtiPrintedVariable

For a Printed Variable to be valid, the outcome variable or template variable must have been defined. The values of response variables cannot be printed directly as their values are implicitly known to the candidate through the interactions they are bound to; if necessary, their values can be assigned to outcomes during responseProcessing and displayed to the candidate as part of an element visible only in the appropriate feedback states.

If the variable's value is NULL then the element is ignored. Variables of baseType string are treated as simple runs of text. Variables of baseType integer or float are converted to runs of text (strings) using the formatting rules described below. Float values should only be formatted in the e, E, f, g, G, r or R styles.

For variables of multiple cardinality, a list of the values within the container is printed, delimited by the string value of the delimiter attribute. Printed variables of record cardinality are unsupported.

qti-printed-variable Supported Attributes

Name	Usage	Value	Default
identifier	Required	string Must resolve to an Outcome, Template, or Context Variable. Response variables are not permitted.	
format	Optional	string The format conversion specifier to use when converting numerical values to strings. The QTI 3 Player uses a "printf" library to apply the format attribute to the variable value.	

QtiPrompt

This is an optional prompt element that can be used to guide the learner. The qti-prompt (“prompt”) element is found in certain interactions. A prompt must not contain any nested interactions. A prompt is not the same as the actual question construct.

qti-prompt Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiRubricBlock

A RubricBlock (qti-rubric-block) element is a container that identifies part of the content that represents instructions to one or more of the actors that view the item. Although RubricBlocks are defined as simpleBlocks they must not contain interactions. The visibility of nested RubricBlocks is determined by the outermost element. In other words, if an element is determined to be hidden then all of its content is hidden including conditionally visible elements for which the conditions are satisfied and that therefore would otherwise be visible.

qti-rubric-block Supported Attributes

Name	Usage	Value	Default
view	Required	identifier The views in which the rubric block's content are to be shown. A list from an enumerated value set of: { author candidate proctor scorer testConstructor tutor } If 'view' contains 'scorer', this rubric block will be hidden from candidates.	
use	Optional	string Denotes how the content contained within the rubric block is intended to be used within	

		systems presenting the rubric block content.	
--	--	--	--

QtiAssociableHotspot

An AssociableHotspot (qti-associable-hotspot) is used to define the hotspots that are associated with the features in the GraphicGapMatch Interaction.

qti-associable-hotspot Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string. The identifier must not be used on any other associable hotspot in the enclosing interaction.	
coords	Required	string The coords attribute provides the coordinates that determine the size and location of an area defined by a corresponding shape. The coordinates themselves are an ordered list of lengths. The interpretation of each length value is dependent on the value of the associated shape as follows: rect: left-x, top-y, right-x, bottom-y; Note that the 'rect' shape is the only supported shape for AssociableHotspots in the QTI 3 Item Player.	
shape	Required	string Note that the 'rect' shape is the only supported shape for	

		AssociableHotspots in the QTI 3 Item Player.	
match-group	Optional	This is used to identify the set of entities that can be matched with this AssociableHotspot. This enables the author to constrain the set of objects that can be matched with the target choice.	
match-max	Required	integer The maximum number of entities this AssociableHotspot may be associated with. If match-max is 0 then there is no restriction.	

QtiGap

A Gap (qti-gap) defines the gap structure that must only appear within a GapMatch Interaction. A Gap may contain at most one GapText or one GapImg choice.

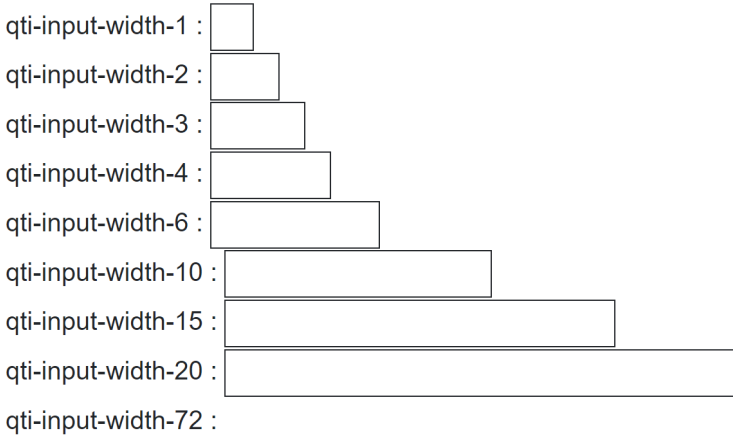
The QTI 3 Player supports ALL QTI 3 Gap shared vocabularies. In the absence of any explicit shared vocabulary, the QTI 3 Player displays a gap with width of 100px.

Example of GapMatch Interaction with Gaps of various qti-input-widths

To specify gap element widths, use the class attribute and qti-input-width-1|2|3|4|6|10|15|20|72 The gap element widths are intended to provide a general visual impression. By default, in the absence of any shared vocabulary, the presentation width of a gap element is left to the Delivery Platform.

Demonstrates all gap element qti-input-width's: 1|2|3|4|6|10|15|20|72

M MM MMM MMMM MMMMMM



qti-gap Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string. The identifier must not be used on any other choice in the enclosing interaction.	
class	Optional	string Use class to specify shared vocabulary. Example: class="qti-input-width-1"	

match-group	Optional	This is used to identify the set of entities that can be matched with this gap. This enables the author to constrain the set of objects that can be matched with the target gap.	
-------------	----------	--	--

QtiGapImg

qti-gap-img Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string. The identifier must not be used on any other choice in the enclosing interaction.	
fixed	Optional	boolean If fixed is true for a choice then the position of this choice within the interaction remains static even if the immediately enclosing interaction supports the shuffling of choices. If no value is specified then the choice is free to be shuffled.	
match-group	Optional	This is used to identify the set of entities that can be matched with this choice. This enables the author to constrain the set of objects that can be matched with the target choice.	
match-max	Required	integer The maximum number of choices this choice may be associated	

		with. If match-max is 0 then there is no restriction.	
--	--	---	--

QtiGapText

qti-gap-text Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string. The identifier must not be used on any other choice in the enclosing interaction.	
fixed	Optional	boolean If fixed is true for a choice then the position of this choice within the interaction remains static even if the immediately enclosing interaction supports the shuffling of choices. If no value is specified then the choice is free to be shuffled.	
match-group	Optional	This is used to identify the set of entities that can be matched with this choice. This enables the author to constrain the set of objects that can be matched with the target choice.	
match-max	Required	integer The maximum number of choices this choice may be associated with. If match-max is 0 then there is no restriction.	

QtiSimpleChoice

qti-simple-choice Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string. The identifier must not be used on any other choice in the enclosing interaction.	
fixed	Optional	boolean If fixed is true for a choice then the position of this choice within the interaction remains static even if the immediately enclosing interaction supports the shuffling of choices. If no value is specified then the choice is free to be shuffled.	

The QTI 3 Player supports math (MathML) variable templating in qti-simple-choices.

QtiSimpleAssociableChoice

qti-simple-associable-choice Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string. The identifier must not be used on any other choice in the enclosing interaction.	
fixed	Optional	boolean If fixed is true for a choice then the position of this choice within the interaction remains static	

		even if the immediately enclosing interaction supports the shuffling of choices. If no value is specified then the choice is free to be shuffled.	
match-group	Optional	This is used to identify the set of entities that can be matched with this choice. This enables the author to constrain the set of objects that can be matched with the target choice.	
match-max	Required	integer The maximum number of choices this choice may be associated with. If match-max is 0 then there is no restriction.	

The QTI 3 Player supports math (MathML) variable templating in qti-simple-associable-choices.

QtiHotspotChoice

A HotspotChoice (qti-hotspot-choice) element defines a hotspot choice that can be selected by the candidate. QTI 3 Player supports pointer-based and keyboard-based hotspot selection. Keyboard navigation of hotspot choices follows the same order in which the hotspot choices are defined.

qti-hotspot-choice Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string. The identifier must not be used on any other hotspot choice in the enclosing interaction.	
coords	Required	string The coords attribute provides the coordinates that determine the	

		size and location of an area defined by a corresponding shape. The coordinates themselves are an ordered list of lengths. The interpretation of each length value is dependent on the value of the associated shape as follows. (a) rect: left-x, top-y, right-x, bottom-y; (b) circle: center-x, center-y, radius. (c) poly: x1, y1, x2, y2, ..., xN, yN. The first x and y coordinate pair and the last should be the same to close the polygon. (d) ellipse: center-x, center-y, h-radius, v-radius. Note that the ellipse shape is deprecated as it is not defined by XHTML; (e) default: no coordinates should be given.	
shape	Required	string Supported shapes are from the enumerated (case-sensitive) value set of: { circle rect poly ellipse } Note: “ellipse” is supported, but deprecated. “default” is not currently supported	

QtiHottext

A HotText (qti-hottext) element defines a HotText area that is used within the content of a QtiHottextInteraction to provide the individual choices. It must not contain any nested interactions or other hottext areas.

qti-hottext Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier	

		Must be a valid identifier string. The identifier must not be used on any other hottext choice in the enclosing interaction.	
--	--	--	--

QtiInlineChoice

qti-inline-choice Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string. The identifier must not be used on any other choice in the enclosing interaction.	
fixed	Optional	boolean If fixed is true for a choice then the position of this choice within the interaction remains static even if the immediately enclosing interaction supports the shuffling of choices. If no value is specified then the choice is free to be shuffled.	

The QTI 3 Player supports math (MathML) variable templating in qti-inline-choices.

Miscellaneous Special HTML Elements

Object

The QTI 3 Player provides support for objects with types that are generally described as “image” types. Currently, no audio or video type component support is provided via the object element. Support for the HTML5 audio and video elements is on the development roadmap.

MathML

The QTI 3 Player component has a dependency on MathJax for MathML rendering. However, it is the responsibility of the QTI 3 Player’s *container* application to load MathJax into the DOM. For example, the QTI 3 Player Sampler application loads MathJax 2.7 with the following script tag:

0001	<script
0002	src='https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.0/MathJax.js?config=TeX-AMS-MML_HTMLorMML'>
0003	</script>

Response and Template Processing Elements

QtiResponseProcessing

Response processing is the process by which the QTI 3 Player assigns outcomes based on the candidate's responses. The outcomes may be used to provide feedback to the candidate. Feedback is either provided immediately following the end of the candidate's attempt or it is provided at some later time, perhaps as part of a summary report on the item session. The end of an attempt, and therefore response processing, must only take place in direct response to a user action or in response to some expected event, such as the end of a test. An item session that enters the suspended state may have values for the response variables that have yet to be submitted for response processing.

qti-response-processing Supported Attributes

Name	Usage	Value	Default
template	Optional	string The rules obtained from the template string are used unless there are internal rules defined within the response processing element itself. <i>Per best practices, when there are internal rules, the template attribute is ignored.</i> Supported templates are from the enumerated (case-sensitive) value set of: { match_correct.xml match_correct map_response.xml map_response map_response_point.xml map_response_point }	

Examples of QtiResponseProcessing with the “template” attribute

0001	<qti-response-processing
0002	template=" https://purl.imsglobal.org/spec/qti/v3p0/rptemplates/match_correct.xml ">
0003	
0004	<!-- Player resolves the above as a “Match Correct” template because the template attribute
0005	terminates with the “match_correct.xml” string -->
0006	
0007	<qti-response-processing
0008	template=" https://purl.imsglobal.org/spec/qti/v3p0/rptemplates/match_correct ">

```

0009
0010 <!-- Player resolves the above as a "Match Correct" template because the template attribute
0011 terminates with the "match_correct" string. -->
0012
0013 <qti-response-processing
0014 template="https://purl.imsglobal.org/spec/qti/v3p0/rptemplates/map_response.xml"/>
0015
0016 <!-- Player resolves the above as a "Map Response" template because the template attribute
0017 terminates with the "map_response.xml" string -->
0018
0019 <qti-response-processing
0020 template="https://purl.imsglobal.org/spec/qti/v3p0/rptemplates/map_response"/>
0021
0022 <!-- Player resolves the above as a "Map Response" template because the template attribute
0023 terminates with the "map_response" string -->
0024
0025 <qti-response-processing
0026 template="https://purl.imsglobal.org/spec/qti/v3p0/rptemplates/match_correct.xml">
0027   <qti-response-condition>
0028     <qti-response-if>
0029       <qti-match>
0030         <qti-variable identifier="RESPONSE"/>
0031         <qti-correct identifier="RESPONSE"/>
0032       </qti-match>
0033       <qti-set-outcome-value identifier="SCORE">
0034         <qti-base-value base-type="float">4</qti-base-value>
0035       </qti-set-outcome-value>
0036     </qti-response-if>
0037     <qti-response-else>
0038       <qti-set-outcome-value identifier="SCORE">
0039         <qti-base-value base-type="float">0</qti-base-value>
0040       </qti-set-outcome-value>
0041     </qti-response-else>
0042   </qti-response-condition>
0043 </qti-response-processing>
0044
0045 <!-- Player ignores the RP "template" attribute because there are internal RP rules -->

```

QtiTemplateProcessing

Template processing consists of one or more templateRules that are followed by the QTI 3 Player in order to assign values to the template variables. Template processing is identical in form to responseProcessing except that the purpose is to assign values to template variables, not outcome variables.

At the end of each invocation of the template processing rules, the QTI 3 Player updates inline/block template visibility, templated MathML and associated Printed Variables.

qti-template-processing Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

Response and Template Processing Rule Elements

QtiExitResponse

The Exit Response (qti-exit-response) rule terminates Response Processing immediately (for this invocation).

qti-exit-response Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiExitTemplate

The Exit Template (qti-exit-template) rule terminates Template Processing immediately (for this invocation).

qti-exit-template Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiLookupOutcomeValue

The LookupOutcomeValue (qti-lookup-outcome-value) rule sets the value of an outcome variable to the value obtained by looking up the value of the associated expression in the lookupTable associated with the outcome's declaration.

qti-lookup-outcome-value Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string that resolves to an outcome variable.	

QtiResponseCondition

This enables the 'If..Then..Else' rules to be defined for the response processing. If the expression given in a ResponseSelf or ResponseElsef evaluates to 'true' then the sub-rules contained within it are followed

and any following ResponseElseIf or ResponseElse parts are ignored for this response condition. If the expression given in a ResponseIf or ResponseElseIf does not evaluate to 'true' then consideration passes to the next ResponseElseIf or, if there are no more ResponseElseIf parts then the sub-rules of the ResponseElse are followed (if specified).

qti-response-condition Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiResponseIf

A ResponseIf (qti-response-if) rule consists of an expression which must have an effective baseType of boolean and single cardinality. It also contains a set of sub-rules. If the expression is true then the sub-rules are processed, otherwise they are skipped (including if the expression is NULL) and the following ResponseElseIf or ResponseElse parts (if any) are considered instead.

qti-response-if Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiResponseElseIf

The ResponseElseIf (qti-response-else-if) rule provides the 'ElseIf' clauses of the 'If..Then..Else' for the response processing functionality. A ResponseElseIf rule consists of an expression which must have an effective baseType of boolean and single cardinality. It also contains a set of sub-rules. If the expression is 'true' then the sub-rules are processed, otherwise they are skipped (including if the expression is NULL) and the following ResponseElseIf or ResponseElse parts (if any) are considered instead.

qti-response-else-if Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiResponseElse

The ResponseElse (qti-response-else) rule provides the 'Else' clause of the 'If..Then..Else' for the response processing functionality. If the expression given in a ResponseIf or ResponseElseIf evaluates to 'true' then the sub-rules contained within it are followed and any following ResponseElseIf or ResponseElse parts are ignored for this response condition. If the expression given in a ResponseIf or ResponseElseIf does

not evaluate to 'true' then consideration passes to the next ResponseElse or, if there are no more ResponseElse parts then the sub-rules of the responseElse are followed (if specified).

qti-response-else Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiResponseProcessingFragment

A ResponseProcessingFragment (qti-response-processing-fragment) is a simple, transparent group of ResponseRules.

qti-response-processing-fragment Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiSetCorrectResponse

The SetCorrectResponse (qti-set-correct-response) rule results in the response variable having its correct value set.

qti-set-correct-response Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string that resolves to a response variable.	

QtiSetDefaultValue

The SetDefaultValue (qti-set-default-value) rule results in the response variable or outcome variable having its default value set. Curiously, the specification does not permit setting of template variable default values.

qti-set-default-value Supported Attributes

Name	Usage	Value	Default
------	-------	-------	---------

identifier	Required	identifier Must be a valid identifier string that resolves to a response or outcome variable.	
------------	----------	--	--

QtiSetOutcomeValue

The SetOutcomeValue (qti-set-outcome-value) rule sets the value of an outcome variable to the value obtained from the associated expression. An outcome variable can be updated with reference to a previously assigned value, in other words, the outcome variable being set may appear in the expression where it takes the value previously assigned to it. Special care is required when using the numeric base-types because floating point values can not be assigned to integer variables and vice-versa. The QtiTruncate, QtiRound or QtiIntegerToFloat operators must be used to achieve numeric type conversion.

qti-set-outcome-value Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string that resolves to an outcome variable.	

QtiSetTemplateValue

The SetTemplateValue (qti-set-template-value) rule sets the value of a template variable to the value obtained from the associated expression. A template variable can be updated with reference to a previously assigned value, in other words, the template variable being set may appear in the expression where it takes the value previously assigned to it. Special care is required when using the numeric base-types because floating point values can not be assigned to integer variables and vice-versa. The QtiTruncate, QtiRound or QtiIntegerToFloat operators must be used to achieve numeric type conversion.

qti-set-template-value Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string that resolves to a template variable.	

QtiTemplateCondition

If the expression given in a templateIf or templateElseIf evaluates to true then the sub-rules contained within it are followed and any following templateElseIf or templateElse parts are ignored for this template condition. If the expression given in a templateIf or templateElseIf does not evaluate to true then consideration passes to the next templateElseIf or, if there are no more templateElseIf parts then the sub-rules of the templateElse are followed (if specified).

qti-template-condition Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiTemplateIf

A TemplateIf (qti-template-if) rule consists of an expression which must have an effective baseType of boolean and single cardinality. It also contains a set of sub-rules. If the expression is true then the sub-rules are processed, otherwise they are skipped (including if the expression is NULL) and the following TemplateElseIf or TemplateElse parts (if any) are considered instead.

qti-template-if Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiTemplateElseIf

The TemplateElseIf (qti-template-else-if) rule provides the 'Elseif' clauses of the 'If..Then..Else' for the template processing functionality. A TemplateElseIf rule consists of an expression which must have an effective baseType of boolean and single cardinality. It also contains a set of sub-rules. If the expression is 'true' then the sub-rules are processed, otherwise they are skipped (including if the expression is NULL) and the following TemplateElseIf or TemplateElse parts (if any) are considered instead.

qti-template-else-if Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiTemplateElse

The TemplateElse (qti-template-else) rule provides the 'Else' clause of the 'If..Then..Else' for the template processing functionality. If the expression given in a TemplateIf or TemplateElseIf evaluates to 'true' then the sub-rules contained within it are followed and any following TemplateElseIf or TemplateElse parts are ignored for this template condition. If the expression given in a TemplateIf or TemplateElseIf does not evaluate to 'true' then consideration passes to the next TemplateElseIf or, if there are no more TemplateElseIf parts then the sub-rules of the TemplateElse are followed (if specified).

qti-template-else Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiTemplateConstraint

A TemplateConstraint (qti-template-constraint) contains an expression which must have an effective baseType of boolean and single cardinality. If the expression is 'false' (including if the expression is NULL), the template variables are set to their default values and templateProcessing is restarted; this happens repeatedly until the expression is 'true' or the maximum number of iterations (in QTI 3 Player max iterations is 100) is reached. In the event that the maximum number of iterations is reached, any default values provided for the variables during declaration are used. Processing then continues with the next templateRule after the templateConstraint, or finishes if there are no further templateRules. By using a templateConstraint, authors can ensure that the values of variables set during templateProcessing satisfy the condition specified by the boolean expression.

For example, two randomly selected numbers might be required which have no common factors. A templateConstraint may occur anywhere as a child of templateProcessing. It may not be used as a child of any other element. Any number of templateConstraints may be used, though two or more consecutive templateConstraints could be combined using the 'and' element to combine their boolean expressions. The maximum number of times that the operations preceding the templateConstraint can be expected to be performed is 100. This prevents the occurrence of an endless loop. It is the responsibility of the author to provide default values for any variables assigned under a templateConstraint.

qti-template-constraint Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

Processing Expression Elements

QtiAnd

This provides the logic 'and' operation (qti-and) across one or more other operators. The and operator takes one or more sub-expressions each with a base-type of boolean and single cardinality. The result is a single boolean which is true if all sub-expressions are true and false if any of them are false.

If one or more sub-expressions are NULL and all others are true then the operator also results in NULL.

qti-and Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiAnyN

The AnyN (qti-any-n) operator takes one or more sub-expressions each with a base-type of boolean and single cardinality. The result is a single boolean which is 'true' if at least min of the sub-expressions are 'true' and at most max of the sub-expressions are 'true'. If more than n - min sub-expressions are 'false' (where n is the total number of sub-expressions) or more than max sub-expressions are 'true' then the result is 'false'. If one or more sub-expressions are NULL then it is possible that neither of these conditions is satisfied, in which case the operator results in NULL. For example, if min is 3 and max is 4 and the sub-expressions have values {true,true,false,NULL} then the operator results in NULL whereas {true,false,false,NULL} results in false and {true,true,true,NULL} results in 'true'. The result NULL indicates that the correct value for the operator cannot be determined.

qti-any-n Supported Attributes

Name	Usage	Value	Default
min	Required	integer or identifier The minimum number of sub-expressions that must be true.	
max	Required	integer or identifier The maximum number of sub-expressions that must be true.	

QtiBaseValue

The BaseValue (qti-base-value) expression returns a single value from the set defined by the given baseType.

qti-base-value Supported Attributes

Name	Usage	Value	Default
base-type	Required	string Enumerated value set of: { boolean float identifier integer string }	

QtiCorrect

The Correct (qti-correct) expression looks up the declaration of a response variable and returns the associated correctResponse or NULL if no correct value was declared.

qti-correct Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string that resolves to a response variable.	

QtiCustomOperator

The Custom Operator (qti-custom-operator) provides an extension mechanism for defining operations or expressions not currently supported by the QTI 3 specification.

The QTI 3 Player supports two custom operators:

- amp:math.stringToFloat
- amp:ee.isEquivalent

amp:math.stringToFloat

The `amp:math.stringToFloat` operator (`qti-custom-operator class="amp:math.stringToFloat"`) conversion operator takes a single sub-expression which must have single cardinality and base-type string. The result is a value of base type float with the same numeric value. If the sub-expression is NULL or cannot be converted to a float then the operator results in NULL.

qti-custom-operator class="amp:math.stringToFloat" Supported Attributes

Name	Usage	Value	Default
This custom operator has no qti attributes			

amp:ee.isEquivalent

The `amp:ee.isEquivalent` operator (`qti-custom-operator class="amp:ee.isEquivalent"`) is a comparison operator that takes two LaTeX sub-expressions, each of which must have single cardinality and base-type of string, and determines if the LaTeX expressions are numerically equivalent. The LaTeX expressions may be a simple number or a rich expression that can be resolved to a number. As of 12/01/2021 `amp:ee.isEquivalent` implements *numeric equivalency only*.

Examples:

- 1) $\sqrt{\frac{4}{2} \times \frac{8}{4}} - 2.0 + 0.53$ is equivalent to 0.53
- 2) 0.530 is equivalent to 0.53

Known Bugs:

- 1) There is a known parser bug that improperly throws a syntax error on a LaTeX string of .53 (there is no "0" in front of the decimal point).
- 2) Some fractions, e.g., $1/4$ can be improperly transformed during evaluation.
Example, the expression $3 + 1/4$ is improperly transformed to 3.75.

qti-custom-operator class="amp:ee.isEquivalent" Supported Attributes

Name	Usage	Value	Default
definition	Optional	string Must be a valid identifier string that resolves to a response variable.	

@author Paul Grudnitski, amp-up.io

@license - The amp:ee.isEquivalent custom operator software is proprietary software copyright amp-up.io and may not be shared, copied, or distributed without the explicit written consent of amp-up.io.

QtiDefault

The Default (qti-default) expression looks up the declaration of an item variable and returns the associated defaultValue or NULL if no default value was declared.

qti-default Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string that resolves to a variable.	

QtiDelete

The Delete (qti-delete) operator takes two sub-expressions which must both have the same base-type. The first sub-expression must have single cardinality and the second must be a multiple or ordered container. The result is a new container derived from the second sub-expression with all instances of the first sub-expression removed. For example, when applied to A and {B,A,C,A} the result is the container {B,C}. If either sub-expression is NULL the result of the operator is NULL. The restrictions that apply to the qti-member operator also apply to the qti-delete operator.

qti-delete Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiDivide

The Divide (qti-divide) operator takes 2 sub-expressions which both have single cardinality and numerical base-types. The result is a single float that corresponds to the first expression divided by the second expression. If either of the sub-expressions is NULL then the operator results in NULL. Item authors should make every effort to ensure that the value of the second expression is never 0, however, if it is zero or the resulting value is outside the value set defined by float (not including positive and negative infinity) then the operator should result in NULL.

qti-divide Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiEqual

The Equal (qti-equal) operator takes two sub-expressions which must both have single cardinality and have a numerical base-type. The result is a single boolean with a value of true if the two expressions are numerically equal and false if they are not.

If either sub-expression is NULL then the operator results in NULL.

qti-equal Supported Attributes

Name	Usage	Value	Default
tolerance-mode	Optional	string Enumerated value set of: { absolute exact relative } When comparing two floating point numbers for equality it is often desirable to have a tolerance to ensure that spurious errors in scoring are not introduced by rounding errors. The tolerance mode determines whether the comparison is done exactly, using an absolute range or a relative range.	"exact"
tolerance	Optional	toleranceList If the tolerance mode is absolute or relative then the tolerance must be specified. The tolerance consists of two positive numbers, t0 and t1, that define the lower and upper bounds. If only one value is given it is used for both. In absolute mode the result of the comparison is true if the value of the second expression, y is within	

		the following range defined by the first value, x: $x-t_0, x+t_1$. In relative mode, t_0 and t_1 are treated as percentages and the following range is used instead: $x*(1-t_0/100), x*(1+t_1/100)$ Examples of tolerance attribute: tolerance="0.1" tolerance="0.1 0.2"	
include-lower-bound	Optional	boolean Controls whether or not the lower bound is included in the comparison.	true
include-upper-bound	Optional	boolean Controls whether or not the upper bound is included in the comparison.	true

QtiEqualRounded

The EqualRounded (qti-equal-rounded) operator takes two sub-expressions which must both have single cardinality and have a numerical base-type. The result is a single boolean with a value of 'true' if the two expressions are numerically equal after rounding and 'false' if they are not.

If either sub-expression is NULL then the operator results in NULL.

qti-equal-rounded Supported Attributes

Name	Usage	Value	Default
rounding-mode	Required	string Enumerated value set of: { decimalPlaces significantFigures } decimalPlaces - Use a number of decimal places. significantFigures - The number of significant decimal places.	

figures	Required	integer or identifier The number of figures to round to. If rounding-mode is "significantFigures", the value of figures must resolve to a non-zero positive integer. If rounding-mode is "decimalPlaces", the value of figures must resolve to an integer greater than or equal to zero.	
---------	----------	---	--

QtiFieldValue

The FieldValue (qti-field-value) operator takes a sub-expression with a record container value. The result is the value of the field with the specified field-identifier.

If there is no field with that identifier then the result of the operator is NULL.

qti-field-value Supported Attributes

Name	Usage	Value	Default
field-identifier	Required	identifier The identifier of the field to be selected.	

QtiGcd

The Gcd (qti-gcd) operator takes 1 or more sub-expressions which all have base-type integer and may have single, multiple or ordered cardinality. The result is a single integer equal in value to the greatest common divisor (gcd) of the argument values. If all the arguments are zero, the result is 0, $\text{gcd}(0,0)=0$; authors should beware of this in calculations which require division by the gcd of random values. If some, but not all, of the arguments are zero, the result is the gcd of the non-zero arguments, $\text{gcd}(0,n)=n$ if n not equal to 0. If any of the sub-expressions is NULL, the result is NULL.

If any of the sub-expressions is not a numerical value, then the result is NULL.

qti-gcd Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiGt

The Gt (qti-gt) operator takes two sub-expressions which must both have single cardinality and have a numerical base-type. The result is a single boolean with a value of true if the first expression is numerically greater than the second and false if it is less than or equal to the second.

If either sub-expression is NULL then the operator results in NULL.

qti-gt Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiGte

The Gte (qti-gte) operator takes two sub-expressions which must both have single cardinality and have a numerical base-type. The result is a single boolean with a value of true if the first expression is numerically greater than or equal to the second and false if it is less than the second.

If either sub-expression is NULL then the operator results in NULL.

qti-gte Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiIndex

The Index (qti-index) operator takes a sub-expression with an ordered container value and any base-type. The result is the nth value of the container. The result has the same base-type as the sub-expression but single cardinality. The first value of a container has index 1, the second 2 and so on. 'n' must be a positive integer. If 'n' exceeds the number of values in the container (or the sub-expression is NULL) then the result of the index operator is NULL. If 'n' is an identifier, it is the value of 'n' at runtime that is used.

qti-index Supported Attributes

Name	Usage	Value	Default
n	Required	integer or identifier	

		Identification of the index ordinal value required.	
--	--	---	--

QtiIntegerDivide

The IntegerDivide (qti-integer-divide) operator takes 2 sub-expressions which both have single cardinality and base-type integer. The result is the single integer that corresponds to the first expression (x) divided by the second expression (y) rounded down to the greatest integer (i) such that $i \leq (x/y)$. If y is 0, or if either of the sub-expressions is NULL then the operator results in NULL.

qti-integer-divide Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiIntegerModulus

The IntegerModulus (qti-integer-modulus) operator takes 2 sub-expressions which both have single cardinality and base-type integer. The result is the single integer that corresponds to the remainder when the first expression (x) is divided by the second expression (y). If z is the result of the corresponding integerDivide operator then the result is $x - z * y$. If y is 0, or if either of the sub-expressions is NULL then the operator results in NULL.

qti-integer-modulus Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiIntegerToFloat

The IntegerToFloat (qti-integer-to-float) conversion operator takes a single sub-expression which must have single cardinality and base-type integer. The result is a value of base type float with the same numeric value.

If the sub-expression is NULL then the operator results in NULL.

qti-integer-to-float Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtilsNull

The IsNull (qti-is-null) operator takes a sub-expression with any base-type and cardinality. The result is a single boolean with a value of true if the sub-expression is NULL and false otherwise. Note that empty containers and empty strings are both treated as NULL.

qti-is-null Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiLcm

The Lcm (qti-lcm) operator takes 1 or more sub-expressions which all have base-type integer and may have single, multiple or ordered cardinality. The result is a single integer equal in value to the lowest common multiple (lcm) of the argument values. If any argument is zero, the result is 0, $\text{lcm}(0,n)=0$; authors should beware of this in calculations which require division by the lcm of random values. If any of the sub-expressions is NULL, the result is NULL.

If any of the sub-expressions is not a numerical value, then the result is NULL.

qti-lcm Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiLt

The Lt (qti-lt) operator takes two sub-expressions which must both have single cardinality and have a numerical base-type. The result is a single boolean with a value of true if the first expression is numerically less than the second and false if it is greater than or equal to the second.

If either sub-expression is NULL then the operator results in NULL.

qti-lt Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiLte

The Lte (qti-lte) operator takes two sub-expressions which must both have single cardinality and have a numerical base-type. The result is a single boolean with a value of true if the first expression is numerically less than or equal to the second and false if it is greater than the second.

If either sub-expression is NULL then the operator results in NULL.

qti-lte Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiMapResponse

This MapResponse (qti-map-response) expression looks up the value of a response variable and then transforms it using the associated mapping, which must have been declared. The result is a single float. If the response variable has single cardinality then the value returned is simply the mapped target value from the map. If the response variable has multiple or ordered cardinality then the value returned is the sum of the mapped target values. This expression cannot be applied to variables of record cardinality. For example, if a mapping associates the identifiers {A,B,C,D} with the values {0,1,0.5,0} respectively then mapResponse will map the single value 'C' to the numeric value 0.5 and the set of values {C,B} to the value 1.5.

If a container contains multiple instances of the same value then that value is counted once only. To continue the example above {B,B,C} would still map to 1.5 and not 2.5.

qti-map-response Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string that resolves to a response variable.	

QtiMatch

The Match (qti-match) operator takes two sub-expressions which must both have the same base-type and cardinality. The result is a single boolean with a value of true if the two expressions represent the same value and false if they do not. If either sub-expression is NULL then the operator results in NULL.

The match operator must not be confused with broader notions of equality such as numerical equality. To avoid confusion, the match operator should not be used to compare subexpressions with base-types of float and must not be used on sub-expressions with a base-type of duration.

qti-match Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiMathConstant

The MathConstant (qti-math-constant) is a QTI expression function. The result is a mathematical constant returned as a single float, e.g. Pi and e.

qti-math-constant Supported Attributes

Name	Usage	Value	Default
name	Required	string Constant name. Enumerated value set of: { pi e }	

QtiMathOperator

The MathOperator(qti-math-operator) operator takes 1 or more sub-expressions which all have single cardinality and have numerical base-types. The trigonometric functions, sin, cos and tan, take one argument in radians, which evaluates to a single float. Other functions take one numerical argument.

Further functions might take more than one numerical argument, e.g. atan2 (two argument arc tan). The result is a single float, except for the functions signum, floor and ceil, which return a single integer. If any of the sub-expressions is NULL, the result is NULL. If any of the sub-expressions falls outside the natural domain of the function called by qti-math-operator, e.g. log(0) or asin(2), then the result is NULL.

The QTI 3 Player supports all built-in math operator functions:

- acos
- acot
- acsc
- asec
- asin
- atan
- atan2
- cos

- cot
- csc
- sec
- sin
- tan
- cosh
- csch
- sech
- sinh
- tanh
- abs
- ceil
- exp
- floor
- ln
- log
- signum
- toDegrees
- toRadians

qti-math-operator Supported Attributes

Name	Usage	Value	Default
name	Required	string Operator name. Enumerated value set of: { sin cos tan sec csc cot asin acos atan atan2 asec acsc acot sinh cosh tanh sech csch coth log ln exp abs signum floor ceil toDegrees toRadians }	

For a complete description of the operators and their arguments, please refer to [Section 8.11 of the QTI 2.2.2 Information Model](#) document.

QtiMax

The Max (qti-max) operator takes 1 or more sub-expressions which all have numerical base-types and may have single, multiple or ordered cardinality. The result is a single float, or, if all sub-expressions are of integer type, a single integer, equal in value to the greatest of the argument values, i.e. the result is

the argument closest to positive infinity. If the arguments have the same value, the result is that same value. If any of the sub-expressions is NULL, the result is NULL.

If any of the sub-expressions is not a numerical value, then the result is NULL.

qti-max Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiMember

The Member (qti-member) operator takes two sub-expressions which must both have the same base-type. The first sub-expression must have single cardinality and the second must be a multiple or ordered container. The result is a single boolean with a value of true if the value given by the first sub-expression is in the container defined by the second sub-expression. If either sub-expression is NULL then the result of the operator is NULL. The member operator should not be used on sub-expressions with a base-type of float because of the poorly defined comparison of values. It must not be used on sub-expressions with a base-type of duration.

qti-member Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiMin

The Min (qti-min) operator takes 1 or more sub-expressions which all have numerical base-types and may have single, multiple or ordered cardinality. The result is a single float, or, if all sub-expressions are of integer type, a single integer, equal in value to the smallest of the argument values, i.e. the result is the argument closest to negative infinity. If the arguments have the same value, the result is that same value. If any of the sub-expressions is NULL, the result is NULL. If any of the sub-expressions is not a numerical value, then the result is NULL.

qti-min Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiMultiple

The Multiple (qti-multiple) operator takes 0 or more sub-expressions all of which must have either single or multiple cardinality. Although the sub-expressions may be of any base-type they must all be of the same base-type. The result is a container with multiple cardinality containing the values of the sub-expressions, sub-expressions with multiple cardinality have their individual values added to the result: containers cannot contain other containers. For example, when applied to A, B and {C,D} the multiple operator results in {A,B,C,D}. All sub-expressions with NULL values are ignored. If no sub-expressions are given (or all are NULL) then the result is NULL.

qti-multiple Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiNot

This provides the logic 'not' (qti-not) operation on the enclosed expression. The not operator takes a single sub-expression with a base-type of boolean and single cardinality. The result is a single boolean with a value obtained by the logical negation of the sub-expression's value.

If the sub-expression is NULL then the not operator also results in NULL.

qti-not Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiOr

This provides the logic 'or' operation across one or more other operators. The or operator takes one or more sub-expressions each with a base-type of boolean and single cardinality. The result is a single boolean which is 'true' if any of the sub-expressions are 'true' and 'false' if all of them are 'false'.

If one or more sub-expressions are NULL and all the others are false then the operator also results in NULL.

qti-or Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiOrdered

The Ordered (qti-ordered) operator takes 0 or more sub-expressions all of which must have either single or ordered cardinality. Although the sub-expressions may be of any base-type they must all be of the same base-type. The result is a container with ordered cardinality containing the values of the sub-expressions, sub-expressions with ordered cardinality have their individual values added (in order) to the result: contains cannot contain other containers. For example, when applied to A, B, {C,D} the ordered operator results in {A,B,C,D}.

Note that the ordered operator never results in an empty container.

All sub-expressions with NULL values are ignored. If no sub-expressions are given (or all are NULL) then the result is NULL.

qti-ordered Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiPatternMatch

The PatternMatch (qti-pattern-match) expression takes a sub-expression which must have single cardinality and a base-type of string. The result is a single boolean with a value of true if the sub-expression matches the regular expression given by pattern and false if it doesn't. If the sub-expression is NULL then the operator results in NULL.

The syntax for the regular expression language is defined in

<https://www.w3.org/TR/xmlschema-2/#regexs>

qti-pattern-match Supported Attributes

Name	Usage	Value	Default
pattern	Required	identifier or string The syntax for the regular expression language is defined here: https://www.w3.org/TR/xmlschema-2/#regexs May refer to a variable. To resolve this, variable declarations are searched first. If none are found	

		then the pattern must be a Regular Expression string.	
--	--	---	--

QtiPower

The Power (qti-power) operator takes 2 sub-expression which both have single cardinality and numerical base-types. The result is a single float that corresponds to the first expression raised to the power of the second. If either or the sub-expressions is NULL then the operator results in NULL. If the resulting value is outside the value set defined by float (not including positive and negative infinity) then the operator shall result in NULL.

qti-power Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiProduct

The Product (qti-product) operator takes 1 or more sub-expressions which all have numerical base-types and may have single, multiple or ordered cardinality. The result is a single float or, if all sub-expressions are of integer type, a single integer that corresponds to the product of the numerical values of the sub-expressions.

If any of the sub-expressions are NULL then the operator results in NULL.

qti-product Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiRandom

The Random (qti-random) operator takes a sub-expression with a multiple or ordered container value and any base-type. The result is a single value randomly selected from the container. The result has the same base-type as the sub-expression but single cardinality.

If the sub-expression is NULL then the result is also NULL.

qti-random Supported Attributes

Name	Usage	Value	Default
------	-------	-------	---------

This element has no qti attributes

QtiRandomFloat

The RandomFloat (qti-random-float) operator selects a random float from the specified range [min,max].

qti-random-float Supported Attributes

Name	Usage	Value	Default
min	Optional	float or identifier The lower bound for the random number generation. May be a variable identifier that resolves to a float value.	"0"
max	Required	float or identifier The upper bound for the random number generation. May be a variable identifier that resolves to a float value.	

QtiRandomInteger

The RandomInteger (qti-random-integer) operator selects a random integer from the specified range [min,max] satisfying $\text{min} + \text{step} * n$ for some integer n .

* For example, with min=2, max=11 and step=3 the values {2,5,8,11} are possible.

qti-random-integer Supported Attributes

Name	Usage	Value	Default
min	Optional	integer or identifier The lower bound for the random number generation. May be a variable identifier that resolves to an integer value.	"0"
max	Required	integer or identifier	

		The upper bound for the random number generation. May be a variable identifier that resolves to an integer value.	
step	Optional	integer or identifier The positive integer step value for the random number generation. May be a variable identifier that resolves to a positive integer value.	"1"

QtiRepeat

The Repeat (qti-repeat) operator takes 0 or more sub-expressions, all of which must have either single or ordered cardinality and the same baseType. The result is an ordered container having the same baseType as its sub-expressions. The container is filled sequentially by evaluating each sub-expression in turn and adding the resulting single values to the container, iterating this process numberRepeats times in total. If number-repeats refers to a variable whose value is less than 1, the value of the whole expression is NULL.

Any sub-expressions evaluating to NULL are ignored. If all sub-expressions are NULL then the result is NULL.

qti-random-float Supported Attributes

Name	Usage	Value	Default
number-repeats	Required	integer or identifier Defines the number of iterations applied to the processing of the contents. May be an integer or a variable identifier.	

QtiRound

The Round (qti-round) operator takes a single sub-expression which must have single cardinality and a numerical base-type. The result is a value of base-type integer formed by rounding the value of the sub-expression. The result is the integer n for all input values in the range $[n-0.5, n+0.5)$. In other words, 6.8 and 6.5 both round up to 7, 6.49 rounds down to 6 and -6.5 rounds up to -6.

If the sub-expression is NULL then the operator results in NULL. If the sub-expression is NaN, then the result is NULL. If the sub-expression is INF, then the result is INF. If the sub-expression is -INF, then the result is -INF.

qti-round Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiRoundTo

The RoundTo (qti-round-to) operator takes one sub-expression which must have single cardinality and a numerical base-type. The result is a single float with the value nearest to that of the expression's value such that when converted to a decimal string it represents the expression rounded by the specified rounding method to the specified precision. If the sub-expression is NULL, then the result is NULL.

If the sub-expression is INF, then the result is INF. If the sub-expression is -INF, then the result is -INF. If the argument is NaN, then the result is NULL. When rounding to n significant figures, the deciding digit is the (n+1)th digit counting from the first non-zero digit from the left in the number. If the deciding digit is 5 or greater, the nth digit is increased by 1 and all digits to its right are discarded; if the deciding digit is less than 5, all digits to the right of the nth digit are discarded. When rounding to n decimal places, the deciding digit is the (n+1)th digit counting to the right from the decimal point. If the deciding digit is 5 or greater, the nth digit is increased by 1 and all digits to its right are discarded; if the deciding digit is less than 5, all digits to the right of the nth digit are discarded.

qti-round-to Supported Attributes

Name	Usage	Value	Default
rounding-mode	Required	string Enumerated value set of: { decimalPlaces significantFigures } decimalPlaces - Use a number of decimal places. significantFigures - The number of significant decimal places.	
figures	Required	integer or identifier The number of figures to round to. If	

		rounding-mode="significantFigure s", the value of figures must be a non-zero positive integer. If rounding-mode="decimalPlaces", the value of figures must be an integer greater than or equal to zero. May be a variable identifier that resolves to an integer value.	
--	--	---	--

QtiStatsOperator

The StatsOperator (qti-stats-operator) operator takes 1 sub-expression which is a container of multiple or ordered cardinality and has a numerical base-type. The result is a single float. If the sub-expression or any value contained therein is NULL, the result is NULL. If any value contained in the sub-expression is not a numerical value, then the result is NULL.

The QTI 3 Player supports all built-in stats operator functions:

- mean
- sampleVariance
- sampleSD
- popVariance
- popSD

qti-stats-operator Supported Attributes

Name	Usage	Value	Default
name	Required	string Operator name. Enumerated value set of: { mean sampleVariance sampleSD popVariance popSD }	

For a complete description of the operators and their arguments, please refer to [Section 8.18 of the QTI 2.2.2 Information Model](#) document.

QtiStringMatch

The StringMatch (qti-string-match) operator takes two sub-expressions which must have single cardinality and a base-type of string. The result is a single boolean with a value of true if the two strings match according to the comparison rules defined by the attributes below and false if they don't.

If either sub-expression is NULL then the operator results in NULL.

qti-string-match Supported Attributes

Name	Usage	Value	Default
case-sensitive	Required	boolean Used to control whether or not the matching is case sensitive. If 'true' then the matching is case sensitive and, for example, "SHell" is not a match of "Shell". If 'false' then the match is not case sensitive and "SHell" is a match of "Shell".	

QtiSubstring

The Substring (qti-substring) operator takes two sub-expressions which must both have an effective base-type of string and single cardinality. The result is a single boolean with a value of true if the first expression is a substring of the second expression and false if it isn't.

If either sub-expression is NULL then the result of the operator is NULL.

qti-substring Supported Attributes

Name	Usage	Value	Default
case-sensitive	Required	boolean Used to control whether or not the substring is matched case sensitively.	

		If 'true' then the match is case sensitive and, for example, "Hell" is not a substring of "Shell". If 'false' then the match is not case sensitive and "Hell" is a substring of "Shell".	
--	--	--	--

QtiSubtract

The Subtract (qti-subtract) operator takes 2 sub-expressions which all have single cardinality and numerical base-types. The result is a single float or, if both sub-expressions are of integer type, a single integer that corresponds to the first value minus the second.

If either of the sub-expressions is NULL then the operator results in NULL.

qti-subtract Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiSum

The Sum (qti-sum) operator provides the 'sum' function on the enclosed set of expressions. The sum operator takes 1 or more sub-expressions which all have numerical base-types and may have single, multiple or ordered cardinality.

The result is a single float or, if all sub-expressions are of integer type, a single integer that corresponds to the sum of the numerical values of the sub-expressions. If any of the sub-expressions are NULL then the operator results in NULL.

qti-sum Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiTruncate

The Truncate (qti-truncate) operator takes a single sub-expression which must have single cardinality and a numerical base-type. The result is a value of base-type integer formed by truncating the value of the sub-expression towards zero.

For example, the value 6.8 becomes 6 and the value -6.8 becomes -6. If the sub-expression is NULL then the operator results in NULL. If the sub-expression is NaN, then the result is NULL. If the sub-expression is INF, then the result is INF. If the sub-expression is -INF, then the result is -INF.

qti-truncate Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiVariable

The Variable (qti-variable) expression looks up the value of an itemVariable that has been declared in a corresponding variableDeclaration or is one of the built-in variables. The result has the base-type and cardinality declared for the variable.

qti-variable Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier Must be a valid identifier string that resolves to a variable declaration.	

Template and Feedback Elements

QtiTemplateBlock

The TemplateBlock (qti-template-block) element is used to define the block content structures that are available for the creation of Item templates. A TemplateBlock must not contain any interactions, either directly or indirectly.

qti-template-block Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier The identifier attribute, in conjunction with the show-hide attribute, determines the visibility of the template.	
template-identifier	Required	identifier Identifier of a template variable that must have a base-type of identifier and be of either single or multiple cardinality.	
show-hide	Optional	string Determines how the visibility of the qti-template-inline is controlled. If set to 'show' then the template is hidden by default and shown only if the associated template variable matches, or contains, the value of the identifier attribute. If set to 'hide' then the template is shown by default and hidden if the associated template variable matches, or contains, the value of the identifier attribute.	"show"

QtiTemplateInline

The TemplateInline (qti-template-inline) element is used to define the inline content structures that are available for the creation of Item templates. A TemplateInline must not contain any interactions, either directly or indirectly.

qti-template-inline Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier The identifier attribute, in conjunction with the show-hide attribute, determines the visibility of the template.	
template-identifier	Required	identifier Identifier of a template variable that must have a base-type of identifier and be of either single or multiple cardinality.	
show-hide	Optional	string Determines how the visibility of the qti-template-inline is controlled. If set to 'show' then the template is hidden by default and shown only if the associated template variable matches, or contains, the value of the identifier attribute. If set to 'hide' then the template is shown by default and hidden if the associated template variable matches, or contains, the value of the identifier attribute.	"show"

QtiFeedbackBlock

The FeedbackBlock (qti-feedback-block) element is used to define Item Body block feedback content that can be presented to the learner. A feedback element that forms part of a non-adaptive Item must not contain any interactions, either directly or indirectly. In an adaptive item, when an interaction is contained in a hidden feedback element it must also be hidden. The candidate must not be able to set or update the value of the associated response variables. Feedback elements can be embedded inside each other, with one exception: qti-feedback-inline cannot contain qti-feedback-block elements.

qti-feedback-block Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier The identifier attribute, in conjunction with the show-hide attribute, determines the visibility of the feedback.	
outcome-identifier	Required	identifier Identifier of an outcome variable that must have a base-type of identifier and be of either single or multiple cardinality.	
show-hide	Optional	string Determines how the visibility of the feedback is controlled. If set to 'show' then the feedback is hidden by default and shown only if the associated outcome variable matches, or contains, the value of the identifier attribute. If set to 'hide' then the template is shown by default and hidden if the associated outcome variable matches, or contains, the value of the identifier attribute.	"show"

QtiFeedbackInline

The FeedbackInline (qti-feedback-inline) element is used to define Item Body inline feedback content that can be presented to the learner. A feedback element that forms part of a non-adaptive Item must not contain any interactions, either directly or indirectly. In an adaptive item, when an interaction is contained in a hidden feedback element it must also be hidden. The candidate must not be able to set or update the value of the associated response variables. Feedback elements can be embedded inside each other, with one exception: qti-feedback-inline cannot contain qti-feedback-block elements.

qti-feedback-inline Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier The identifier attribute, in conjunction with the show-hide attribute, determines the visibility of the feedback.	
outcome-identifier	Required	identifier Identifier of an outcome variable that must have a base-type of identifier and be of either single or multiple cardinality.	
show-hide	Optional	string Determines how the visibility of the feedback is controlled. If set to 'show' then the feedback is hidden by default and shown only if the associated outcome variable matches, or contains, the value of the identifier attribute. If set to 'hide' then the template is shown by default and hidden if the associated outcome variable matches, or contains, the value of the identifier attribute.	"show"

QtiModalFeedback

The ModalFeedback (qti-modal-feedback) element is shown to the candidate - in the form of a modal dialog - directly following response processing. The value of an outcome variable is used in conjunction with the show-hide and identifier attributes to determine whether or not the feedback is shown. The content of the ModalFeedback must not contain any interactions.

qti-modal-feedback Supported Attributes

Name	Usage	Value	Default
identifier	Required	identifier The identifier attribute, in conjunction with the show-hide attribute, determines the visibility of the feedback.	
outcome-identifier	Required	identifier Identifier of an outcome variable that must have a base-type of identifier and be of either single or multiple cardinality.	
show-hide	Optional	string Determines how the visibility of the feedback is controlled. If set to 'show' then the feedback is hidden by default and shown only if the associated outcome variable matches, or contains, the value of the identifier attribute. If set to 'hide' then the template is shown by default and hidden if the associated outcome variable matches, or contains, the value of the identifier attribute.	"show"
title	Optional	string If the title is specified, QTI 3 Player displays the title in the header of the modal dialog.	

QtiContentBody

A ContentBody (qti-content-body) element is a container for content inside of ModalFeedback.

qti-content-body Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

Catalog Elements

QtiCatalogInfo

A CatalogInfo (qti-catalog-info) element is a container that holds one or more catalogs and their content. Content inside CatalogInfo is considered "dormant" and is not included for delivery to candidates by default. A candidate's Personal Needs and Preferences (PNP) profile (or assessment program settings) will indicate whether the candidate should be presented with any of the possible supports included within the CatalogInfo element.

qti-catalog-info Supported Attributes

Name	Usage	Value	Default
This element has no qti attributes			

QtiCatalog

A Catalog (qti-catalog) element is a container of content or resource references that is outside the content body node. A catalog holds support-specific dormant content that can be made active (a part of the perceivable content presented to the candidate) based on the candidate's PNP information (or an assessment program's settings). A catalog is referenced from a specific portion of the content body by a specific, unique identifier, which matches the catalog's identifier. A catalog contains one or more "cards", each of which address a specific support/feature.

qti-catalog Supported Attributes

Name	Usage	Value	Default
id	Required	String The unique identifier used to locate the catalog from a data-catalog-idref attribute.	

QtiCard

A Card (qti-card) element is a data structure within a catalog which contains dormant HTML content or a resource reference for a specific support/feature. A card may also contain multiple CardEntry containers. For example, you might have multiple CardEntry nodes for different language versions of a particular support. The qti-card element is a container for the following elements:

- qti-card-entry
- qti-html-content

- qti-file-href

qti-card Supported Attributes

Name	Usage	Value	Default
support	Required	String This attribute names either pre-defined supports or a custom-named support (via the “ext:” prefix). The named support will indicate for whom the dormant content is intended. Use only one card for any particular named support in a catalog. Examples: “glossary-on-screen” “keyword-translation” “ext:sbac-glossary-illustration”	
xml:lang	Optional	String An ISO-639 language code. Current supported codes from the set of: { ar cmn de en es fr hmn it ja ko my nl pa ru so tl uk vi yue zh }	Empty string “”

QtiCardEntry

A CardEntry (qti-card-entry) element is a content container within a catalog card. Each instance of a CardEntry provides a different resource where an attribute (often custom attributes) on the CardEntry element declares the difference between the resources, and where the attribute value aligns with a specific preference/need from the candidate's PNP (or an assessment program's settings). For example, there could be multiple CardEntry nodes for different language versions for a particular support.

The qti-cardelement element is a container for the following elements:

- qti-html-content
- qti-file-href

qti-card-entry Supported Attributes

Name	Usage	Value	Default
default	Optional	String An attribute that denotes whether the particular card is the default content for the support (listed in the Card attribute). If the CardEntry attribute values do not identify the proper content for a candidate, use the content designated as default. Only one of the CardEntry instances can have a default designation.	“false”
xml:lang	Optional	String An ISO-639 language code. Current supported codes from the set of: { ar cmn de en es fr hmn it ja ko my nl pa ru so tl uk vi yue zh }	Empty string “”

QtiHtmlContent

An HtmlContent (qti-html-content) element is a container for HTML content inside of a Card or CardEntry element.

qti-html-content Supported Attributes

Name	Usage	Value	Default
This element has no supported qti attributes			

QtiFileRef

A FileRef (qti-file-ref) element is a container for the URI for associated file content inside of a Card or CardEntry element.

qti-file-ref Supported Attributes

Name	Usage	Value	Default
------	-------	-------	---------

mime-type	Required	mime-type string; e.g, "audio/mp3" The type of file i.e., content, using the mime-type vocabulary.	
-----------	----------	---	--

Example of QtiCatalogInfo

This example demonstrates a QtiCatalog that provides supplemental Glossary, Keyword Translation, and ext:sbac-glossary-illustration content for one term (“accurate” - data-catalog-idref=“ae029”).

```

0001 <qti-assessment-item ... xml:lang="en">
0002 ...
0003 <qti-item-body>
0004 ...
0005 <p>Indicate which of the following statements are <span
0006 data-catalog-idref="ae029">accurate.</span></p>
0007 ...
0008 </qti-item-body>
0009
0010 <qti-catalog-info>
0011   <qti-catalog identifier="ae029">
0012     <qti-card support="keyword-translation">
0013       <!-- This card entry supports the Spanish keyword translation of ae029 -->
0014       <qti-card-entry xml:lang="es">
0015         <qti-html-content>
0016           <div>
0017             <span>preciso</span>
0018             <audio controls="controls">
0019               <source src="translations/es/preciso.mp3" type="audio/mpeg" />
0020             </audio>
0021           </div>
0022         </qti-html-content>
0023       </qti-card-entry>
0024       <!-- This card entry supports the German keyword translation of ae029 -->
0025       <qti-card-entry xml:lang="de">
0026         <qti-html-content>
0027           <div>
0028             <span>genau</span>
0029             <audio controls="controls">
0030               <source src="translations/de/genau.mp3" type="audio/mpeg" />
0031             </audio>
0032           </div>
0033         </qti-html-content>
0034       </qti-card-entry>
0035     </qti-card>
0036     <!-- This card supports the standard glossary-on-screen support -->
0037     <qti-card support="glossary-on-screen">
0038       <qti-html-content>
0039         <span>correct, exact, and without mistakes</span>
0040       </qti-html-content>
0041     </qti-card>
0042     <!-- This card supports the custom support: ext:sbac-glossary-illustration -->
0043     <qti-card support="ext:sbac-glossary-illustration">
0044       <qti-html-content>
0045         
0046       </qti-html-content>
0047     </qti-card>
0048   </qti-catalog>
0049 </qti-catalog-info>
0050 </qti-assessment-item>

```

Annotated Item XML Examples

This section of the Reference contains annotated examples to assist content authors when constructing QTI 3 item solutions for so-called, “TDX - baseline”, “TDX - showex”, and “TDX - solve” items.

There are many ways to construct such solutions. These examples provide patterns and guidance for doing so. However, the assumption is that, over time, new and improved patterns will evolve out of these initial solutions.

TDX - baseline

A TDX “Baseline” item is the simplest form of TDX as it contains no adaptivity (adaptive=”false”).

The QTI 3 XML in this example also implements Template Processing and template variables, thus enabling a virtually-unlimited number of combinations of the template variables and enabling virtually-unlimited numbers of “Baseline” combinations of this item template.

```

0001 <qti-assessment-item
0002   xmlns="http://www.imsglobal.org/xsd/imsqtiassi_v3p0"
0003   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
0004   xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqtiassi_v3p0
0005   https://purl.imsglobal.org/spec/qti/v3p0/schema/xsd/imsqti_asiv3p0_v1p0.xsd"
0006   identifier="tdx-baseline-01" title="Compute the Sales Tax - baseline"
0007   adaptive="false" time-dependent="false">
0008
0009   <!-- The text entry interaction has a base-type of string -->
0010   <qti-response-declaration identifier="RESPONSE"
0011     base-type="string" cardinality="single"/>
0012
0013   <!-- Define a variable to be used to store a correct float representation of the
0014     RESPONSE variable -->
0015   <qti-response-declaration identifier="RESPONSE_FLOAT"
0016     base-type="float" cardinality="single"/>
0017
0018   <!-- Define a variable to bind to the End Attempt Interaction mxl-controller-bar -->
0019   <qti-response-declaration identifier="EndAttempt"
0020     base-type="boolean" cardinality="single" />
0021
0022   <qti-outcome-declaration identifier="SCORE"
0023     base-type="float" cardinality="single"
0024     normal-maximum="1" normal-minimum="0">
0025     <qti-default-value>
0026       <qti-value>0</qti-value>
0027     </qti-default-value>
0028   </qti-outcome-declaration>
0029
0030   <!-- Define a feedback variable; its baseType is "identifier" so that it can contain the
0031     identifier of the feedback message -->
0032   <qti-outcome-declaration identifier="FEEDBACK"
0033     base-type="identifier" cardinality="single"/>
0034
0035   <!-- Define a variable used to display the student's response in feedback -->
0036   <qti-outcome-declaration identifier="FEEDBACK_RESPONSE"
0037     base-type="string" cardinality="single"/>
0038
0039   <qti-template-declaration identifier="SALESPRICE"

```

```

0049         cardinality="single" base-type="float"/>
0041
0042 <qti-template-declaration identifier="TAX_PERCENT"
0043         cardinality="single" base-type="float"/>
0044
0045 <qti-template-declaration identifier="SALESTAX"
0046         cardinality="single" base-type="float"/>
0047
0048 <qti-template-declaration identifier="TOTAL"
0049         cardinality="single" base-type="float"/>
0050
0051 <qti-template-processing>
0052   <qti-set-template-value identifier="SALESPRICE">
0053     <qti-integer-to-float>
0054       <qti-random-integer min="2" max="99" step="1"/>
0055     </qti-integer-to-float>
0056   </qti-set-template-value>
0057   <qti-set-template-value identifier="TAX_PERCENT">
0058     <qti-divide>
0059       <qti-random-integer min="300" max="700" step="25"/>
0060       <qti-base-value base-type="float">100</qti-base-value>
0061     </qti-divide>
0062   </qti-set-template-value>
0063   <qti-set-template-value identifier="SALESTAX">
0064     <qti-divide>
0065       <qti-variable identifier="TAX_PERCENT"/>
0066       <qti-base-value base-type="float">100</qti-base-value>
0067     </qti-divide>
0068   </qti-set-template-value>
0069   <qti-set-template-value identifier="TOTAL">
0070     <qti-round-to rounding-mode="decimalPlaces" figures="2">
0071       <qti-product>
0072         <qti-variable identifier="SALESTAX"/>
0073         <qti-variable identifier="SALESPRICE"/>
0074       </qti-product>
0075     </qti-round-to>
0076   </qti-set-template-value>
0077   <qti-set-correct-response identifier="RESPONSE_FLOAT">
0078     <qti-variable identifier="TOTAL"/>
0079   </qti-set-correct-response>
0080 </qti-template-processing>
0081
0082 <qti-item-body>
0083   <div class="qti3-player-item-card-bordered-rounded qti3-player-item-card-raised-rounded">
0084     <div class="qti3-player-item-card-body qti-padding-2">
0085       <div class="qti-layout-row">
0086         <div class="qti-layout-col8">
0087           <p>Find the sales tax.</p>
0088           <table class="table table-bordered">
0089             <thead>
0090               <tr>
0091                 <th class="qti-align-center" colspan="3">Sales Tax</th>
0092               </tr>
0093               <tr>
0094                 <th scope="col" class="qti-align-center qti-width-1-3">Selling Price</th>
0095                 <th scope="col" class="qti-align-center qti-width-1-3">Rate of Sales Tax</th>
0096                 <th scope="col" class="qti-align-center qti-width-1-3">Sales Tax</th>
0097               </tr>
0098             </thead>
0099             <tbody>
0100               <tr>
0101                 <td class="qti-align-center">
0102                   $<qti-printed-variable identifier="SALESPRICE" format="%.2f" />
0103                 </td>

```

```

0104         <td class="qti-align-center">
0105             <qti-printed-variable identifier="TAX_PERCENT"/>%
0106         </td>
0107     </tr>
0108 </tbody>
0109 </table>
0110 <hr />
0111 </div>
0112 </div>
0113 <p>
0114     The sales tax is $ <qti-text-entry-interaction class="qti-input-width-6"
0115     pattern-mask="([0-9.\-]{0,8})" response-identifier="RESPONSE" /> .
0116 </p>
0117 </div> <!-- /card-body -->
0118
0119 <div class="qti3-player-item-card-footer qti-height-14 qti-padding-2">
0120     <qti-end-attempt-interaction response-identifier="EndAttempt"
0121     title="Check Answer"
0122     class="endattempt-controller-bar"
0123     data-steps="1"
0124     data-hideprogress="true"
0125     data-hastemplates="true"/>
0126 </div> <!-- /card-footer -->
0127
0128 </div> <!-- /card -->
0129 </qti-item-body>
0130
0131 <qti-response-processing>
0132     <!-- Save off the response so we can echo it back to the student in feedback - if needed -->
0133     <qti-set-outcome-value identifier="FEEDBACK_RESPONSE">
0134         <qti-variable identifier="RESPONSE"/>
0135     </qti-set-outcome-value>
0136 <qti-response-condition>
0137     <!-- Evaluate the response -->
0138     <qti-response-if>
0139         <qti-is-null>
0140             <qti-variable identifier="RESPONSE"/>
0141         </qti-is-null>
0142         <qti-set-outcome-value identifier="FEEDBACK">
0143             <qti-base-value base-type="identifier">incorrect</qti-base-value>
0144         </qti-set-outcome-value>
0145     </qti-response-if>
0146     <qti-response-else-if>
0147         <!-- Check numeric equivalence -->
0148         <qti-equal tolerance-mode="exact">
0149             <!-- The value of RESPONSE is compared with the correct value saved in
0150             the RESPONSE_FLOAT declaration-->
0151             <qti-custom-operator class="amp:math.stringToFloat">
0152                 <qti-variable identifier="RESPONSE"/>
0153             </qti-custom-operator>
0154             <qti-correct identifier="RESPONSE_FLOAT"/>
0155         </qti-equal>
0156
0157         <!-- Response is numerically equivalent to the correct response. Now check form. -->
0158         <qti-response-condition>
0159             <qti-response-if>
0160                 <!-- Does the string have a form of *0.00 ? -->
0161                 <qti-pattern-match pattern="[0-9]*\.[0-9]{2}">
0162                     <qti-variable identifier="RESPONSE"/>
0163                 </qti-pattern-match>
0164                 <qti-set-outcome-value identifier="SCORE">
0165                     <qti-base-value base-type="float">1</qti-base-value>
0166                 </qti-set-outcome-value>

```

```

0167     <qti-set-outcome-value identifier="FEEDBACK">
0168     <qti-base-value base-type="identifier">correct</qti-base-value>
0169     </qti-set-outcome-value>
0170 </qti-response-if>
0171 <qti-response-else>
0172     <qti-set-outcome-value identifier="FEEDBACK">
0173     <qti-base-value base-type="identifier">wrongformat</qti-base-value>
0174     </qti-set-outcome-value>
0174 </qti-response-else>
0175 </qti-response-condition>
0176 </qti-response-else-if>
0177 <qti-response-else>
0178     <!-- Depending on whether the input matches the correct answer or not, FEEDBACK is
0179         given the value of the identifier of the appropriate feedback message -->
0180     <qti-set-outcome-value identifier="FEEDBACK">
0181     <qti-base-value base-type="identifier">incorrect</qti-base-value>
0182     </qti-set-outcome-value>
0183 </qti-response-else>
0184 </qti-response-condition>
0185 </qti-response-processing>
0186
0187 <!-- Note how the identifiers in the following modalFeedback elements match those
0188       of the setOutcomeValue elements above -->
0189 <qti-modal-feedback outcome-identifier="FEEDBACK" show-hide="show" identifier="correct">
0190 <qti-content-body>
0191     <div>Well done! Correct.</div>
0192 </qti-content-body>
0193 </qti-modal-feedback>
0194
0195 <qti-modal-feedback outcome-identifier="FEEDBACK" show-hide="show" identifier="wrongformat">
0196 <qti-content-body>
0197     <div>
0198     <p>
0199         Almost correct. You entered the correct sales tax amount, but in
0200         the <em>wrong format</em>.
0201     </p>
0202     </div>
0203 </qti-content-body>
0204 </qti-modal-feedback>
0205
0206 <qti-modal-feedback outcome-identifier="FEEDBACK" show-hide="show" identifier="incorrect">
0207 <qti-content-body>
0208     <div>Sorry. Incorrect.</div>
0209 </qti-content-body>
0210 </qti-modal-feedback>
0211 </qti-assessment-item>

```

1. **Line 0007** - This item is not adaptive (adaptive="false"). An End Attempt action triggered by a click on "Check Answer" terminates the candidate's item session and results in computation of the SCORE and feedback (if any).
2. **Lines 0015-0016** - Declare a float variable used to store the correct numeric value of the sales tax amount computed by the Template Processing block for this template.
3. **Line 0051** - Start of Template Processing block. Template processing computes the values of four template variables and the correct response for the template variables.

4. **Line 0054** - Use `QtiRandomInteger` to generate a `SALESPRICE` variable value between \$2 dollars and \$99 dollars with \$1 dollar increments.
5. **Line 0059** - Use `QtiRandomInteger` to generate a `SALES_TAX` variable value between 3.00% and 7.00%, with 0.25% increments.
6. **Line 0070** - Use `QtiRoundTo` (decimal places = 2) to generate the `TOTAL` amount rounded to 2 decimal places.
7. **Lines 0077-0079** - Set the correct response from the template processing computations.
8. **Line 0102** - Display the `SALESPRICE` template variable with 2 decimal places with `QtiPrintedVariable` and `format="%.2f"`.
9. **Line 0105** - Display the `TAX_PERCENT` template variable with `QtiPrintedVariable`.
10. **Lines 0114-0115** - `TextEntryInteraction` (simple FIB) has a pattern-mask which only permits the candidate to enter digits, a minus sign ("-"), or a period ((".")), with a maximum of 8 characters. This prevents runaway responses, responses with spaces, and it also prevents candidates from entering other operators or complex expressions which are unwanted in this particular item.
11. **Lines 0120-0125** - Configure the `QtiEndAttempt` `endattempt-controller-bar`.

`title="Check Answer"` - display the text "Check Answer" on the EndAttempt action button.
`data-hastemplates="true"` - results in the display of the "New Question" template button.
`data-hideprogress="true"` - results in hiding of the Progress Stepper.
`data-steps="1"` - On non-adaptive items this should always be set to "1".

12. **Line 0148** - Check whether or not the student's response (in the `RESPONSE` variable) is numerically equal to the correct response that was generated in template processing and saved in the `RESPONSE_FLOAT` variable on **Lines 0077-0079**. If the `RESPONSE` is *not* numerically equal, set the modal feedback identifier to "incorrect".
13. **Line 0151** - Use the `CustomOperator` `amp:math.stringToFloat` to convert the `RESPONSE` string to a float in order to check numeric equivalence on **Line 0148**.
14. **Line 0161** - If we get here, the `RESPONSE` is numerically equal. Now check the pattern of the string `RESPONSE` to determine whether or not there are the proper 2 digits after the decimal point for currency expressions. Use `QtiPatternMatch` with `pattern="[0-9]*\.[0-9]{2}"`.
15. **Lines 0164-0169** - If the `RESPONSE` has a proper format (we already determined that the float value of `RESPONSE` is numerically equal to the correct response), award a `SCORE` of 1, and `FEEDBACK` of

“correct”.

16. **Lines 0172-0174** - The RESPONSE was numerically equivalent, but improperly formatted. Set FEEDBACK to “wrongformat”.
17. **Lines 0189-0193** - Definition of the 100% correct modal feedback element which is displayed at the end of response processing when the FEEDBACK outcome variable is set to “correct”.
18. **Lines 0195-0204** - Definition of the partially correct modal feedback element which is displayed at the end of response processing when the FEEDBACK outcome variable is set to “wrongformat”.
19. **Lines 0206-0211** - Definition of the modal feedback element which is displayed on numerically incorrect responses at the end of response processing when the FEEDBACK outcome variable is set to “incorrect”.

TDX - showex

Unlike the TDX “Baseline” item example above, a TDX “Show an Example” item utilizes the “adaptive” Feedback features built into QTI and into the QTI 3 Player component itself.

The QTI 3 XML in this example also implements Template Processing and template variables, thus enabling a virtually-unlimited number of combinations of the template variables and enabling virtually-unlimited numbers of examples of this item template.

```

0001 <qti-assessment-item
0002   xmlns="http://www.imsglobal.org/xsd/imsqtiasi v3p0"
0003   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
0004   xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqtiasi v3p0
0005   https://purl.imsglobal.org/spec/qti/v3p0/schema/xsd/imsqti_asiv3p0_v1p0.xsd"
0006   identifier="tdx-showex-01" title="Compute the Sales Tax - showex"
0007   adaptive="true" time-dependent="false">
0008
0009   <!-- Define a variable to bind to the End Attempt Interaction mxl-controller-bar -->
0010   <qti-response-declaration identifier="EndAttempt"
0011     base-type="boolean" cardinality="single" />
0012
0013   <!-- This item is not scored, but we declare SCORE out of habit -->
0014   <qti-outcome-declaration identifier="SCORE"
0015     base-type="float" cardinality="single">
0016     <qti-default-value>
0017       <qti-value>0</qti-value>
0018     </qti-default-value>
0019   </qti-outcome-declaration>
0020
0021   <!-- Define a STEP variable to keep track of our current step in the adaptive item.
0022     Initialize STEP to a value of 1 (first step) -->
0023   <qti-outcome-declaration identifier="STEP"
0024     base-type="integer" cardinality="single">
0025     <qti-default-value>
0026       <qti-value>1</qti-value>
0027     </qti-default-value>
0028   </qti-outcome-declaration>
0029
0030   <!-- Define a feedback variable named BODY. Its cardinality is "multiple" so that it can
0031     contain the identifiers of several feedback elements and, in this case, it is initialized
0032     to the identifier "step1" of the first feedback element of this adaptive item. -->
0033   <qti-outcome-declaration identifier="BODY"
0034     base-type="identifier" cardinality="multiple">
0035     <qti-default-value>
0036       <qti-value>step1</qti-value>
0037     </qti-default-value>
0038   </qti-outcome-declaration>
0039
0040   <qti-template-declaration identifier="SALESPRICE"
0041     cardinality="single" base-type="float"
0042     math-variable="false" param-variable="false"/>
0043
0044   <qti-template-declaration identifier="TAX_PERCENT"
0045     cardinality="single" base-type="float"
0046     math-variable="true" param-variable="false"/>
0047
0048   <qti-template-declaration identifier="SALESTAX"
0049     cardinality="single" base-type="float"
0050     math-variable="false" param-variable="false"/>
0051
0052   <qti-template-declaration identifier="TOTAL"

```

```

0053             cardinality="single" base-type="float"
0054             math-variable="false" param-variable="false"/>
0055
0056 <qti-template-processing>
0057   <qti-set-template-value identifier="SALESPRICE">
0058     <qti-integer-to-float>
0059       <qti-random-integer min="2" max="99" step="1"/>
0060     </qti-integer-to-float>
0061   </qti-set-template-value>
0062   <qti-set-template-value identifier="TAX_PERCENT">
0063     <qti-divide>
0064       <qti-random-integer min="300" max="700" step="25"/>
0065       <qti-base-value base-type="float">100</qti-base-value>
0066     </qti-divide>
0067   </qti-set-template-value>
0068   <qti-set-template-value identifier="SALESTAX">
0069     <qti-divide>
0070       <qti-variable identifier="TAX_PERCENT"/>
0071       <qti-base-value base-type="float">100</qti-base-value>
0072     </qti-divide>
0073   </qti-set-template-value>
0074   <qti-set-template-value identifier="TOTAL">
0075     <qti-round-to rounding-mode="decimalPlaces" figures="2">
0076       <qti-product>
0077         <qti-variable identifier="SALESTAX"/>
0078         <qti-variable identifier="SALESPRICE"/>
0079       </qti-product>
0080     </qti-round-to>
0081   </qti-set-template-value>
0082 </qti-template-processing>
0083
0084 <qti-item-body>
0085   <div class="qti3-player-item-card-bordered-rounded qti3-player-item-card-raised-rounded">
0086     <div class="qti3-player-item-card-body qti-padding-2">
0087       <div class="qti-layout-row">
0088         <div class="qti-layout-col8">
0089           <p>Find the sales tax.</p>
0090           <table class="table table-bordered">
0091             <thead>
0092               <tr>
0093                 <th class="qti-align-center" colspan="3">Sales Tax</th>
0094               </tr>
0095               <tr>
0096                 <th scope="col" class="qti-align-center qti-width-1-3">Selling Price</th>
0097                 <th scope="col" class="qti-align-center qti-width-1-3">Rate of Sales Tax</th>
0098                 <th scope="col" class="qti-align-center qti-width-1-3">Sales Tax</th>
0099               </tr>
0100             </thead>
0101             <tbody>
0102               <tr>
0103                 <td class="qti-align-center">
0104                   $<qti-printed-variable identifier="SALESPRICE" format="%.2f" />
0105                 </td>
0106                 <td class="qti-align-center">
0107                   <qti-printed-variable identifier="TAX_PERCENT"/>%
0108                 </td>
0109                 <td class="qti-align-center">?</td>
0110               </tr>
0111             </tbody>
0112           </table>
0113           <hr />
0114         </div>
0115       </div>

```

```

0116 <-- This Feedback is displayed if the BODY variable contains the identifier "step1".
0117     Because BODY has a default value of "step1", this feedback is displayed at the
0118     beginning of the item session. -->
0119 <qti-feedback-block outcome-identifier="BODY" identifier="step1" show-hide="show">
0120   <qti-content-body>
0121     <p>Use the percent equation, part = percent × whole, to find the sales tax.</p>
0122   </qti-content-body>
0123 </qti-feedback-block>
0124
0125 <-- This Feedback is displayed if the BODY variable contains the identifier "step2". -->
0126 <qti-feedback-block outcome-identifier="BODY" identifier="step2" show-hide="show">
0127   <qti-content-body>
0128     <p>The sales rate times the selling price represents the dollar amount of the
0129       sales tax.</p>
0130   </qti-content-body>
0131 </qti-feedback-block>
0132
0133 <-- This Feedback is displayed if the BODY variable contains the identifier "step3". -->
0134 <qti-feedback-block outcome-identifier="BODY" identifier="step3" show-hide="show">
0135   <qti-content-body>
0136     <p>Substitute into the percent equation.</p>
0137     <table class="qti-margin-s-2">
0138       <tbody>
0139         <tr>
0140           <td class="qti-align-right qti-valign-middle qti-width-24">sales tax</td>
0141           <td class="qti-valign-middle">= sales tax rate × selling price</td>
0142         </tr>
0143       </tbody>
0144     </table>
0145   </qti-content-body>
0146 </qti-feedback-block>
0147
0148 <-- Displayed Feedback if the BODY variable contains the identifier "step4". -->
0149 <qti-feedback-block identifier="step4" outcome-identifier="BODY" show-hide="show">
0150   <qti-content-body>
0151     <table class="qti-margin-s-2">
0152       <tbody>
0153         <tr>
0154           <!-- This is just an empty spacer cell for left side of equation -->
0155           <td class="qti-align-right qti-valign-middle qti-width-24"><span> </span></td>
0156           <td class="qti-valign-middle">= <qti-printed-variable identifier="TAX_PERCENT"/>% ×
0157 <qti-printed-variable identifier="SALESPRICE" format="%2f" /></td>
0158         </tr>
0159       </tbody>
0160     </table>
0161   </qti-content-body>
0162 </qti-feedback-block>
0163
0164 <!-- Displayed for Feedback Step 5 -->
0165 <qti-feedback-block identifier="step5" outcome-identifier="BODY" show-hide="show">
0166   <qti-content-body>
0167     <p class="qti-margin-t-2">The fraction form of the sales tax rate is
0168       <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline" overflow="scroll">
0169         <mrow>
0170           <mfrac>
0171             <mrow><mi>TAX_PERCENT</mi></mrow>
0172             <mrow><mi>100</mi></mrow>
0173           </mfrac>
0174         </mrow>
0175       </math>.
0176     </p>
0177   </qti-content-body>
0178 </qti-feedback-block>

```

```

0179 <!-- Displayed for Feedback Step 6 -->
0180 <qti-feedback-block outcome-identifier="BODY" identifier="step6" show-hide="show">
0181 <qti-content-body>
0182 <p>Solve the percent equation.</p>
0183 <table class="qti-margin-s-2">
0184 <tbody>
0185 <tr>
0186 <td class="qti-align-right qti-valign-middle qti-width-24">sales tax</td>
0187 <td class="qti-valign-middle">= <math xmlns="http://www.w3.org/1998/Math/MathML"
0188 display="inline" overflow="scroll">
0189 <mrow>
0190 <mfrac>
0191 <mrow><mi>TAX_PERCENT</mi></mrow>
0192 <mrow><mi>100</mi></mrow>
0193 </mfrac>
0194 </mrow>
0195 </math> × <qti-printed-variable identifier="SALESPRICE"
0196 format="%.2f" />
0197 </td>
0198 </tr>
0199 </tbody>
0200 </table>
0201 </qti-content-body>
0202 </qti-feedback-block>
0203
0204 <!-- Displayed for Feedback Step 7 -->
0205 <qti-feedback-block outcome-identifier="BODY" identifier="step7" show-hide="show">
0206 <qti-content-body>
0207 <table class="qti-margin-s-2 qti-margin-b-1">
0208 <tbody>
0209 <tr>
0210 <!-- This is just an empty spacer cell for left side of equation -->
0211 <td class="qti-width-24"><span> </span></td>
0212 <td>= <qti-printed-variable identifier="TOTAL" format="%.2f" /></td>
0213 </tr>
0214 </tbody>
0215 </table>
0216 </qti-content-body>
0217 </qti-feedback-block>
0218
0219 <!-- Displayed for Feedback Step 8. Done! -->
0220 <qti-feedback-block outcome-identifier="BODY" identifier="step8" show-hide="show">
0221 <qti-content-body>
0222 <p class="qti-margin-t-2 qti-margin-b-3">
0223 The sales tax is $<qti-printed-variable identifier="TOTAL" format="%.2f" />.
0224 </p>
0225 </qti-content-body>
0226 </qti-feedback-block>
0227
0228 </div> <!-- /card-body -->
0229
0230 <div class="qti3-player-item-card-footer qti-height-14 qti-padding-2">
0231 <qti-end-attempt-interaction response-identifier="EndAttempt"
0232 title="Continue"
0233 class="endattempt-controller-bar"
0234 data-steps="8"
0235 data-controller-type="showexample"
0236 data-hastemplates="true"/>
0237 </div> <!-- /card-footer -->
0238
0239 </div> <!-- /card -->
0240 </qti-item-body>
0241
0242 <!-- RP is executed every time the candidate clicks EndAttempt. However, because this is an

```

```

0243     adaptive item, the item session will not end until completionStatus has a value
0244     of "completed"; i.e., at the end of Step 8 -->
0245 <qti-response-processing>
0246 <qti-response-condition>
0247 <qti-response-if>
0248 <qti-equal>
0249 <qti-variable identifier="STEP"/>
0250 <qti-base-value base-type="integer">1</qti-base-value>
0251 </qti-equal>
0252 <!-- If STEP = 1, increment STEP to 2 -->
0253 <qti-set-outcome-value identifier="STEP">
0254 <qti-base-value base-type="integer">2</qti-base-value>
0255 </qti-set-outcome-value>
0256 <!-- Add the step2 identifier to BODY so that the step2 Feedback is displayed -->
0257 <qti-set-outcome-value identifier="BODY">
0258 <qti-multiple>
0259 <qti-variable identifier="BODY"/>
0260 <qti-base-value base-type="identifier">step2</qti-base-value>
0261 </qti-multiple>
0262 </qti-set-outcome-value>
0263 </qti-response-if>
0264
0265 <qti-response-else-if>
0266 <qti-equal>
0267 <qti-variable identifier="STEP"/>
0268 <qti-base-value base-type="integer">2</qti-base-value>
0269 </qti-equal>
0270 <!-- If STEP = 2, increment STEP to 3 -->
0271 <qti-set-outcome-value identifier="STEP">
0272 <qti-base-value base-type="integer">3</qti-base-value>
0273 </qti-set-outcome-value>
0274 <!-- Add the step3 identifier to BODY so that the step3 Feedback is displayed -->
0275 <qti-set-outcome-value identifier="BODY">
0276 <qti-multiple>
0277 <qti-variable identifier="BODY"/>
0278 <qti-base-value base-type="identifier">step3</qti-base-value>
0279 </qti-multiple>
0280 </qti-set-outcome-value>
0281 </qti-response-else-if>
0282
0283 <qti-response-else-if>
0284 <qti-equal>
0285 <qti-variable identifier="STEP"/>
0286 <qti-base-value base-type="integer">3</qti-base-value>
0287 </qti-equal>
0288 <!-- If STEP = 3, increment STEP to 4 -->
0289 <qti-set-outcome-value identifier="STEP">
0290 <qti-base-value base-type="integer">4</qti-base-value>
0291 </qti-set-outcome-value>
0292 <!-- Add the step4 identifier to BODY so that the step4 Feedback is displayed -->
0293 <qti-set-outcome-value identifier="BODY">
0294 <qti-multiple>
0295 <qti-variable identifier="BODY"/>
0296 <qti-base-value base-type="identifier">step4</qti-base-value>
0297 </qti-multiple>
0298 </qti-set-outcome-value>
0299 </qti-response-else-if>
0300
0301 <qti-response-else-if>
0302 <qti-equal>
0303 <qti-variable identifier="STEP"/>
0304 <qti-base-value base-type="integer">4</qti-base-value>
0305 </qti-equal>
0306 <!-- If STEP = 4, increment STEP to 5 -->

```

```

0307 <qti-set-outcome-value identifier="STEP">
0308   <qti-base-value base-type="integer">5</qti-base-value>
0309 </qti-set-outcome-value>
0310 <!-- Add the step5 identifier to BODY so that the step5 Feedback is displayed -->
0311 <qti-set-outcome-value identifier="BODY">
0312   <qti-multiple>
0313     <qti-variable identifier="BODY"/>
0314     <qti-base-value base-type="identifier">step5</qti-base-value>
0315   </qti-multiple>
0316 </qti-set-outcome-value>
0317 </qti-response-else-if>
0318
0319 <qti-response-else-if>
0320   <qti-equal>
0321     <qti-variable identifier="STEP"/>
0322     <qti-base-value base-type="integer">5</qti-base-value>
0323   </qti-equal>
0324   <!-- If STEP = 5, increment STEP to 6 -->
0325   <qti-set-outcome-value identifier="STEP">
0326     <qti-base-value base-type="integer">6</qti-base-value>
0327   </qti-set-outcome-value>
0328   <!-- Add the step6 identifier to BODY so that the step6 Feedback is displayed -->
0329   <qti-set-outcome-value identifier="BODY">
0330     <qti-multiple>
0331       <qti-variable identifier="BODY"/>
0332       <qti-base-value base-type="identifier">step6</qti-base-value>
0333     </qti-multiple>
0334   </qti-set-outcome-value>
0335 </qti-response-else-if>
0336
0337 <qti-response-else-if>
0338   <qti-equal>
0339     <qti-variable identifier="STEP"/>
0340     <qti-base-value base-type="integer">6</qti-base-value>
0341   </qti-equal>
0342   <!-- If STEP = 6, increment STEP to 7 -->
0343   <qti-set-outcome-value identifier="STEP">
0344     <qti-base-value base-type="integer">7</qti-base-value>
0345   </qti-set-outcome-value>
0346   <!-- Add the step7 identifier to BODY so that the step7 Feedback is displayed -->
0347   <qti-set-outcome-value identifier="BODY">
0348     <qti-multiple>
0349       <qti-variable identifier="BODY"/>
0350       <qti-base-value base-type="identifier">step7</qti-base-value>
0351     </qti-multiple>
0352   </qti-set-outcome-value>
0353 </qti-response-else-if>
0354
0355 <qti-response-else-if>
0356   <qti-equal>
0357     <qti-variable identifier="STEP"/>
0358     <qti-base-value base-type="integer">7</qti-base-value>
0359   </qti-equal>
0360   <!-- If STEP = 7, increment STEP to 8 -->
0361   <qti-set-outcome-value identifier="STEP">
0362     <qti-base-value base-type="integer">8</qti-base-value>
0363   </qti-set-outcome-value>
0364   <!-- Add the step8 identifier to BODY so that the step8 Feedback is displayed -->
0365   <qti-set-outcome-value identifier="BODY">
0366     <qti-multiple>
0367       <qti-variable identifier="BODY"/>
0368       <qti-base-value base-type="identifier">step8</qti-base-value>
0369     </qti-multiple>
0370   </qti-set-outcome-value>

```

```

0371     </qti-response-else-if>
0372
0373     <qti-response-else>
0374         <!-- STEP is 8. We are done. Set completionStatus to completed. -->
0375         <qti-set-outcome-value identifier="STEP">
0376             <qti-base-value base-type="integer">8</qti-base-value>
0377         </qti-set-outcome-value>
0378         <!-- completionStatus must be specifically set to completed in adaptive questions -->
0379         <qti-set-outcome-value identifier="completionStatus">
0380             <qti-base-value base-type="identifier">completed</qti-base-value>
0381         </qti-set-outcome-value>
0382     </qti-response-else>
0383 </qti-response-condition>
0384 </qti-response-processing>
0385 </qti-assessment-item>

```

1. **Line 0007** - This item is adaptive (adaptive="true"). An End Attempt action triggered by a click on "Check Answer" executes the response processing and sets outcome variables. This happens repeatedly until the built-in QTI outcome variable "completionStatus" is set to the value "completed".
2. **Lines 0010-0011** - Define a response variable named "EndAttempt" to be bound to the EndAttempt interaction which provides the endattemptl-controller-bar UX. EndAttempt interactions must have a boolean base-type and single cardinality.
3. **Lines 0044-0046** - Define a template variable named "TAX_PERCENT". The value of TAX_PERCENT will be used in some MathML in Feedback step 5 (**line 0167**) and Feedback step 6 (**line 0187**). Consequently, TAX_PERCENT has the math-variable="true" attribute on **line 0046**.
4. **Lines 0056-0083** - Template Processing block. Template processing computes the values of four template variables each time a new template is requested.
5. **Lines 0119-0123** - Feedback block step1 displayed on initial load because the BODY variable has a default value of "step1".

Use the percent equation, $\text{part} = \text{percent} \times \text{whole}$, to find the sales tax.

6. **Lines 0126-0131** - Feedback block step2 displayed after first click of EndAttempt "Continue" button.

The sales rate times the selling price represents the dollar amount of the sales tax.

7. **Lines 0134-0146** - Feedback block step3 displayed after second click of EndAttempt "Continue" button.

Substitute into the percent equation.

$$\text{sales tax} = \text{sales tax rate} \times \text{selling price}$$

8. **Lines 0149-0162** - Feedback block step4 displayed after third click on EndAttempt “Continue” button. The sales tax percentage (5) is a PrintedVariable of TAX_PERCENT (**line 0156**) and the sales price amount (48.0) is a PrintedVariable of SALESPRICE (**lines 0156-0157**) with format=”%.2f”.

$$= 5\% \times 48.00$$

9. **Lines 0165-0177** - Feedback block step5 displayed after fourth click on EndAttempt “Continue” button. Because the fraction 5/100 is MathML, we’ve used math-variable=”true” to stamp the TAX_PERCENT value (5) into the MathML expression (**line 0171**) `<mrow><mi>TAX_PERCENT</mi></mrow>`.

The fraction form of the sales tax rate is $\frac{5}{100}$.

10. **Lines 0180-0202** - Feedback block step6 displayed after fifth click on EndAttempt “Continue” button. Because the fraction 5/100 is MathML, we’ve used math-variable=”true” to stamp the TAX_PERCENT value (5) into the MathML expression (**line 0191**) `<mrow><mi>TAX_PERCENT</mi></mrow>`. The sales price amount (48.0) is a PrintedVariable of SALESPRICE (**lines 0195-0196**) with format=”%.2f”.

Solve the percent equation.

$$\text{sales tax} = \frac{5}{100} \times 48.00$$

11. **Lines 0205-0217** - Feedback block step7 displayed after sixth click on EndAttempt “Continue” button. The total sales tax value (2.4) is a PrintedVariable of TOTAL (**line 0212**) with format=”%.2f”.

$$= 2.40$$

12. **Lines 0220-0226** - Final Feedback block step8 displayed after seventh click on EndAttempt “Continue” button. The total sales tax value (2.4) is a PrintedVariable of TOTAL (**line 0223**) with format=”%.2f”.

The sales tax is \$2.40.

The EndAttempt interaction “Continue” button is disabled because there are no further steps.



13. **Lines 0231-0236** - Configure the QtiEndAttempt endattempt-controller-bar.

title="Continue" - display the text “Continue” on the EndAttempt action button.
 data-hastemplates="true" - results in the display of the “New Question” template button.
 data-controller-type="showexample" - always use this (or “solve”) on adaptive items.
 data-steps="8" - endattempt-controller-bar displays max of 8 steps in the Progress Stepper.

14. **Lines 0245-0384** - Response Processing maintains two state variables: STEP and BODY in order to display the various FeedbackBlock elements. STEP is incremented by 1 each time the Response Processing is executed; i.e., each time the candidate clicks the EndAttempt “Continue” button.

The BODY variable is multiple cardinality. The current step identifier (“step1”, “step2”, “step3”, etc.) is appended to the BODY array each time Response Processing is executed.

15. **Lines 0379-0382** - The built-in outcome variable “completionStatus” - only used on adaptive items - is set to the “completed” value, indicating that the EndAttempt interaction has truly ended the attempt.

TDX - solve

A TDX “Solve This” item combines adaptive Feedback features of TDX “Show an Example” items with interaction types and Modal Feedback features of TDX “Baseline” items.

The QTI 3 XML in this example also implements Template Processing and template variables, thus enabling a virtually-unlimited number of combinations of the template variables and enabling virtually-unlimited numbers of “Solve This” combinations of this item template.

```

0001 <qti-assessment-item
0002   xmlns="http://www.imsglobal.org/xsd/imsqtiasi v3p0"
0003   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
0004   xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqtiasi v3p0
0005   https://purl.imsglobal.org/spec/qti/v3p0/schema/xsd/imsqti_asiv3p0_v1p0.xsd"
0006   identifier="tdx-solve-01" title="Compute the Sales Tax - solve"
0007   adaptive="true" time-dependent="false">
0008
0009   <!-- Define a variable to be used in Feedback step2 choice interaction -->
0010   <qti-response-declaration identifier="RESPONSE2"
0011     base-type="identifier" cardinality="single">
0012     <qti-correct-response>
0013       <qti-value>A</qti-value>
0014     </qti-correct-response>
0015   </qti-response-declaration>
0016
0017   <!-- Define a variable to be used in Feedback step4 text entry interaction -->
0018   <qti-response-declaration identifier="RESPONSE4"
0019     base-type="string" cardinality="single"/>
0020
0021   <!-- Define a variable to be used to store a correct float representation of the
0022     RESPONSE4 variable -->
0023   <qti-response-declaration identifier="RESPONSE4_FLOAT"
0024     base-type="float" cardinality="single"/>
0025
0026   <!-- Define a variable to be used in Feedback step7 text entry interaction -->
0027   <qti-response-declaration identifier="RESPONSE7"
0028     base-type="string" cardinality="single"/>
0029
0030   <!-- Define a variable to be used to store a correct float representation of the
0031     RESPONSE7 variable -->
0032   <qti-response-declaration identifier="RESPONSE7_FLOAT"
0033     base-type="float" cardinality="single"/>
0034
0035   <!-- Define a variable to bind to the End Attempt Interaction mx1-controller-bar -->
0036   <qti-response-declaration identifier="EndAttempt"
0037     base-type="boolean" cardinality="single" />
0038
0039   <!-- This item is not scored, but we declare SCORE out of habit. -->
0040   <qti-outcome-declaration identifier="SCORE"
0041     base-type="float" cardinality="single">
0042     <qti-default-value>
0043       <qti-value>0</qti-value>
0044     </qti-default-value>
0045   </qti-outcome-declaration>
0046
0047   <!-- Define a STEP variable to keep track of our current step in the adaptive item.
0048     Initialize STEP to a value of 1 (first step) -->
0049   <qti-outcome-declaration identifier="STEP"
0050     base-type="integer" cardinality="single">
0051     <qti-default-value>
0052       <qti-value>1</qti-value>

```

```

0053 </qti-default-value>
0054 </qti-outcome-declaration>
0055
0056 <!-- Define a feedback variable named BODY. Its cardinality is "multiple" so that it can
0057      contain the identifiers of several feedback elements and, in this case, it is initialized
0058      to the identifier "step1" of the first feedback element of this adaptive item. -->
0059 <qti-outcome-declaration identifier="BODY"
0060      base-type="identifier" cardinality="multiple">
0061   <qti-default-value>
0062     <qti-value>step1</qti-value>
0063   </qti-default-value>
0064 </qti-outcome-declaration>
0065
0066 <!-- Define a variable to bind to Modal Feedback2 (feedback for step2 RESPONSE2) -->
0067 <qti-outcome-declaration identifier="FEEDBACK2"
0068      base-type="identifier" cardinality="single"/>
0069
0070 <!-- Define a variable to bind to Modal Feedback4 (feedback for step4 RESPONSE4) -->
0071 <qti-outcome-declaration identifier="FEEDBACK4"
0072      base-type="identifier" cardinality="single"/>
0073
0074 <!-- Define a variable to bind to Modal Feedback7 (feedback for step7 RESPONSE7) -->
0075 <qti-outcome-declaration identifier="FEEDBACK7"
0076      base-type="identifier" cardinality="single"/>
0077
0078 <!-- Define a variable used to display the student's in step7 feedback -->
0079 <qti-outcome-declaration identifier="FEEDBACK7_RESPONSE"
0080      base-type="string" cardinality="single"/>
0081
0082 <qti-template-declaration identifier="SALESPRICE"
0083      cardinality="single" base-type="float"
0084      math-variable="false" param-variable="false"/>
0085
0086 <qti-template-declaration identifier="TAX_PERCENT"
0087      cardinality="single" base-type="float"
0088      math-variable="true" param-variable="false"/>
0089
0090 <qti-template-declaration identifier="SALESTAX"
0091      cardinality="single" base-type="float"
0092      math-variable="false" param-variable="false"/>
0093
0094 <qti-template-declaration identifier="TOTAL"
0095      cardinality="single" base-type="float"
0096      math-variable="false" param-variable="false"/>
0097
0098 <qti-template-processing>
0099   <qti-set-template-value identifier="SALESPRICE">
0100     <qti-integer-to-float>
0101       <qti-random-integer min="2" max="99" step="1"/>
0102     </qti-integer-to-float>
0103   </qti-set-template-value>
0104   <qti-set-template-value identifier="TAX_PERCENT">
0105     <qti-divide>
0106       <qti-random-integer min="300" max="700" step="25"/>
0107       <qti-base-value base-type="float">100</qti-base-value>
0108     </qti-divide>
0109   </qti-set-template-value>
0110   <qti-set-template-value identifier="SALESTAX">
0111     <qti-divide>
0112       <qti-variable identifier="TAX_PERCENT"/>
0113       <qti-base-value base-type="float">100</qti-base-value>
0114     </qti-divide>
0115   </qti-set-template-value>
0116   <qti-set-template-value identifier="TOTAL">

```

```

0116 <qti-round-to rounding-mode="decimalPlaces" figures="2">
0117 <qti-product>
0118 <qti-variable identifier="SALESTAX"/>
0119 <qti-variable identifier="SALESPRICE"/>
0120 </qti-product>
0121 </qti-round-to>
0122 </qti-set-template-value>
0123 <qti-set-correct-response identifier="RESPONSE4_FLOAT">
0124 <qti-variable identifier="SALESPRICE"/>
0125 </qti-set-correct-response>
0126 <qti-set-correct-response identifier="RESPONSE7_FLOAT">
0127 <qti-variable identifier="TOTAL"/>
0128 </qti-set-correct-response>
0129 </qti-template-processing>
0130
0131 <qti-item-body>
0132 <div class="qti3-player-item-card-bordered-rounded qti3-player-item-card-raised-rounded">
0133 <div class="qti3-player-item-card-body qti-padding-2">
0134 <div class="qti-layout-row">
0135 <div class="qti-layout-col8">
0136 <p>Find the sales tax.</p>
0137 <table class="table table-bordered">
0138 <thead>
0139 <tr>
0140 <th class="qti-align-center" colspan="3">Sales Tax</th>
0141 </tr>
0142 <tr>
0143 <th scope="col" class="qti-align-center qti-width-1-3">Selling Price</th>
0144 <th scope="col" class="qti-align-center qti-width-1-3">Rate of Sales Tax</th>
0145 <th scope="col" class="qti-align-center qti-width-1-3">Sales Tax</th>
0146 </tr>
0147 </thead>
0148 <tbody>
0149 <tr>
0150 <td class="qti-align-center">
0151 <qti-printed-variable identifier="SALESPRICE" format="%.2f" />
0152 </td>
0153 <td class="qti-align-center">
0154 <qti-printed-variable identifier="TAX_PERCENT"/>%
0155 </td>
0156 <td class="qti-align-center">?</td>
0157 </tr>
0158 </tbody>
0159 </table>
0160 <hr />
0161 </div>
0162 </div>
0163
0164 <-- This Feedback is displayed if the BODY variable contains the identifier "step1".
0165 <-- Because BODY has a default value of "step1", this feedback is displayed at the
0166 <-- beginning of the item session. -->
0167 <qti-feedback-block outcome-identifier="BODY" identifier="step1" show-hide="show">
0168 <qti-content-body>
0169 <p>Use the percent equation, part = percent × whole, to find the sales tax.</p>
0170 </qti-content-body>
0171 </qti-feedback-block>
0172
0173 <-- This Feedback is displayed if the BODY variable contains the identifier "step2". -->
0174 <qti-feedback-block outcome-identifier="BODY" identifier="step2" show-hide="show">
0175 <qti-content-body>
0176 <p>Which of these represents the dollar amount of the sales tax?</p>
0177
0178 <!-- This is the multiple choice interaction presented in step2 where we ask the
    the candidate about how to compute sales tax. Modal feedback is provided

```

```

0179         after the candidate clicks Continue. -->
0180 <qti-choice-interaction class="qti-labels-none" max-choices="1"
0181         shuffle="true" response-identifier="RESPONSE2">
0182     <qti-simple-choice identifier="A">
0183         Sales tax rate times selling price
0184     </qti-simple-choice>
0185     <qti-simple-choice identifier="B">
0186         Sales tax rate divided by selling price
0187     </qti-simple-choice>
0188     <qti-simple-choice identifier="C">
0189         Sales tax rate plus selling price
0190     </qti-simple-choice>
0191     <qti-simple-choice identifier="D">
0192         Sales tax rate minus selling price
0193     </qti-simple-choice>
0194 </qti-choice-interaction>
0195 </qti-content-body>
0196 </qti-feedback-block>
0197
0198 <-- This Feedback is displayed if the BODY variable contains the identifier "step3". -->
0199 <qti-feedback-block outcome-identifier="BODY" identifier="step3" show-hide="show">
0200     <qti-content-body>
0201         <p>Substitute into the percent equation.</p>
0202         <table class="qti-margin-s-2">
0203             <tbody>
0204                 <tr>
0205                     <td class="qti-align-right qti-valign-middle qti-width-24">sales tax</td>
0206                     <td class="valign-middle">= sales tax rate × selling price</td>
0207                 </tr>
0208             </tbody>
0209         </table>
0210     </qti-content-body>
0211 </qti-feedback-block>
0212
0213
0214 <-- Displayed Feedback if the BODY variable contains the identifier "step4". -->
0215 <qti-feedback-block identifier="step4" outcome-identifier="BODY" show-hide="show">
0216     <qti-content-body>
0217         <table class="qti-margin-s-2">
0218             <tbody>
0219                 <tr>
0220                     <!-- This is just an empty spacer cell for left side of equation -->
0221                     <td class="qti-align-right qti-valign-middle qti-width-24"><span> </span></td>
0222                     <!-- Here, we nest a text entry interaction to capture candidate's understanding
0223                          of selling price. -->
0224                     <td class="valign-middle">= <qti-printed-variable identifier="TAX_PERCENT"/>% ×
0225 <qti-text-entry-interaction response-identifier="RESPONSE4"
0226         class="qti-input-width-6" pattern-mask="([0-9.\-]{0,8})"
0227         />
0228                     </td>
0229                 </tr>
0230             </tbody>
0231         </table>
0232     </qti-content-body>
0233 </qti-feedback-block>
0234
0235 <!-- Displayed for Feedback Step 5 -->
0236 <qti-feedback-block identifier="step5" outcome-identifier="BODY" show-hide="show">
0237     <qti-content-body>
0238         <p class="qti-margin-t-2">The fraction form of the sales tax rate is
0239         <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline" overflow="scroll">
0240             <mrow>
0241                 <mfrac>
0242                     <mrow><mi>TAX_PERCENT</mi></mrow>

```

```

0243         <mrow><mi>100</mi></mrow>
0244     </mfrac>
0245 </mrow>
0246 </math>
0247 </p>
0248 </qti-content-body>
0249 </qti-feedback-block>
0250
0251 <!-- Displayed for Feedback Step 6 -->
0252 <qti-feedback-block outcome-identifier="BODY" identifier="step6" show-hide="show">
0253 <qti-content-body>
0254 <p>Solve the percent equation.</p>
0255 <table class="qti-margin-s-2">
0256 <tbody>
0257 <tr>
0258 <td class="qti-align-right qti-valign-middle qti-width-24">sales tax</td>
0259 <td class="qti-valign-middle">= <math xmlns="http://www.w3.org/1998/Math/MathML"
0260 display="inline" overflow="scroll">
0261 <mrow>
0262 <mfrac>
0263 <mrow><mi>TAX_PERCENT</mi></mrow>
0264 <mrow><mi>100</mi></mrow>
0265 </mfrac>
0266 </mrow>
0267 </math> × <qti-printed-variable identifier="SALESPRICE"
0268 format="%.2f" />
0269 </td>
0270 </tr>
0271 </tbody>
0272 </table>
0273 </qti-content-body>
0274 </qti-feedback-block>
0275
0276 <!-- Displayed for Feedback Step 7 -->
0277 <qti-feedback-block outcome-identifier="BODY" identifier="step7" show-hide="show">
0278 <qti-content-body>
0279 <table class="qti-margin-s-2 qti-margin-b-1">
0280 <tbody>
0281 <tr>
0282 <!-- This is just an empty spacer cell for left side of equation -->
0283 <td class="qti-width-24"><span> </span></td>
0284 <!-- Here, we nest a text entry interaction to capture candidate's understanding
0285 of how to compute the final sales tax amount. -->
0286 <td>= <qti-text-entry-interaction response-identifier="RESPONSE7"
0287 class="qti-input-width-6" pattern-mask="([0-9.\-]{0,8})"
0288 /> <span class="text-muted">(Round to two decimal places as needed)</span>
0289 </td>
0290 </tr>
0291 </tbody>
0292 </table>
0293 </qti-content-body>
0294 </qti-feedback-block>
0295
0296 <!-- Displayed for Feedback Step 8. Done! -->
0297 <qti-feedback-block outcome-identifier="BODY" identifier="step8" show-hide="show">
0298 <qti-content-body>
0299 <p class="qti-margin-t-2 qti-margin-b-3">
0300 The sales tax is $<qti-printed-variable identifier="TOTAL" format="%.2f" />.
0301 </p>
0302 </qti-content-body>
0303 </qti-feedback-block>
0304
0305 </div> <!-- /card-body -->
0306

```

```

0307 <div class="qti3-player-item-card-footer qti-height-14 qti-padding-2">
0308 <qti-end-attempt-interaction response-identifier="EndAttempt"
0309 title="Continue"
0310 class="endattempt-controller-bar"
0311 data-steps="8"
0312 data-controller-type="solve"
0313 data-hastemplates="true"/>
0314 </div> <!-- /card-footer -->
0315
0316 </div> <!-- /card -->
0317 </qti-item-body>
0318
0319 <!-- RP is executed every time the candidate clicks EndAttempt. However, because this is an
0320 adaptive item, the item session will not end until completionStatus has a value
0321 of "completed"; i.e., at the end of Step 8 -->
0322 <qti-response-processing>
0323 <qti-response-condition>
0324 <qti-response-if>
0325 <qti-equal>
0326 <qti-variable identifier="STEP"/>
0327 <qti-base-value base-type="integer">1</qti-base-value>
0328 </qti-equal>
0329 <!-- If STEP = 1, increment STEP to 2 -->
0330 <qti-set-outcome-value identifier="STEP">
0331 <qti-base-value base-type="integer">2</qti-base-value>
0332 </qti-set-outcome-value>
0333 <!-- Add the step2 identifier to BODY so that the step2 Feedback is displayed -->
0334 <qti-set-outcome-value identifier="BODY">
0335 <qti-multiple>
0336 <qti-variable identifier="BODY"/>
0337 <qti-base-value base-type="identifier">step2</qti-base-value>
0338 </qti-multiple>
0339 </qti-set-outcome-value>
0340 </qti-response-if>
0341
0342 <qti-response-else-if>
0343 <qti-equal>
0344 <qti-variable identifier="STEP"/>
0345 <qti-base-value base-type="integer">2</qti-base-value>
0346 </qti-equal>
0347 <!-- On step2 we need to evaluate RESPONSE2 (the MC) and show the appropriate
0348 modal feedback to step2 -->
0349 <qti-response-condition>
0350 <qti-response-if>
0351 <qti-match>
0352 <qti-variable identifier="RESPONSE2"/>
0353 <qti-correct identifier="RESPONSE2"/>
0354 </qti-match>
0355 <qti-set-outcome-value identifier="FEEDBACK2">
0356 <qti-base-value base-type="identifier">correct</qti-base-value>
0357 </qti-set-outcome-value>
0358 </qti-response-if>
0359 <qti-response-else>
0360 <qti-set-outcome-value identifier="FEEDBACK2">
0361 <qti-base-value base-type="identifier">incorrect</qti-base-value>
0362 </qti-set-outcome-value>
0363 </qti-response-else>
0364 </qti-response-condition>
0365 <!-- Increment STEP to 3 -->
0366 <qti-set-outcome-value identifier="STEP">
0367 <qti-base-value base-type="integer">3</qti-base-value>
0368 </qti-set-outcome-value>
0369 <!-- Add the step3 identifier to BODY so that the step3 Feedback is displayed -->
0370 <qti-set-outcome-value identifier="BODY">

```

```

0371     <qti-multiple>
0372     <qti-variable identifier="BODY"/>
0373     <qti-base-value base-type="identifier">step3</qti-base-value>
0374     </qti-multiple>
0375   </qti-set-outcome-value>
0376 </qti-response-else-if>
0377
0378 <qti-response-else-if>
0379   <qti-equal>
0380     <qti-variable identifier="STEP"/>
0381     <qti-base-value base-type="integer">3</qti-base-value>
0382   </qti-equal>
0383   <!-- Turn off Step 2 Feedback. Otherwise, step2 Modal will be displayed again. -->
0384   <qti-set-outcome-value identifier="FEEDBACK2">
0385     <qti-base-value base-type="identifier">step3</qti-base-value>
0386   </qti-set-outcome-value>
0387   <!-- Increment STEP to 4 -->
0388   <qti-set-outcome-value identifier="STEP">
0389     <qti-base-value base-type="integer">4</qti-base-value>
0390   </qti-set-outcome-value>
0391   <!-- Add the step4 identifier to BODY so that the step4 Feedback is displayed -->
0392   <qti-set-outcome-value identifier="BODY">
0393     <qti-multiple>
0394     <qti-variable identifier="BODY"/>
0395     <qti-base-value base-type="identifier">step4</qti-base-value>
0396     </qti-multiple>
0397   </qti-set-outcome-value>
0398 </qti-response-else-if>
0399
0400 <qti-response-else-if>
0401   <qti-equal>
0402     <qti-variable identifier="STEP"/>
0403     <qti-base-value base-type="integer">4</qti-base-value>
0404   </qti-equal>
0405   <!-- On step4 we need to evaluate RESPONSE4 (the text entry) and show the appropriate
0406         modal feedback to step4 -->
0407   <qti-response-condition>
0408     <qti-response-if>
0409       <!-- Check numeric equivalence -->
0410       <qti-equal tolerance-mode="exact">
0411         <!-- The value of RESPONSE4 is compared with the correct value identified in the
0412               RESPONSE4_FLOAT declaration -->
0413         <qti-custom-operator class="math.stringToFloat">
0414           <qti-variable identifier="RESPONSE4"/>
0415         </qti-custom-operator>
0416         <qti-correct identifier="RESPONSE4_FLOAT"/>
0417       </qti-equal>
0418       <qti-set-outcome-value identifier="FEEDBACK4">
0419         <qti-base-value base-type="identifier">correct</qti-base-value>
0420       </qti-set-outcome-value>
0421     </qti-response-if>
0422     <qti-response-else>
0423       <qti-set-outcome-value identifier="FEEDBACK4">
0424         <qti-base-value base-type="identifier">incorrect</qti-base-value>
0425       </qti-set-outcome-value>
0426     </qti-response-else>
0427   </qti-response-condition>
0428   <!-- Increment STEP to 5 -->
0429   <qti-set-outcome-value identifier="STEP">
0430     <qti-base-value base-type="integer">5</qti-base-value>
0431   </qti-set-outcome-value>
0432   <!-- Add the step5 identifier to BODY so that the step5 Feedback is displayed -->
0433   <qti-set-outcome-value identifier="BODY">
0434     <qti-multiple>

```

```

0435     <qti-variable identifier="BODY"/>
0436     <qti-base-value base-type="identifier">step5</qti-base-value>
0437   </qti-multiple>
0438 </qti-set-outcome-value>
0439 </qti-response-else-if>
0440
0441 <qti-response-else-if>
0442   <qti-equal>
0443     <qti-variable identifier="STEP"/>
0444     <qti-base-value base-type="integer">5</qti-base-value>
0445   </qti-equal>
0446   <!-- Turn off Step 4 Feedback. Otherwise, step4 Modal will be displayed again. -->
0447   <qti-set-outcome-value identifier="FEEDBACK4">
0448     <qti-base-value base-type="identifier">step5</qti-base-value>
0449   </qti-set-outcome-value>
0450   <!-- Increment STEP to 6 -->
0451   <qti-set-outcome-value identifier="STEP">
0452     <qti-base-value base-type="integer">6</qti-base-value>
0453   </qti-set-outcome-value>
0454   <!-- Add the step6 identifier to BODY so that the step6 Feedback is displayed -->
0455   <qti-set-outcome-value identifier="BODY">
0456     <qti-multiple>
0457       <qti-variable identifier="BODY"/>
0458       <qti-base-value base-type="identifier">step6</qti-base-value>
0459     </qti-multiple>
0460   </qti-set-outcome-value>
0461 </qti-response-else-if>
0462
0463 <qti-response-else-if>
0464   <qti-equal>
0465     <qti-variable identifier="STEP"/>
0466     <qti-base-value base-type="integer">6</qti-base-value>
0467   </qti-equal>
0468   <!-- If STEP = 6, increment STEP to 7 -->
0469   <qti-set-outcome-value identifier="STEP">
0470     <qti-base-value base-type="integer">7</qti-base-value>
0471   </qti-set-outcome-value>
0472   <!-- Add the step7 identifier to BODY so that the step7 Feedback is displayed -->
0473   <qti-set-outcome-value identifier="BODY">
0474     <qti-multiple>
0475       <qti-variable identifier="BODY"/>
0476       <qti-base-value base-type="identifier">step7</qti-base-value>
0477     </qti-multiple>
0478   </qti-set-outcome-value>
0479 </qti-response-else-if>
0480
0481 <qti-response-else-if>
0482   <qti-equal>
0483     <qti-variable identifier="STEP"/>
0484     <qti-base-value base-type="integer">7</qti-base-value>
0485   </qti-equal>
0486   <!-- Save off the response so we can echo it back to the student in feedback - if needed -->
0487   <qti-set-outcome-value identifier="FEEDBACK7_RESPONSE">
0488     <qti-variable identifier="RESPONSE7"/>
0489   </qti-set-outcome-value>
0490   <qti-response-condition>
0491     <qti-response-if>
0492       <!-- Check numeric equivalence -->
0493       <qti-equal tolerance-mode="exact">
0494         <!-- The value of RESPONSE7 is compared with the correct value identified in the
0495              RESPONSE7_FLOAT declaration-->
0496         <qti-custom-operator class="math.stringToFloat">
0497           <qti-variable identifier="RESPONSE7"/>
0498         </qti-custom-operator>

```

```

0499     <qti-correct identifier="RESPONSE7_FLOAT"/>
0500 </qti-equal>
0501 <!-- Response is numerically equivalent to the correct response. Now check form. -->
0502 <qti-response-condition>
0503     <!-- Depending on whether or not the input matches the correct answer, FEEDBACK7 is given
0504         the value of the identifier of the appropriate feedback message-->
0505     <qti-response-if>
0506         <qti-pattern-match pattern="[0-9]*\.[0-9]{2}">
0507             <qti-variable identifier="RESPONSE7"/>
0508         </qti-pattern-match>
0509         <qti-set-outcome-value identifier="FEEDBACK7">
0510             <qti-base-value base-type="identifier">correct</qti-base-value>
0511         </qti-set-outcome-value>
0512     </qti-response-if>
0513     <qti-response-else>
0514         <qti-set-outcome-value identifier="FEEDBACK7">
0515             <qti-base-value base-type="identifier">wrongformat</qti-base-value>
0516         </qti-set-outcome-value>
0517     </qti-response-else>
0518 </qti-response-condition>
0519 </qti-response-if>
0520 <qti-response-else>
0521     <qti-set-outcome-value identifier="FEEDBACK7">
0522         <qti-base-value base-type="identifier">incorrect</qti-base-value>
0523     </qti-set-outcome-value>
0524 </qti-response-else>
0525 </qti-response-condition>
0526 <!-- Increment STEP to 8 -->
0527 <qti-set-outcome-value identifier="STEP">
0528     <qti-base-value base-type="integer">8</qti-base-value>
0529 </qti-set-outcome-value>
0530 <!-- Add the step8 identifier to BODY so that the step8 Feedback is displayed -->
0531 <qti-set-outcome-value identifier="BODY">
0532     <qti-multiple>
0533         <qti-variable identifier="BODY"/>
0534         <qti-base-value base-type="identifier">step8</qti-base-value>
0535     </qti-multiple>
0536 </qti-set-outcome-value>
0537 </qti-response-else-if>
0538
0539 <qti-response-else>
0540     <!-- STEP is 8. We are done. Set completionStatus to completed. -->
0541     <qti-set-outcome-value identifier="STEP">
0542         <qti-base-value base-type="integer">8</qti-base-value>
0543     </qti-set-outcome-value>
0544     <!-- completionStatus must be specifically set to "completed" in adaptive questions -->
0545     <qti-set-outcome-value identifier="completionStatus">
0546         <qti-base-value base-type="identifier">completed</qti-base-value>
0547     </qti-set-outcome-value>
0548 </qti-response-else>
0549 </qti-response-condition>
0550 </qti-response-processing>
0551
0552 <!-- Modal feedback message for correct answer to RESPONSE2 -->
0553 <qti-modal-feedback outcome-identifier="FEEDBACK2" identifier="correct" show-hide="show">
0554     <qti-content-body><p>Well done!</p></qti-content-body>
0555 </qti-modal-feedback>
0556
0557 <!-- Modal feedback message for incorrect answer to RESPONSE2 -->
0558 <qti-modal-feedback outcome-identifier="FEEDBACK2" identifier="incorrect" show-hide="show">
0559     <qti-content-body>
0560         <p>Sorry, your answer is not correct.</p>
0561         <p>Correct Answer: sales tax rate times selling price</p>
0562     </qti-content-body>

```

```

0563 </qti-modal-feedback>
0564
0565 <!-- Modal feedback message for correct answer to RESPONSE4 -->
0566 <qti-modal-feedback outcome-identifier="FEEDBACK4" identifier="correct" show-hide="show">
0567   <qti-content-body>Fantastic!</qti-content-body>
0568 </qti-modal-feedback>
0569
0570 <!-- Modal feedback message for incorrect answer to RESPONSE4 -->
0571 <qti-modal-feedback outcome-identifier="FEEDBACK4" identifier="incorrect" show-hide="show">
0572   <qti-content-body>
0573     <p>Sorry, your answer is not correct.</p>
0574     <p>Correct Answer: <qti-printed-variable identifier="SALESPRICE" format="%.2f" /></p>
0575   </qti-content-body>
0576 </qti-modal-feedback>
0577
0578 <!-- Modal feedback message for correct answer to RESPONSE7 -->
0579 <qti-modal-feedback outcome-identifier="FEEDBACK7" show-hide="show" identifier="correct">
0580   <qti-content-body>
0581     <p>Well done!</p>
0582   </qti-content-body>
0583 </qti-modal-feedback>
0584
0585 <!-- Modal feedback message for wrong format answer to RESPONSE7 -->
0586 <qti-modal-feedback outcome-identifier="FEEDBACK7" show-hide="show" identifier="wrongformat">
0587   <qti-content-body>
0588     <p>
0589       Almost correct. You entered the correct sales tax amount, but in
0590       the <em>wrong format</em>.
0591     </p>
0592     <p>You entered: <qti-printed-variable identifier="FEEDBACK7_RESPONSE" /></p>
0593     <p>However, currency is expressed with exactly two digits after the decimal point.</p>
0594   </qti-content-body>
0595 </qti-modal-feedback>
0596
0597 <!-- Modal feedback message for incorrect answer to RESPONSE7 -->
0598 <qti-modal-feedback outcome-identifier="FEEDBACK7" show-hide="show" identifier="incorrect">
0599   <qti-content-body>
0600     <p>Sorry, your answer is not correct.</p>
0601     <p>Correct Answer: <qti-printed-variable identifier="TOTAL" format="%.2f" /></p>
0602   </qti-content-body>
0603 </qti-modal-feedback>
0604 </qti-assessment-item>

```

1. **Line 0007** - This item is adaptive (adaptive="true"). An End Attempt action triggered by a click on "Check Answer" executes the response processing and sets outcome variables. This happens repeatedly until the built-in QTI outcome variable "completionStatus" is set to the value "completed".
2. **Lines 0018-0033** - Declare response variables for the three interactions in this item. RESPONSE2 is a QtiChoiceInteraction. RESPONSE4 and RESPONSE7 are QtiTextEntryInteractions declared with a "string" base-type. RESPONSE4 and RESPONSE7 also have corresponding RESPONSE4_FLOAT and RESPONSE7_FLOAT (float base-type) declarations that template processing uses to store the correct numeric values of RESPONSE4 and RESPONSE7, respectively.

3. **Lines 0036-0037** - Define a response variable named “EndAttempt” to be bound to the EndAttempt interaction which provides the mxl-controller-bar UX. EndAttempt interactions must have a boolean base-type and single cardinality.
4. **Lines 0067-0080** - Define four outcome variables (FEEDBACK2, FEEDBACK4, FEEDBACK7, FEEDBACK7_RESPONSE) to be used to show/hide Modal Feedback upon evaluation of the RESPONSE2, RESPONSE4, RESPONSE7 interactions.
5. **Lines 0086-0088** - Define a template variable named “TAX_PERCENT”. The value of TAX_PERCENT will be used in some MathML in Feedback step 5 (**line 0238**) and Feedback step 6 (**line 0259**). Consequently, TAX_PERCENT has the math-variable=”true” attribute on **line 0048**.
6. **Lines 0098-0129** - Template Processing block. Template processing computes the values of four template variables each time a new template is requested.
7. **Lines 0167-0171** - Feedback block step1 displayed on initial load because the BODY variable has a default value of “step1”.

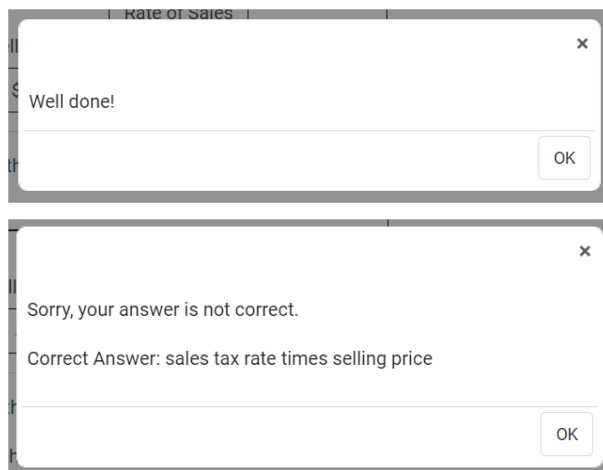
Use the percent equation, $\text{part} = \text{percent} \times \text{whole}$, to find the sales tax.

8. **Lines 0174-0196** - Feedback block step2 displayed after first click of EndAttempt “Continue” button. Feedback block step2 contains a nested QtiChoiceInteraction (**lines 0180-0194**) of single cardinality (radio buttons) bound to response variable RESPONSE2. The QtiChoiceInteraction’s class is set to “qti-labels-none” to imitate the TDX multiple choice UX; i.e., no labels on the choices. The shuffle attribute is set to “true” which tells the QTI 3 Player to randomize the choices each time a new template is requested.

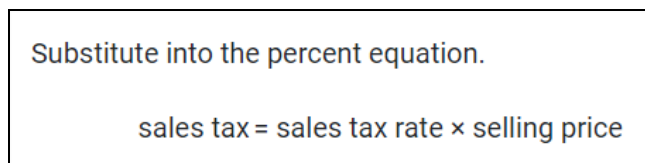
Which of these represents the dollar amount of the sales tax?

- ☐ Sales tax rate minus selling price
- ☐ Sales tax rate plus selling price
- ☐ Sales tax rate times selling price
- ☐ Sales tax rate divided by selling price

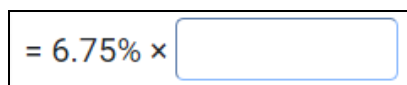
Upon clicking the EndAttempt “Continue” button the QTI 3 Player invokes response processing and evaluates the candidate’s response to RESPONSE2, resulting in the display of the “correct” FEEDBACK2 Modal (**lines 0553-0555**), or the “incorrect” FEEDBACK2 Modal (**lines 0558-0563**).



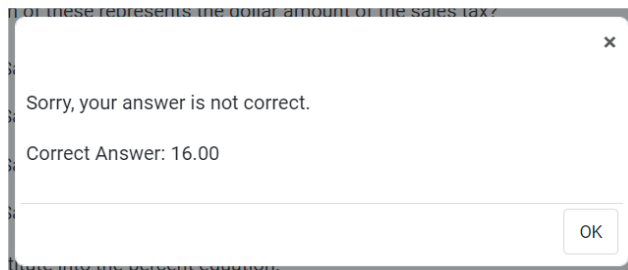
9. **Lines 0199-0211** - Feedback block step3 displayed after second click of EndAttempt “Continue” button.



10. **Lines 0215-0233** - Feedback block step4 displayed after third click on EndAttempt “Continue” button. The sales tax percentage (4.75) is a PrintedVariable of TAX_PERCENT (**line 0224**). Feedback block step4 contains a nested QtTextEntryinteraction (**lines 0224-0226**) bound to response variable RESPONSE4.



Upon clicking the EndAttempt “Continue” button the QTI 3 Player invokes response processing and evaluates the candidate’s response to RESPONSE4, resulting in the display of the “correct” FEEDBACK4 Modal (**lines 0566-0568**), or the “incorrect” FEEDBACK4 Modal (**lines 0571-0575**).



The “incorrect” Modal feedback for RESPONSE4 contains a PrintedVariable of the correct answer to RESPONSE4 computed during template processing.

11. **Lines 0236-0249** - Feedback block step5 displayed after fourth click on EndAttempt “Continue” button. Because the fraction 6.75/100 is MathML, we’ve used math-variable=”true” to stamp the TAX_PERCENT value (6.75) into the MathML expression (**line 0242**) `<mrow><mi>TAX_PERCENT</mi></mrow>`.

The fraction form of the sales tax rate is $\frac{6.75}{100}$.

12. **Lines 0252-0274** - Feedback block step6 displayed after fifth click on EndAttempt “Continue” button. Because the fraction 6.75/100 is MathML, we’ve used math-variable=”true” to stamp the TAX_PERCENT value (6.75) into the MathML expression (**line 0263**) `<mrow><mi>TAX_PERCENT</mi></mrow>`. The sales price amount (16.0) is a PrintedVariable of SALESPRICE (**lines 0267-0268**) with format=”%.2f”.

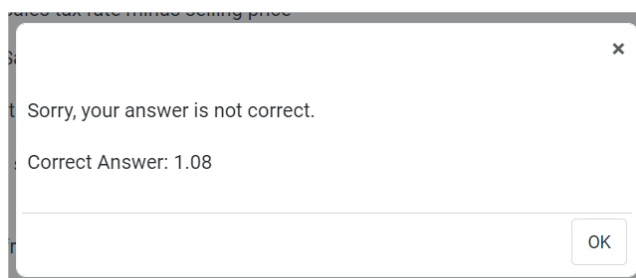
Solve the percent equation.

$$\text{sales tax} = \frac{6.75}{100} \times 16.00$$

13. **Lines 0277-0294** - Feedback block step7 displayed after sixth click on EndAttempt “Continue” button. Feedback block step7 contains a nested QtiTextEntryinteraction (**lines 0286-0288**) bound to response variable RESPONSE7.

=

Upon clicking the EndAttempt “Continue” button the QTI 3 Player invokes response processing and evaluates the candidate’s response to RESPONSE7, resulting in the display of the “correct” FEEDBACK7 Modal (**lines 0578-0582**), the “wrongformat” FEEDBACK7 Modal (**lines 0585-0594**), or the “incorrect” FEEDBACK7 Modal (**lines 0597-0602**).



The “incorrect” Modal feedback for RESPONSE7 contains a PrintedVariable of the correct answer to RESPONSE7 computed during template processing.

14. **Lines 0297-0303** - Final Feedback block step8 displayed after seventh click on EndAttempt “Continue” button. The total sales tax value (1.08) is a PrintedVariable of TOTAL (**line 0300**) with format=”%.2f”.

Thus, the sales tax is \$1.08.

The EndAttempt interaction “Continue” button is disabled because there are no further steps.



15. **Lines 0308-0313** - Configure the QtiEndAttempt endattempt-controller-bar.

title=”Continue” - display the text “Continue” on the EndAttempt action button.
 data-hastemplates=”true” - results in the display of the “New Question” template button.
 data-controller-type=”solve” - always use this (or “showexample”) on adaptive items.
 data-steps=”8” - endattempt-controller-bar displays max of 8 steps in the Progress Stepper.

16. **Lines 0322-0550** - Response Processing maintains two state variables: STEP and BODY in order to display the various FeedbackBlock elements. STEP is incremented by 1 each time the Response Processing is executed; i.e., each time the candidate clicks the EndAttempt “Continue” button.

The BODY variable is multiple cardinality. The current step identifier (“step1”, “step2”, “step3”, etc.) is appended to the BODY array each time Response Processing is executed.

17. **Lines 0545-0547** - The built-in outcome variable “completionStatus” - only used on adaptive items - is set to the “completed” value, indicating that the EndAttempt interaction has truly ended the attempt.