

Lecture 08: JS Numbers, Strings, Objects, User Input

ITP 303 Full-Stack Web Development

More DOM Events

Keyboard Events

<code>onkeydown</code>	Key pressed.
<code>onkeyup</code>	Key released.
<code>onkeypress</code>	Fired immediately after <code>onkeydown</code> . Only works with keys used for “typing” text. Not fired by Ctrl, Alt, arrows, etc.

Form Events

<code>oninput</code>	Form element value changed.
<code>onchange</code>	Form element state changed. Text fields fired after losing focus.
<code>onsubmit</code>	Form is submitted.

Forms & User Input

HTML Forms

Allow for user interaction and data collection.

Result (with additional styling):

Text Field:

Email Field:

Radio Buttons:

☐ Radio 1 ☐ Radio 2

Checkboxes:

☐ Checkbox 1 ☐ Checkbox 2

Drop-down:

Option 1

Textarea:

Submit

Reset

```
<form action="" method="">
  <label for="text-id">Text Field:</label>
  <input type="text" id="text-id" name="text-name">

  <label for="email-id">Email Field:</label>
  <input type="email" id="email-id" name="text-name">

  <label for="">Radio Buttons:</label>
  <label>
    <input type="radio" id="radio-1" name="radio-name" value="val-1">
      Radio 1
    </label>
    <label>
      <input type="radio" id="radio-2" name="radio-name" value="val-2">
        Radio 2
      </label>

  <label for="">Checkboxes:</label>
  <label>
    <input type="checkbox" id="checkbox-1" name="checkbox-name" value="val-1">
      Checkbox 1
    </label>
    <label>
      <input type="checkbox" id="checkbox-2" name="checkbox-name" value="val-2">
        Checkbox 2
      </label>

  <label for="select-id">Drop-down:</label>
  <select id="select-id" name="select-name">
    <option value="val1">Option 1</option>
    <option value="val2">Option 2</option>
  </select>

  <label for="textarea-id">Textarea:</label>
  <textarea id="textarea-id" name="textarea-name"></textarea>

  <button type="submit">Submit</button>
  <button type="reset">Reset</button>
</form>
```

Form Tags

Define HTML form.

<code>action</code>	Specifies where form data is sent upon submission.
---------------------	--

<code>method</code>	Specifies HTTP method used to send data: <code>POST</code> or <code>GET</code> .
---------------------	--

```
<form action="" method="">
```

```
...
```

```
</form>
```

Text Fields

Single-line text fields.

Result (with additional styling):

Text Field:

Text Field:

```
<input type="text" id="text-id" name="text-name">
```

Field name submitted
along with data.

Email Fields

Text fields with basic email validation.

Result (with additional styling):

Email Field:

Email Field:

```
<input type="email" id="email-id" name="text-name">
```

Radio Buttons

Group of mutually exclusive fields.

Result (with additional styling):

☐ Radio 1 ☐ Radio 2

Group radio buttons by setting same `name` attribute value.

```
<input type="radio" id="radio-1" name="radio-name" value="val-1">
```

Radio 1

```
<input type="radio" id="radio-2" name="radio-name" value="val-2">
```

Radio 2

Value submitted when radio button is selected.

Checkboxes

Group of independent fields.

Result (with additional styling):

☐ Checkbox 1 ☐ Checkbox 2

Group checkboxes by setting same `name` attribute value.

```
<input type="checkbox" id="checkbox-1" name="checkbox-name" value="val-1">  
Checkbox 1
```

```
<input type="checkbox" id="checkbox-2" name="checkbox-name" value="val-2">  
Checkbox 2
```

Value submitted when checkbox is selected.

Drop-Down Lists

Pop-up menu that allows users to choose a single option from the list.

Result (with additional styling):

Drop-down:

✓ Option 1
Option 2

Drop-down:

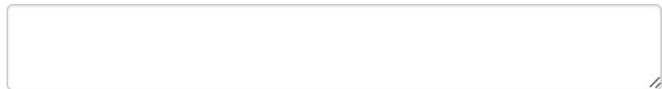
```
<select id="select-id" name="select-name">  
  <option value="val1">Option 1</option>  
  <option value="val2">Option 2</option>  
</select>
```

Textareas

Multi-line text fields.

Result (with additional styling):

Textarea:



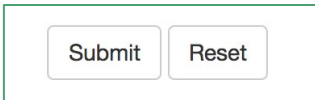
Textarea:

```
<textarea id="textarea-id" name="textarea-name"></textarea>
```

Buttons

Allow users to submit or reset forms.

Result (with additional styling):



```
<button type="submit">Submit</button>  
<button type="reset">Reset</button>
```

Form Labels

Captions for form elements.

Clicking a label triggers assigned form field.

Usage 1 Set `for` attribute to `id` of corresponding form field.

Usage 2 Wrap label around the form field.

Set `for` value to `id` of corresponding field.

```
<label for="text-id">Text field:</label>
<input type="text" id="text-id" name="text-name">
```

```
<label for="select-id">Drop-down:</label>
<select id="select-id" name="select-name">
  <option value="val1">Option 1</option>
  <option value="val2">Option 2</option>
</select>
```

```
<label>
  <input type="radio" id="radio-1" name="radio-name" value="val-1">
  Radio 1
</label>
```

Placeholder Attribute

Text displayed within form field before user input.

Usually used to specify hints or expected formats.

Result (with additional styling):

Text Field:

Text Field Placeholder

Email Field:

ttrojan@usc.edu

Phone Field:

(123) 123-1234

```
<label for="text-id">Text Field:</label>
<input type="text" id="text-id" name="text-name" placeholder="Text Field Placeholder">

<label for="email-id">Email Field:</label>
<input type="email" id="email-id" name="text-name" placeholder="ttrojan@usc.edu">

<label for="phone-id">Phone Field:</label>
<input type="text" id="phone-id" name="text-name" placeholder="(123) 123-1234">
```

Strings

Strings

Strings are series of characters.

- Sometimes treated like arrays with letters in each slot.

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o		W	o	r	l	d	!

`.replace(find, replace)`

Find first occurrence of *find* and replace it with *replace*.

`.substr(start, [length])`

Return substring starting at index *start*.
Length of substring is optional.

```
var myString = "Hello World!";
```

```
console.log( myString.replace('Hello', 'Hi') );  
// Returns "Hi World!"
```

```
console.log( myString.substr(6) );  
// Returns "World!"
```

```
console.log( myString.substr(0, 5) );  
// Returns "Hello"
```

String Operations

<code>.length</code>	Length of the string.
<code>.indexOf (sub)</code>	Return index of first occurrence of <i>sub</i> within the string. Returns -1 if <i>sub</i> is not found.
<code>.toLowerCase (str)</code>	Convert <i>str</i> to lowercase.
<code>.toUpperCase (str)</code>	Convert <i>str</i> to uppercase.

Resource: [String Reference](#).

```
console.log( str.length );
```

```
if (str.indexOf("olo") > -1 ) {  
  console.log('Found');  
}
```

```
if ( str.toLowerCase() == str2.toLowerCase() ) {  
  console.log('Same strings (case-insensitive)');  
}
```

```
if( str.toUpperCase().indexOf('LOREM') == -1 ) {  
  console.log('Not Found');  
}
```

Substring methods

<code>.substr()</code>	Returns the characters in a string beginning at the specified location through the specified number of characters.
<code>.substring()</code>	Returns the characters in a string between two indexes into the string.
<code>.slice()</code>	Extracts a section of a string and returns a new string.
<code>.split()</code>	Splits a String object into an array of strings by separating the string into substrings.

Full list here: https://www.w3schools.com/js/js_string_methods.asp