

Meta Description: Use the power of jQuery selectors in web development. Learn how mastering jQuery selectors helps developers to create dynamic and interactive websites.

jQuery Selectors

jQuery selectors	1
jQuery selectors syntax	2
The role of selectors in jQuery	2
The \$() factory function	2
List of jQuery selectors	2
Conclusion	10
FAQs	10

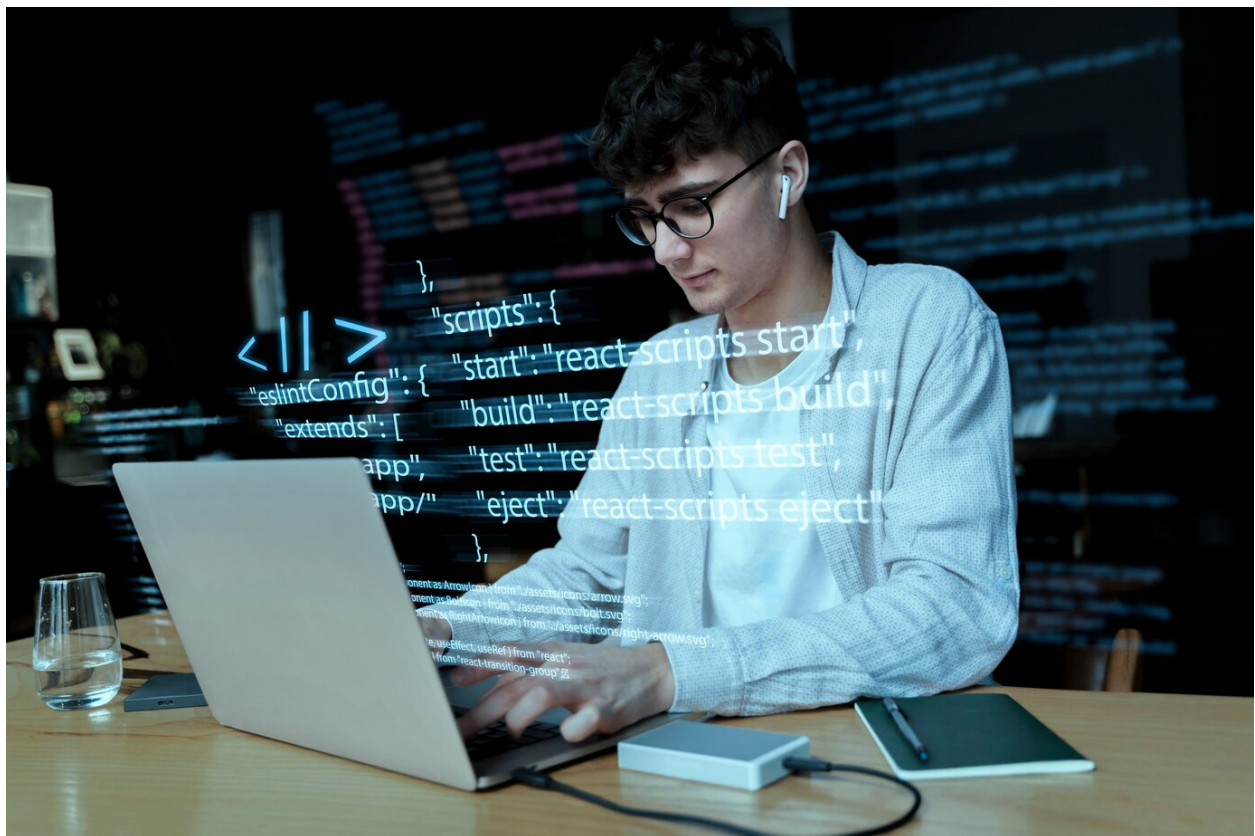


Image Source

Alt Text: A programmer using JavaScript on his computer

As I started learning web development, **jQuery selectors** became my best friends. They made it easy to pick and change things on web pages. Earlier, figuring out which part of a page to change felt like solving a puzzle. But jQuery selectors made it simple. They helped me select exactly what I needed and tweak it just right. Whether it was selecting items or moving around on the page, jQuery selectors were like magic tools that made web development more fun.

jQuery selectors are a fundamental aspect of jQuery, allowing developers to efficiently manipulate HTML elements within a document. They serve as the backbone for various dynamic web functionalities and are crucial for DOM traversal and manipulation.

jQuery selectors and filters enable developers to [target specific HTML elements](#) within a webpage using a concise and powerful syntax. Whether it is selecting elements by their tag name, class, ID, or attributes, jQuery selectors provide a flexible and efficient way to interact with DOM elements. This capability is essential for building dynamic and interactive web applications.

jQuery Selectors Syntax

The syntax for **jQuery selectors** follows a simple pattern:

```
$(selector).action();
```

- The `$()` function initializes jQuery and selects elements based on the provided selector.
- The selector parameter specifies the criteria for selecting elements within the DOM.
- The `.action()` method performs operations or manipulations on the selected elements.

The Role of Selectors in jQuery

Selectors play a central role in jQuery, allowing developers to:

- Dynamically access and modify HTML elements, based on various criteria.
- Traverse the DOM tree to locate specific elements or groups of elements.
- Apply CSS styles, add event handlers, or perform other actions on selected elements.

- Create responsive and interactive user interfaces by selectively targeting and modifying elements in real time.

The \$() Factory Function

The `$()` function, also known as the jQuery factory function, is the primary entry point for using jQuery. It initializes the jQuery library and provides access to a wide range of DOM manipulation methods. By passing a selector as an argument to `$()`, developers can efficiently select and interact with elements on the webpage.

List of jQuery Selectors

Here is the complete list of selectors along with **jQuery selector examples** and descriptions:

- **jQuery universal selector**

`*`

- **Example:** `$("*")`
- **Description:** It selects all elements in the DOM.

- **jQuery ID selector**

`#id`

- **Example:** `$("#lastname")`
- **Description:** It selects the element with the specified ID attribute.

- **Class selector jQuery**

`.class`

- **Example:** `$(".intro")`
- **Description:** It selects all elements with the specified class.

- **jQuery multi class selector**

.class1, .class2

- **Example:** \$("intro, .demo")
- **Description:** It selects all elements that have either class intro or class demo.

- **jQuery element selector**

element

- **Example:** \$("p")
- **Description:** It selects all elements with the specified element type.

- **jQuery multiple elements selector**

el1,el2,el3

- **Example:** \$("h1, div, p")
- **Description:** It selects all elements of the specified types.

- **jQuery first element selector**

:first

- **Example:** \$("p:first")
- **Description:** It selects the first <p> element.

- **jQuery last element selector**

:last

- **Example:** \$("p:last")
- **Description:** It selects the last <p> element.

- **jQuery even element selector**

:even

- **Example:** \$("tr:even")
- **Description:** It selects all even <tr> elements.

- **jQuery odd element selector**

:odd

- **Example:** \$("tr:odd")
- **Description:** It selects all odd <tr> elements.

- **jQuery first child selector**

:first-child

- **Example:** \$("p:first-child")
- **Description:** It selects all elements that are the first child of their parent.

- **jQuery first of type selector**

:first-of-type

- **Example:** \$("p:first-of-type")
- **Description:** It selects all elements that are the first of their type among their siblings.

- **jQuery last child selector**

:last-child

- **Example:** \$("p:last-child")
- **Description:** It selects all elements that are the last child of their parent.

- **jQuery last of type selector**

:last-of-type

- **Example:** \$("p:last-of-type")

- **Description:** It selects all elements that are the last of their type among their siblings.

- **jQuery nth child selector**

:nth-child(n)

- **Example:** \$("p:nth-child(2)")
- **Description:** It selects all elements that are the nth child of their parent.

- **jQuery nth last child selector**

:nth-last-child(n)

- **Example:** \$("p:nth-last-child(2)")
- **Description:** It selects all elements that are the nth child of their parent, counting from the last child.

- **jQuery nth of type selector**

:nth-of-type(n)

- **Example:** \$("p:nth-of-type(2)")
- **Description:** It selects all elements that are the nth of their type among their siblings.

- **jQuery nth last of type selector**

:nth-last-of-type(n)

- **Example:** \$("p:nth-last-of-type(2)")
- **Description:** It selects all elements that are the nth of their type among their siblings, counting from the last child.

- **jQuery only child selector**

:only-child

- **Example:** \$("p:only-child")
- **Description:** It selects all elements that are the only child of their parent.

- **jQuery only of type selector**

:only-of-type

- **Example:** \$("p:only-of-type")
- **Description:** It selects all elements that are the only child of their parent of the same type.

- **jQuery parent > child selector**

parent > child

- **Example:** \$("div > p")
- **Description:** It selects all <p> elements that are a direct child of a <div> element.

- **jQuery parent descendant selector**

parent descendant

- **Example:** \$("div p")
- **Description:** It selects all <p> elements that are descendants of a <div> element.

- **jQuery element + next selector**

element + next

- **Example:** \$("div + p")
- **Description:** It selects the <p> element that is next to each <div> element.

- **jQuery element ~ siblings selector**

element ~ siblings

- **Example:** \$("div ~ p")
- **Description:** It selects all <p> elements that appear after a <div> element.

- **jQuery EQ selector**

:eq(index)

- **Example:** \$("ul li:eq(3)")
- **Description:** It selects the fourth element in a list (index starts at 0).

- **jQuery greater than selector**

:gt(no)

- **Example:** \$("ul li:gt(3)")
- **Description:** It selects list elements with an index greater than 3.

- **jQuery less than selector**

:lt(no)

- **Example:** \$("ul li:lt(3)")
- **Description:** It selects list elements with an index less than 3.

- **jQuery not selector**

:not(selector)

- **Example:** \$("input:not(:empty)")
- **Description:** It selects all input elements that are not empty.

- **jQuery header selector**

:header

- **Example:** \$("header")
- **Description:** It selects all header elements <h1>, <h2>, etc.

- **jQuery animated selector**

:animated

- **Example:** \$(":animated")
- **Description:** It selects all animated elements.

- **jQuery focus selector**

:focus

- **Example:** \$("focus")
- **Description:** It selects the element that currently has focus.

- **jQuery Contains Selector**

:contains(text)

- **Example:** \$("contains('Hello')")
- **Description:** It selects all elements that contain the specified text.

- **jQuery has selector**

:has(selector)

- **Example:** \$("div:has(p)")
- **Description:** It selects all <div> elements that have a <p> element.

- **jQuery empty selector**

:empty

- **Example:** \$("div:empty")
- **Description:** It selects all empty elements.

- **jQuery parent selector :parent**

- **Example:** \$("":parent")
- **Description:** It selects all elements that are a parent of another element.

- **jQuery hidden selector**

:hidden

- **Example:** \$("p:hidden")
- **Description:** It selects all hidden <p> elements.

- **jQuery visible selector**

:visible

- **Example:** \$("table:visible")
- **Description:** It selects all visible tables.

- **jQuery root selector**

:root

- **Example:** \$("":root")
- **Description:** It selects the document's root element.

- **jQuery language selector**

:lang(language)

- **Example:** \$("p:lang(de)")
- **Description:** It selects all <p> elements with a lang attribute value starting with the specified language code.

- **jQuery attribute selector**

[attribute]

- **Example:** \$("[href]")
- **Description:** It selects all elements with the specified attribute.

- **jQuery attribute value selector**

[attribute=value]

- **Example:** \$("[href='default.htm']")
- **Description:** It selects all elements with the specified attribute value.

- **jQuery attribute not equal selector**

[attribute!=value]

- **Example:** \$("[href!='default.htm']")
- **Description:** It selects all elements with an attribute value not equal to the specified value.

- **jQuery attribute ends with selector**

[attribute\$=value]

- **Example:** \$("[href\$='.jpg']")
- **Description:** It selects all elements with an attribute value ending with the specified value.

- **jQuery attribute starts with selector**

[attribute^=value]

- **Example:** \$("[title^='Tom']")
- **Description:** It selects all elements with an attribute value starting with the specified value.

- **jQuery attribute contains word selector**

[attribute~=value]

- **Example:** \$("[title~='hello']")
- **Description:** It selects all elements with an attribute value containing the specified word.

- **jQuery attribute contains selector**

[attribute*=value]

- **Example:** \$("[title*='hello']")
- **Description:** It selects all elements with an attribute value containing the specified value.

- **jQuery input selector**

:input

- **Example:** \$("input:text")
- **Description:** It selects all input elements.

- **jQuery text input selector**

:text

- **Example:** \$("input:text")
- **Description:** It selects all text input elements.

- **jQuery password input selector**

:password

- **Example:** \$("input:password")
- **Description:** It selects all password input elements.

- **jQuery radio input selector**

:radio

- **Example:** \$("input:radio")
- **Description:** It selects all radio input elements.

- **jQuery checkbox input selector**

:checkbox

- **Example:** \$("input:checkbox")
- **Description:** It selects all checkbox input elements.

- **jQuery submit button selector**

:submit

- **Example:** \$("input:submit")
- **Description:** It selects all submit button input elements.

- **jQuery reset button selector**

:reset

- **Example:** \$("input:reset")
- **Description:** It selects all reset button input elements.

- **jQuery button selector**

:button

- **Example:** \$("input:button")
- **Description:** It selects all button input elements.

- **jQuery image input selector**

:image

- **Example:** \$("input:image")
- **Description:** It selects all image input elements.

- **jQuery file input selector**

:file

- **Example:** \$("input:file")
- **Description:** It selects all file input elements.

- **jQuery enabled selector**

:enabled

- **Example:** \$("input:enabled")
- **Description:** It selects all enabled input elements.

- **jQuery disabled selector**

:disabled

- **Example:** \$("input:disabled")
- **Description:** It selects all disabled input elements.

- **jQuery selected option selector**

:selected

- **Example:** \$("option:selected")
- **Description:** It selects all selected option elements.

- **jQuery checked checkbox selector**

:checked

- **Example:** \$("input:checked")
- **Description:** It selects all checked checkbox input elements.

Conclusion

jQuery selectors are vital tools for web development. They enable developers to precisely target elements within a webpage and manipulate them effectively. By mastering jQuery selectors, developers can enhance the functionality and interactivity of their websites, creating engaging user experiences. Understanding and using these selectors empowers developers to streamline their coding processes and achieve desired outcomes efficiently.

FAQs

1. What are the four 4 valid types of jQuery selectors?

The four valid types of jQuery selectors are:

- Name Selector
- ID Selector
- Class Selector
- Attribute Selector

2. What are the selectors of jQuery?

jQuery offers a wide range of selectors for targeting HTML elements, including:

- Name Selector
- ID Selector
- Class Selector
- Universal Selector
- **jquery multi selector**
- Child Selector
- Descendant Selector
- Sibling Selector
- Attribute Selector
- Form Element Selector
- Visibility Selector
- Content Selector

- Hierarchy Selector
- Filter Selector
- Position Selector
- State Selector

3. How to select a specific element using jQuery?

To select a specific element using jQuery, you can use various selectors such as:

- ID Selector: `$("#elementID")`
- Class Selector: `$(".className")`
- Name Selector: `$("elementName")`
- Attribute Selector: `$("[attribute='value']")`

4. What are the 4 CSS selectors?

The four primary CSS selectors are:

- Type Selector
- Class Selector
- ID Selector
- Universal Selector

5. How many types of selectors are there in jQuery?

There are numerous types of selectors in jQuery, including basic selectors, attribute selectors, child selectors, content filters, visibility filters, and more, offering developers extensive flexibility in targeting HTML elements.

6. How to use XPath in jQuery?

jQuery does not have built-in support for XPath. However, you can use plugins or libraries to incorporate XPath-like selectors in jQuery. Alternatively, you can use native DOM methods or libraries like XPath.js for XPath queries in JavaScript.

Ref Link:

https://www.w3schools.com/jquery/jquery_ref_selectors.asp

<https://www.javatpoint.com/jquery-selectors>

<https://www.geeksforgeeks.org/selectors-in-jquery/>

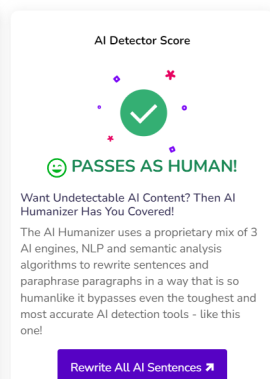
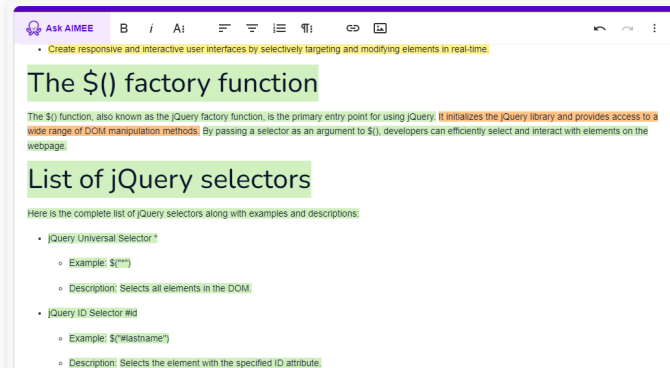
<https://dzone.com/refcardz/jquery-selectors#:~:text=the%20matched%20set-.Types%20of%20jQuery%20Selectors,find%20elements%20within%20the%20DOM.>

The AI Content Detector + AI Humanizer for ChatGPT.

Our AI Checker is trained on blog posts, wikipedia entries, essays, and more articles found online and across multiple large language models (LLM's).

With AI watermarking coming on the scene and the need for a ChatGPT Detector, you should be aware of AI probabilities for your content.

The AI Detector will let you know in seconds if the content you have reads like it is written by a human or if it sounds like it came from an AI system like ChatGPT, GPT-4, Bard, Claude, and Gemini. Then, use the AI Humanizer to produce the most humanlike AI writing possible, giving you undetectable AI content.

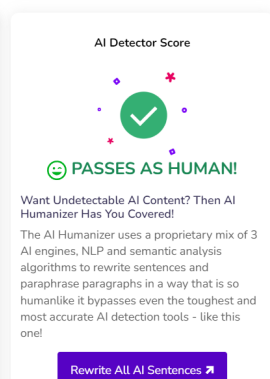
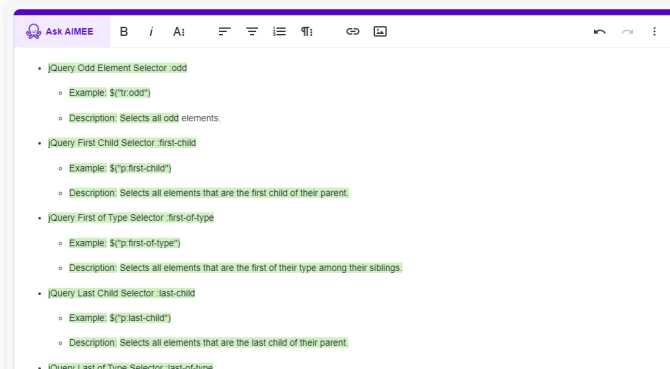


The AI Content Detector + AI Humanizer for ChatGPT.

Our AI Checker is trained on blog posts, wikipedia entries, essays, and more articles found online and across multiple large language models (LLM's).

With AI watermarking coming on the scene and the need for a ChatGPT Detector, you should be aware of AI probabilities for your content.

The AI Detector will let you know in seconds if the content you have reads like it is written by a human or if it sounds like it came from an AI system like ChatGPT, GPT-4, Bard, Claude, and Gemini. Then, use the AI Humanizer to produce the most humanlike AI writing possible, giving you undetectable AI content.



The AI Content Detector + AI Humanizer for ChatGPT.

Our AI Checker is trained on blog posts, wikipedia entries, essays, and more articles found online and across multiple large language models (LLM's).

With AI watermarking coming on the scene and the need for a ChatGPT Detector, you should be aware of AI probabilities for your content.

The AI Detector will let you know in seconds if the content you have reads like it is written by a human or if it sounds like it came from an AI system like ChatGPT, GPT-4, Bard, Claude, and Gemini. Then, use the AI Humanizer to produce the most humanlike AI writing possible, giving you undetectable AI content.

