

Selector in jQuery

The jQuery library exploits the power of CSS (Cascading Style Sheets) Selector to help us quickly and easily access elements or groups of elements in the DOM.

The jQuery library exploits the power of CSS (Cascading Style Sheets) Selector to help us quickly and easily access elements or groups of elements in the DOM.

A jQuery Selector is a function that uses Expressions to find the matching of elements from a DOM on a given standard basis. In a simple way, you can say that the Selector is used to select one or more HTML elements by using jQuery. When an element is selected, we can perform various operations on that selected element.

The \$ () base function in jQuery

Factory Function can translate as base function because for me, it is where everything is created. jQuery Selector starts with a dollar sign and parentheses \$ (). The base function \$ () uses three blocks while selecting elements in a given document:

1. **Tag Name** : Shows a card name available in the DOM. For example: \$ ('p') selects all paragraphs in the element.
2. **Tag ID (Tag ID)**: Represents a tag name available with the given ID in the DOM. The example \$ ('#some-id') selects all single elements in the document that have an ID of some-id.
3. **Class Tag** (class of tag): Represents a tag available with the given class in the DOM. For example \$ ('.Some-class') selects all elements in the document that have a class that is some-class.

All of the above can be used on itself or in conjunction with another Selector. All jQuery Selector builds on the same rules except some tweaks.

Note - The \$ () base function is synonymous with a **jQuery** function (). So in case you are using any other JavaScript library, there may be conflicts here, then you can change the \$ symbol to **jQuery** and you can use the **jQuery ()** function instead of the function. \$ ().

For example

Here is a simple example using Tag Selector. It will select all elements with the tag name p and will set the background color to "yellow".

The jQuery Example

```
class = "myclass" >?ây là ?o?n ??u tiên .
```

```
id = "myid" >?ây là ?o?n th? hai .
```

This is the third paragraph.

It will produce the following result:

```
This is the first paragraph .
```

```
This is the second paragraph .
```

```
This is the third paragraph.
```

How to use Selector in jQuery?

Selector is very useful and will need to require every step while using jQuery. They receive the correct element when you want from your HTML document.

The following table lists the basic selectors and explains them by examples when you click and the corresponding link:

1. **Name:** Select all elements that match the element with the given Name.
2. **#ID:** Select a single element that matches the given ID.
3. **.Class:** Select all elements that match the given Class.
4. **Universal (*):** Select all elements available in a DOM.
5. **Multiple Elements E, F, G:** Select the combined results from all given E, F or G selectors.

Example of Selector in jQuery

Similar to the above syntax and examples, the following examples will help you understand more about the different types of other useful Selectors.

Selector Description \$ (*)

This selector selects all elements in the document.

\$ ("p> *")

This selector selects all elements that are children of a paragraph element.

\$ ("# specialID")

This function Selector receives the element with id = "specialID"

\$ (" . specialClass")

This selector accepts all elements that have a class that is *specialClass* .

\$ ("li: not (.myclass)")

Select all elements matched by the tag

1. without class = "myclass".

\$ ("a # specialID.specialClass")

This selector matches the link with an id being a *specialID* and a class that is *specialClass* .

\$ ("p a.specialClass")

This selector matches a class that is *specialClass* declared in the element

.

\$ ("ul li: first")

This selector only accepts elements

1. first of the element

.

\$ ("# container p")

Select all elements matched by

which is a child of an element with an id that is a *container* .

\$ ("li> ul")

Select all elements matched by

which is the child of an element

1. match.

\$ ("strong + em")

Select all matching elements by *following immediately an element he is matched by* .

\$ ("p ~ ul")

Select all elements matched by

which followed an element he was matched by

.

\$ ("code, em, strong")

Select all elements matched by *or hoc* .

\$ ("p strong, .myclass")

Select all elements matched by **which are children of an element matched by**

as well as all elements that have a class is *myclass* .

\$ (": empty")

Select all elements without child elements.

\$ ("p: empty")

Select all elements matched by

without child elements

\$ ("div [p]")

Select all elements matched by

which contains an element matched by

.

\$ ("p [.myclass]")

Select all elements matched by

which contains an element with a class is *myclass* .

\$ ("a [@rel]")

Select all matching elements by having a *rel* attribute.

\$ ("input [@ name = myname]")

Select all matching elements by having a name value exactly equivalent to *myname*.

\$ ("input [@ tên ^ = myname]")

Select all matching elements by having a name value starting with *myname* .

\$ ("a [@ rel \$ = self]")

Select all elements matched by which the end attribute value is *self* .

\$ ("a [@ href * = domain.com]")

Select all elements matched by which has an href value containing domain.com.

\$ ("li: even")

Select all elements matched by

1. which has an even index value.

\$ ("tr: odd")

Select all matching elements by having an odd index value.

\$ ("li: first")

Select element

1. Firstly.

\$ ("li: last")

Select element

1. final.

\$ ("li: visible")

Select all elements matched by

1. which is visible (visible).

\$ ("li: hidden")

Select all elements matched by

1. which is hidden (hidden).

\$ (": radio")

Select all radio buttons in the Form.

\$ (": checked")

Select all checked dialogs in Form.

\$ (": input")

Select only form elements (input, select, textarea, button).

\$ (": text")

Only select text elements (input [type = text]).

\$ ("li: eq (2)")

Select element

1. Tuesday.

\$ ("li: eq (4)")

Select element

1. Thursday.

\$ ("li: lt (2)")

Select all elements matched by

1. before the second element; In other words, select two elements
2. Firstly.

\$ ("p: lt (3)")

Select all elements matched by

before the fourth element; In other words, choose three elements

Firstly.

\$ ("li: gt (1)")

Select all elements matched by

1. after the second element.

\$ ("p: gt (2)")

Select all elements matched by

after the third element.

\$ ("div / p")

Select all elements matched by

which is a child of an element matched by

.

\$ ("div // code")

Select all elements matched by mà là con c?a m?t ph?n t? ???c so kh?p b?i

.

\$ ("// p // a")

Select all elements matched by which are children of an element matched by

.

\$ ("li: first-child")

Select all elements matched by

1. which is the first child of the parent element.

\$ ("li: last-child")

Select all elements matched by

1. which is the last child of the parent element.

`$ (": parent")`

Select all elements that are the parent of another element, including text.

`$ ("li: contains (second)")`

Select all elements matched by

1. which contains the text is second.

You can use all of these selectors with any HTML / XML element in the general way. For example, if the \$ Selector ("li: first") works for the element

1. then, \$ ("p: first") will also work for the element

.

According to Tutorialspoint

Previous article: [Basic about jQuery](#)

Next article: [Properties in jQuery](#)

You finished reading the article "**Selector in jQuery**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.