

2022_04_01

April 1, 2022

1 2022-04-01

1.0.1 Corso ITS

1.1 Magento & e-commerce software

1.2 ### JavaScript (Andrea Ribuoli)

1.3 # jQuery

1.4 il selettore all

`$("*")`

Probabilmente non verrà mai utilizzato... ma esiste.

1.5 il selettore di un tipo elemento HTML

`$("h1")`

Restituisce gli elementi DOM del tipo specificato.

1.6 il selettore di una classe di elementi

`$(".la-mia-classe")`

Restituisce gli elementi DOM accomunati da essere marcati con un particolare attributo **class** (specificato dopo il carattere .).

1.7 il selettore id

`$("#il-mio-id")`

Restituisce l'elemento DOM avente l'**id** specificato (dopo il carattere #)

1.8 il selettore multiplo

```
$("#h1, .la-mia-classe, #il-mio-id")
```

Restituisce **tutti** gli oggetti che corrispondono ad uno o più dei selettori elencati (**OR**).

1.9 I selettori first (primo) e last (ultimo)

```
$("#p:first")
```

```
$("#h2:last")
```

Restituisce la prima o l'ultima occorrenza del tipo elemento indicato.

1.10 Il selettore eq

```
$(".la-mia-classe:eq(3)")
```

Restituisce la n-esima occorrenza di un elemento nel DOM (l'indice del primo elemento è **0**)

1.11 I selettori even (pari) e odd (dispari)

```
$("#td:even")
```

```
$("#p:odd")
```

Restituisce gli elementi con indice pari o dispari: i pari sono il primo, il terzo, il quinto!

1.12 Il selettore contains

```
$("#p:contains('text')")
```

Seleziona gli elementi DOM che contengono una determinata sotto-stringa

1.13 Il selettore not

```
$("#p:not('.ignorami')")
```

Da utilizzarsi per impostare un criterio che **non** deve essere soddisfatto perchè un elemento DOM sia selezionato. Nell'esempio vengono esclusi i paragrafi (**p**) della classe **ignorami**.

1.14 Il selettore attributo uguale a

```
$("input[type='text']")
```

```
$("[href='https://google.com']")
```

Seleziona tutti gli elementi DOM che soddisfano **anche** (AND) la condizione sull'attributo (specificata tra parentesi quadre).

1.15 Il selettore attributo diverso da

```
$("a[href!='https://google.com']")
```

Come il precedente, con la condizione circa l'attributo negata.

1.16 Il selettore esiste l'attributo

```
$("button[onclick]")
```

```
$("[href]")
```

La condizione è che l'attributo esista, indipendentemente dal valore assegnato.

1.17 Il selettore l'attributo contiene

```
$("a[href*='google.com']")
```

Gli elementi selezionati hanno l'attributo indicato con un valore che include la sottostringa specificata.

1.18 I selettori l'attributo inizia per e l'attributo finisce con

```
$("[href^='https://']")
```

```
$("[href$='.html']")
```

Come il precedente con il vincolo che la sottostringa specificata si trovi all'inizio o alla fine del valore per l'attributo.

1.19 L'uso combinato di più selettori

```
$("input[type='text'][class*='name']")
```

Le diverse condizioni vengono poste in AND logico (gli elementi DOM vengono selezionati se soddisfano **tutte** le condizioni). Nell'esempio vengono selezionati gli elementi DOM di tipo **input** per cui contemporaneamente:

- sia specificato l'attributo **type** con valore **text**
- abbiano un nome classe che include la sottostringa **name**

2 il metodo append

```
$("#p").append("<span>Fine</span>")
```

Aggiunge quanto specificato alla fine del contenuto dell'elemento DOM.

3 il metodo prepend

```
$("#p").prepend("<span>C'era una volta,</span>")
```

Aggiunge quanto specificato all'inizio del contenuto dell'elemento DOM.

4 il metodo before

```
$("#p").before('<h1>Intestazione</h1>')
```

Per inserire un elemento prima degli elementi DOM selezionati.

5 il metodo after

```
$("#p").after('<h1>Questo è tutto!</h1>')
```

Per inserire un elemento dopo gli elementi DOM selezionati.

6 il metodo wrap

```
$("#p").wrap("<div class='wrap'></div>")
```

Per avvolgere **ciascun elemento** selezionato nella struttura HTML indicata (**div** nell'esempio)

7 il metodo wrapAll

```
$("#p").wrapAll("<div class='wrap'></div>")
```

Per avvolgere **tutti gli elementi** selezionati in una unica struttura HTML indicata (**div** nell'esempio)

8 i metodi prependTo, appendTo, insertBefore, eccetera

```
$("<span>Fine</span>").appendTo("p")
```

```
$("<span>C'era una volta, </span>").prependTo("p")
```

```
$('<h1>Intestazione</h1>').insertBefore("p")
```

```
$('<h1>Questo è tutto!</h1>').insertAfter("p")
```

Producono gli stessi effetti dei precedenti ma adottano una notazione **reverse**.

9 il metodo remove

```
$(".rimuovimi").remove()
```

Utilizzabile per rimuovere gli elementi selezionati dal DOM

10 il metodo detach

```
$(".spostami-all-inizio")  
  .detach()  
  .insertBefore("p:first")
```

Rimuove un elemento dal DOM e lo reinserisce in una diversa posizione. Si noti l'uso della notazione **reverse**.

11 il metodo empty

```
$('.content').empty()
```

Rimuove tutto il contenuto all'**interno** degli elementi selezionati.

12 il metodo unwrap

```
$('.elementi-impacchettati').unwrap()
```

Per rimuovere gli elementi che avvolgono gli elementi selezionati.

13 il metodo `replaceWith`

```
$('.rimpiazzami').replaceWith("<h1>Sostituito!</h1>")
```

Utilizzabile per rimpiazzare gli elementi selezionati con un diverso contenuto.

14 il metodo `replaceAll`

```
$("<h1>Sostituito!</h1>").replaceAll('.rimpiazzami')
```

La sintassi è invertita rispetto al *replaceWith*.

15 il metodo `attr`

```
$("#un-certo-id").attr("id")
```

```
$("#un-certo-id").attr("id", "un-nuovo-id")
```

Utilizzabile per estrarre un attributo o, eventualmente, per modificarlo.

16 il metodo `prop`

```
$("#il-mio-input").prop("value")
```

```
$("#il-mio-input").prop("value", "Nuovo valore assegnato!")
```

Utilizzabile per estrarre una proprietà o, eventualmente, per modificarla.

17 i metodi `removeAttr` e `removeProp`

```
$("p").removeAttr("class")
```

```
$("p").removeProp("value")
```

Utilizzabili per rimuovere completamente un dato attributo o, rispettivamente, proprietà (in pratica portandole ad essere **undefined**).

18 il metodo `css`

```
$(".error").css("color", "red")
```

Per modificare il valore della proprietà CSS indicata.

19 il metodo **on**

20 il metodo **trigger**

21 il metodo **off**

22 il metodo **ready**

As of jQuery 3.0, only the first syntax is recommended; the other syntaxes still work but are deprecated.

23 il metodo **resize**

24 il metodo **scroll**

25 il metodo **click**

26 il metodo **dblclick**

27 il metodo **mouseenter**

28 il metodo **mouseleave**

29 il metodo `keydown`

30 il metodo `keyup`

31 il metodo `keypress`

32 il metodo `focus`

33 il metodo `change`

34 il metodo `val`

35 il metodo `select`

36 il metodo `blur`

37 il metodo `submit`

38 il metodo `hide`

39 il metodo `show`

40 il metodo `toggle`

41 il metodo `animate`

42 il metodo `delay`

43 il metodo `queue`

44 il metodo `stop`

45 il metodo `fadeOut`

46 il metodo `fadeIn`

47 il metodo `fadeToggle`

48 il metodo `fadeTo`

49 il metodo `slideUp`

50 il metodo `slideDown`

51 il metodo `slideToggle`

52 il metodo `height`

53 il metodo `width`

54 il metodo `innerHeight`

55 il metodo `innerWidth`

56 il metodo `outerHeight`

57 il metodo `outerWidth`

58 il metodo `find`

59 il metodo `children`

60 il metodo `parent`

61 il metodo `parents`

62 il metodo **siblings**

63 il metodo **each**

64 il metodo **filter**

65 il metodo **has**

66 il metodo **is**
