

Esempi JQuery

- Esempi JQuery
 - SELETTORI
 - FILTRI
 - FILTRI SU TABELLE
 - FILTRI SU MODULI
 - checkbox
 - option
 - manipolare il DOM
 - prepend()
 - before()
 - append()
 - after()
 - html()
 - addClass()
 - removeClass()
 - html() + val()
 - val()
 - .css()
 - ATTRIBUTI SET & GET
 - GETTERS
 - SETTERS
 - GLI EVENTI
 - DOM ready() Eseguire una funzione solo al caricamento della pagina
 - eventi legati al mouse
 - eventi legati alla pressione di tasti
 - GLI EFFETTI
 - slideUp()
 - slideDown()
 - Ottenere un effetto Apri/chiedi

- Gestire la velocità dell'animazione
- Eseguire un'azione al completamento dell'animazione
- ANIMATE
- Scroller di pagina
- Scroller di pagina verticale
- Scroller di pagina orizzontale
- SLIDER DI IMMAGINI
- VALIDARE i moduli
- Implementiamo il form
- FORM VALIDATION: LOGICA

SELETTORI

I Selettori in jquery permettono di prendere il controllo di uno o più oggetti di una pagina web, per selezionare un elemento si usa la funzione \$ a cui viene passato un "qualcosa", qui sotto alcuni esempi di Selettori.

- Selettore di ID => \$("#blocco")
- Selettore di Classe => \$(".elenco")
- Selettore tramite nome TAG => \$("p")
- Selettore tramite attributi => \$("[title*= titolo]")

```
<body>
<div id="blocco"> <!-- $('#blocco') -->
<p><strong>ELENCO:</strong> III C </p> $("strong")
$("p strong")
<ul class="elenco"> <!-- $('.elenco') -->
<li>PAOLO</li>
<li>PIETRO</li>
<li>CARLO</li>
<li>YASMINE</li>
</ul>
<strong>Fine elenco</strong> $("strong")
</div>
```

FILTRI

I filtri permettono di ottenere una maggiore precisione nella selezione degli elementi fatta con i selettori

- Filtri semplici (Basic Filter)
 - :first
 - :last
 - :not()
 - :eq()
- Filtri sul contenuto (Content Filter)
 - :empty
 - :contains(text)
 - :has(selector)
- Filtri sugli attributi (Attribute)
 - [attribute]
 - [attribute=value]
 - [attribute!=value]
- Filtri sugli elementi dei form (Form)
 - :input
 - :text
 - :submit
 - :disabled
 - :checked

FILTRI SU TABELLE

Cognome	Provincia	#
MAURO	TO	presente
PAOLO	TO	assente
CARLO	MI	presente
PIETRO	MI	presente
YASMINE	RM	assente

- `$("#th").css("background-color", "#9bbb59");`
- `$("#tr:odd").css ("background-color", "#dee7d1");`
- `$("#tr:even").css ("background-color", "#eff3ea");`
- `$("td:contains('assente')").css('color','red');`
- `$("td[align=left]").css('text-align','center');`

FILTRI SU MODULI

checkbox

```
<form name="nome_form" id="id_form">
<label>Colori preferiti</label>
<input type="checkbox" value="rosso" /> Rosso
<input type="checkbox" value="giallo" /> Giallo
<input type="checkbox" value="blu" /> Blu
<input type="button" value="sono selezionati..."
onclick="dammi_selezionati()" />
</form>

<script type="text/javascript">
function dammi_selezionati(){
    var stringa='';
    $(".checked").each( function() {
        stringa+=" "+ $(this).val() +" ";
    });
    alert(stringa);
}
```

```
}  
</script>
```

option

```
<form name="nome_form" id="id_form">  
<label>Colori preferiti</label><br />  
<select id="colori" multiple="multiple">  
<option value="rosso">Rosso</option>  
<option value="blu">Blu</option>  
<option value="giallo">Giallo</option>  
</select>  
<input type="button" onclick="dammi_selezionato()"   
value="Dammi selezionati">  
</form>  
<script type="text/javascript">  
function dammi_selezionato() {  
    var stringa='';  
    $("#colori option:selected").each( function()   
{  
        stringa+=" "+ $(this).val() + " ";  
    });  
    alert(stringa);  
} ;  
</script>
```

manipolare il DOM

Alcune funzioni che permettono di manipolare il DOM,
per un elenco completo vedi la documentazione su
<http://api.jquery.com/category/manipulation/>

prepend()

```
<div id="blocco">
<h1>Esempio</h1>
<div id="blocco_interno"></div>
</div>

$("h1").prepend("<span>Primo</span>");
```

before()

```
<div id="blocco">
<h1>Esempio</h1>
<div id="blocco_interno"></div>
</div>

$("h1").before("<span>Primo</span>");
```

append()

```
<div id="blocco">
<h1>Esempio</h1>
<div id="blocco_interno"></div>
</div>

$("h1").append("<span>Primo</span>");
```

after()

```
<div id="blocco">
<h1>Esempio</h1>
<div id="blocco_interno"></div>
</div>

$("h1").after("<span>Primo</span>");
```

html()

```
<div id="blocco">
<h1>Esempio</h1>
<div id="blocco_interno"></div>
</div>

$("#blocco_interno").html("<strong>Ciao , sono un
testo dinamico</strong>");
```

addClass()

```
<div id="blocco">
<h1>Esempio</h1>
<div id="blocco_interno"></div>
</div>

$("#blocco_interno").addClass("selezionata");
```

removeClass()

```
<div id="blocco">
  <h1>Esempio</h1>
  <div id="blocco_interno" class="selezionata">
  </div>
</div>

$("#blocco_interno").removeClass("selezionata");
```

html() + val()

```
<form>
  <input type="text" id="testo" name="testo"
  value="ciao" />
</form>
<span></span>

$("#span").html($("#testo").val());
```

val()

```
<form>
  <input type="text" id="testo" name="testo" />
</form>
<span></span>

$("#testo").val('Hello world') ;
```

.css()

Possiamo gestire le regole css di un elemento attraverso la funzione .css

- sintassi: \$("#elemento).css (regola-css , valore);
- \$("#elemento).css("color", "red");
- \$("#elemento).css("height", "100px");
- \$("#elemento).css("border", "1px solid red");
- \$("#elemento).css("background-color", "#ffcc90");

```
//accorpare le regole in un oggetto
$("#elemento").css( {
regolaCss : "valore", //sintassi camelCase per la
proprietà
regolaCss : "valore",
regolaCss : "valore" //l'ultimo lemento senza virgola
} );
```

```
//accorpare le regole in un oggetto
$("#elemento").css({ "color": "red",
"height" : "100px",
"border" : "1px solid red",
"background-color" : "#ffcc90"
});
```

ATTRIBUTI SET & GET

Con JQuery possiamo leggere e impostare i valori per ogni attributo del DOM, ad esempio:

- leggere e scrivere dinamicamente dentro un elemento p
- leggere e scrivere il contenuto di un input text

- leggere e scrivere una proprietà CSS
-

GETTERS

- `.html()`
 - `.attr('id')`
 - `.val()`
 - `.width()`
-

SETTERS

- `.html('<p>ciao</p>')`
- `.attr('id','blocco_testo')`
- `.val('ciao')`
- `.width('120px')`

GLI EVENTI

Con jquery possiamo intercettare alcuni eventi e dar seguito ad una operazione quando questi si verificano

- <http://api.jquery.com/category/events>
- Click
- Doppio click
- Focus
- Pressione di un tasto
- Caricamento
- Mouse
- Selezione

- Caricamento della pagina
- Ridimensionamento

Struttura di una funzione per intercettare un evento Quando su questo/i elemento/i Si verifica questo evento

```
$("button").click(function() {  
  // qualche operazione  
});
```

```
<html>  
<head>  
<script src="percorso/jquery.js"></script>  
<script style="text/javascript">  
  $('#pulsante').click(function() {  
    $('#box').slideUp()  
  }  
</script>  
</head>  
<body>  
<input type="button" value="cliccami" id="pulsante"  
>  
<div id="box" style="width:100px; height:100px;  
background-color:red"></div>  
</body>  
</html>
```

DOM ready() Eseguire una funzione solo al caricamento della pagina

```
<head>
```

```
<script type="text/javascript">
$(document).ready(function() {
// fai qualcosa
});
</script>
</head>
```

```
<head>
<script type="text/javascript">
$(function() {
// fai qualcosa
});
</script>
</head>
```

eventi legati al mouse

- mouseover
- mouseout
- mousedown
- mouseleave
- ...

```
<html>
<head>
<script src="jquery.js"></script>
<script type="text/javascript">
$(function() {
$('p').mouseover(function() {
alert('Ciao, sono una finestra attivata da un
evento');
});
})
</script>
```

```
</head>
<body>
<p>Passa con il mouse sopra la frase</p>
</body>
</html>
```

eventi legati alla pressione di tasti

- keydown
- keyup
- keypress
- focusin,
- focusout
- <http://api.jquery.com/category/events/keyboard-events/>

```
<html>
<head>
<script src="jquery.js"></script>
<script type="text/javascript">
$(function() {
$('#parola').keypress(function(evento) {
var keycode = (evento.which ? evento.which :
evento.keyCode);
$('#p').append(' '+evento.which+'
'+String.fromCharCode(keycode));
});
})
</script>
</head>
<body>
<input type="text" id="parola" />
<p>Hai premuto i seguenti tasti (in codice ASCII ) :
<br /></p>
</body>
</html>
```

GLI EFFETTI

jQuery ci permettere di realizzare dei semplici effetti animati con pochissime linee di codice. <http://api.jquery.com/category/effects/>

Gli effetti li possiamo raggruppare in 4 tipologie

- FadeIn / fadeOut / fadeToggle
- Hide / Show
- Slide up / Slide Down
- Animate

slideUp()

```
<html>
<head>
<script src="jquery.js"></script>
<script type="text/javascript">
$(function() {
$("#pulsante").click(function () {
$("#box").slideUp();
});
})
</script>
</head>
<body>
<input type="button" value="cliccami" id="pulsante"
/>
<div id="box" style="width: 100px; height: 100px;
background-color: red;"></div>
</body>
</html>
```

slideDown()

```

<html>
<head>
<script src="jquery.js"></script>
<script type="text/javascript">
$(function() {
$("#pulsante").click(function () {
$("#box").slideDown();
});
})
</script>
</head>
<body>
<input type="button" value="cliccami" id="pulsante"
/>
<div id="box" style="width: 100px; height: 100px;
background-color: red;
display:none;"></div>
</body>
</html>

```

Ottenere un effetto Apri/chiudi

```

<html>
<head>
<script src="jquery.js"></script>
<script type="text/javascript">
$(function() {
$("#pulsante").click(function () {
$("#box").slideToggle();
});
})
</script>
</head>
<body>
<input type="button" value="cliccami" id="pulsante"

```

```
/>
<div id="box" style="width: 100px; height: 100px;
background-color: red;"></div>
</body>
</html>
```

Gestire la velocità dell'animazione

```
<html>
<head>
<script src="jquery.js"></script>
<script type="text/javascript">
$(function(){
$("#pulsante").click(function () {
$("#box").slideToggle(3000);
});
})
</script>
</head>
<body>
<input type="button" value="cliccami" id="pulsante"
/>
<div id="box" style="width: 100px; height: 100px;
background-color: red;"></div>
</body>
</html>
```

Eseguire un'azione al completamento dell'animazione

```
<html>
<head>
```



```

<script src="jquery.js"></script>
<script type="text/javascript">
$(function() {
$("#pulsante").click(function () {
$("#box").slideToggle(1000, function() {
alert("Animazione completata");
} );
});
})
</script>
</head>
<body>
<input type="button" value="cliccami" id="pulsante"
/>
<div id="box" style="width: 100px; height: 100px;
background-color: red;"></div>
</body>
</html>

```

ANIMATE

Con la funzione animate, si possono creare "transizioni" sulle proprietà css di tipo numerico di un oggetto.

```

<script type="text/javascript">
$(function() {
$("#pulsante").click(function () {
$('#box').animate({left: '+=500' },
1000,
function() { alert("Animazione completata"); }
);
});
}) ;
</script>
</head>

```

```
<body>
<input type="button" value="cliccami" id="pulsante"
/>
<div id="box" style="width: 100px; height: 100px;
background-color: red;position:
relative; "></div>
```

Scroller di pagina

Lo scroller di pagina viene usato per spostarsi velocemente tra i contenuti di una pagina web. Un tipico esempio sono le ancore "torna su" che troviamo alla fine di un testo che permettono con un click di tornare in cima allo stesso

Scroller di pagina verticale

```
<script type="text/javascript">
function vai_a(id)
{
$('html,body').animate({scrollTop:
$("#"+id).offset().top},1000);
}
</script>
<div id="blocco_1" class="blocco" style=" background-
color: red;">
<div class="menu">
<a href="#" onclick="vai_a('blocco_1');return false;"
>Vai al blocco 1</a>
<a href="#" onclick="vai_a('blocco_2');return false;"
>Vai al blocco 2</a>
<a href="#" onclick="vai_a('blocco_3');return false;"
>Vai al blocco 3</a>
</div>
Testo Primo Blocco 1
</div>
```

Scroller di pagina orizzontale

```
<script type="text/javascript">
function vai_a(id)
{
$('html,body').animate({ scrollLeft: :
$("#"+id).offset().left},1000);
}
</script>
<div id="blocco_1" class="blocco" style=" background-
color: red;">
<div class="menu">
<a href="#" onclick="vai_a('blocco_1');return false;"
>Vai al blocco 1</a>
<a href="#" onclick="vai_a('blocco_2');return false;"
>Vai al blocco 2</a>
<a href="#" onclick="vai_a('blocco_3');return false;"
>Vai al blocco 3</a>
</div>
Testo Primo Blocco 1
</div>
```

SLIDER DI IMMAGINI

```
<script>
$(function(){
$('#blocco img:gt(0)').hide();
setInterval(function(){ $('#blocco :first-
child').fadeOut(20).next('img').
fadeIn(700).end().appendTo($('#blocco'));;}, 3000);
});
</script>
<style type="text/css">
.blocco{ width: 500px; height:350px; overflow:
```

```
hidden;}
.blocco img{ width: 500px; height:350px; float:
left;}
</style>
<div id="blocco" class="blocco">



</div>
```

VALIDARE i moduli

Molto utili per guidare l'utente nella compilazione e segnalare degli errori.

Implementiamo il form

```
<li>
<label>Nome</label>
<input type="text" id="nome" name="nome"
class="obbligatorio" />
</li>
<li>
<label>Indirizzo</label>
<input type="text" id="indirizzo" name="indirizzo" />
</li>
<li>
<label>Email</label>
<input type="text" id="email" name="email"
class="obbligatorio" />
</li>
<li>
<input type="button" id="invia" name="invia"
value="invia" onclick="controlla()" />
</li>
```

FORM VALIDATION: LOGICA

```
function controlla(){
$.obbligatorio ".each( function() {
if( ($(this).val()=="") ){
errore=true;
$(this).prev().addClass('rosso');
}
else{ $(this).prev().removeClass('rosso'); }
})
if(errore==true)
{ alert("Il form contiene errore/i, si prega di
correggere."); }
else
{alert("Complimenti! Non ci sono errori!"); }
}
</script>
```