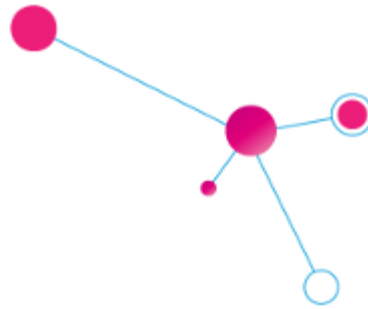




Introduction à jQuery





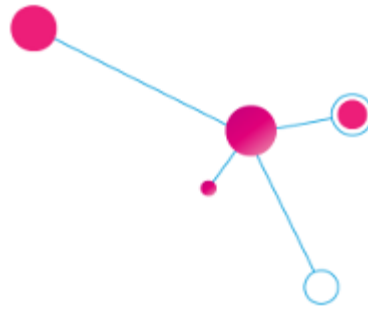
Qu'est-ce que jQuery ?

jQuery est une bibliothèque JavaScript libre et multiplateforme créé pour faciliter le scripting coté client pour manipuler de l'HTML et du CSS.



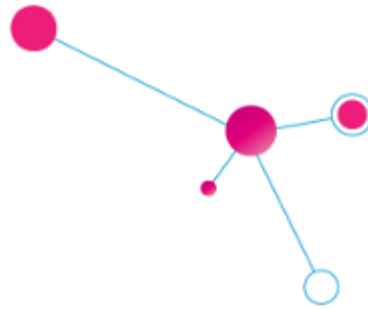
À quoi ça sert ?

- une approche '**Write less, do more**'
- Parcourir et modifier le DOM
- Manipuler et gérer les événements
- Mettre en place des effets visuels et des animations
- Manipuler des feuilles de style CSS
- Ajax
- Utilitaires



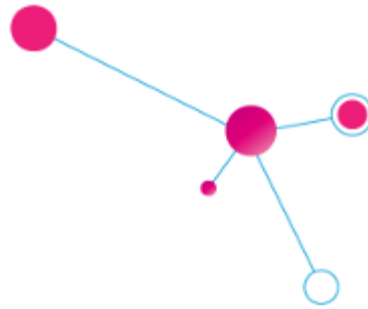
Commencer avec jQuery

- Inclure jQuery dans votre document avant la balise `</body>`
- En local : `<script src="js/jquery.js"></script>`
- Une bibliothèque hébergée :
 - `<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>`

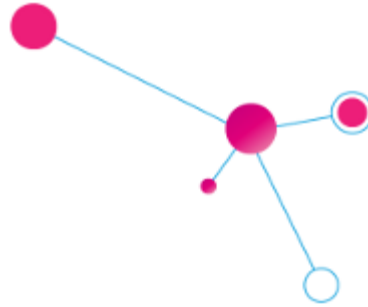


Commencer avec jQuery

- Les fonctions jQuery devraient s'exécuter uniquement une fois le DOM (la structure HTML) de votre page chargé. Pour s'en assurer, on les englobe dans une fonction globale qui s'active uniquement quand le DOM est chargé :
- ```
$(document).ready(function() {
 // YOUR CODE GOES HERE
});
```



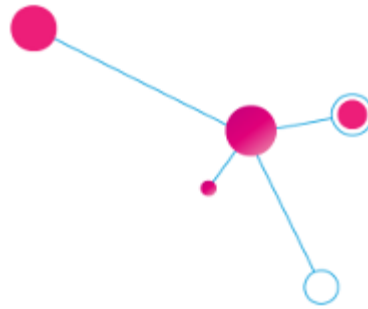
# Quelques exemples de jQuery



- `$( 'div' ).hide( );`
- `$( '#form' ).addClass( 'alert' );`
- `$( '.address-box' ).css( 'background-color', 'blue' );`
- `$( 'button' ).click( function( ) {  
    $( 'div' ).show( );  
} )`



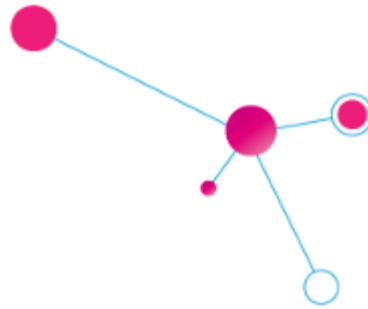
# L'enchaînement des méthodes jQuery



- Les méthodes jQuery peuvent s'enchaîner pour faire plusieurs actions sur le le même élément.
- `$( 'div' ).css( 'background-color', 'brown' ).fadeIn( ).append( '<p>Success!</p>' );`



# La fonction jQuery()



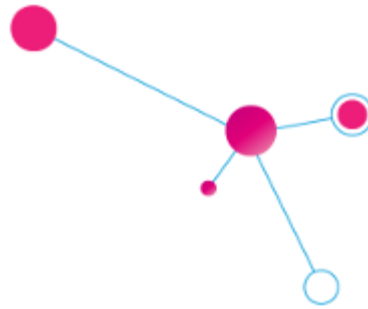
- La fonction jQuery() peut s'écrire aussi comme \$(). Elle est la fonction de base de jQuery et elle permet d'accéder à toutes les méthodes de la bibliothèque.





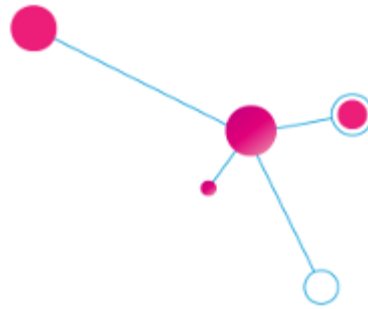
# La fonction jQuery()

- Sélectionner des éléments avec les sélecteurs CSS
  - `jQuery( '#mon_id' )` OU `$( '#mon_id' )`
  - `jQuery( '.ma_classe' )` OU `$( '.ma_class' )`
  - `jQuery( 'div' )` OU `$( 'div' )`
- Appeler des méthodes sur les éléments sélectionnés :
  - `$( '#mon_id' ).addClass( "blue" );`
  - `$( 'p' ).hide();`
- Appeler d'autres méthodes jQuery :
  - ```
$.get( "test.php", function( data ) {  
    alert( "Data Loaded: " + data );  
});
```



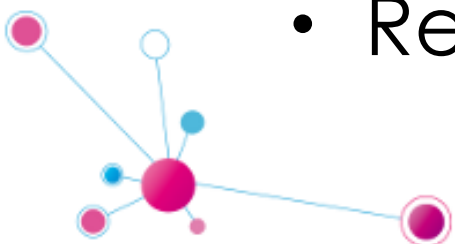
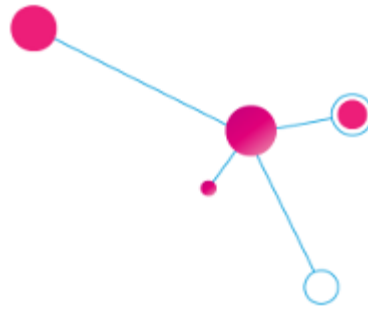
La fonction jQuery()

- Sélectionner des éléments avec les sélecteurs CSS
 - `jQuery('#mon_id')` OU `$('#mon_id')`
 - `jQuery('.ma_classe')` OU `$('.ma_class')`
 - `jQuery('div')` OU `$('div')`
- Appeler des méthodes sur les éléments sélectionnés :
 - `$('#mon_id').addClass("blue");`
 - `$('p').hide();`
- Appeler d'autres méthodes jQuery :
 - ```
$.get("test.php", function(data) {
 alert("Data Loaded: " + data);
});
```



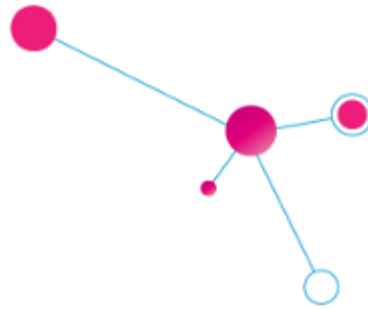
# Exercises

- Télécharger les documents HTML via ce lien : <http://cours.marknightingale.net/files/jquery/exo1.zip>
- Avec jQuery et en ajoutant et supprimant des classes :
  - Rendre le premier div bleu
  - Rendre le deuxième div vert
  - Rendre le premier div bleue, le deuxième vert, le troisième jaune
  - Rendre tous les trois divs de test jaune



# Les méthodes de jQuery

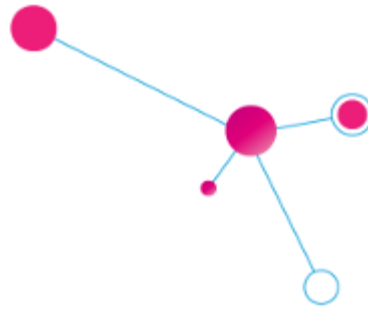
- La documentation de jQuery est très complète : <https://api.jquery.com>
- Quelques méthodes utiles :
  - .hide()
  - .show()
  - .toggle()
  - .html()
  - .text()
  - .val()
  - .append()
  - .prepend()
  - .css()
  - .attr()
  - .addClass()
  - .removeClass()
  - .toggleClass()



## `.hide()`, `.show()`, `.toggle()`

Les méthodes `.hide()`, `.show()` et `.toggle()` permettent d'afficher ou cacher des éléments du DOM. En effet ces méthodes définissent simplement des styles inline sur la propriété `'display:'`.

Toggle permet d'alterner entre `hide()` et `show()` en fonction de l'état actuel de l'élément : si l'élément est actuellement caché il sera affiché, et vice-versa.



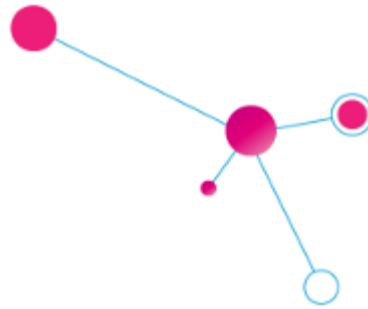
## .html(), .val(), .text()

Ces méthodes permettent de récupérer et définir le contenu d'un élément.

`var contenu = $('div').html()` > récupère le contenu entier d'un div (HTML, texte etc.).

`var text = $('div').text()` > récupère le text du div (sans les balises HTML)

`var val = $('input').val()` > récupère la valeur d'un champ input



## .html(), .val(), .text()

Les méthodes peuvent également être utilisées pour définir ces mêmes attributs

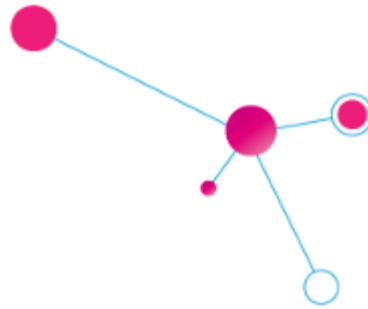
`$( 'div' ).html( '<p>Paragraphe</p>' )` > défini le contenu du div (on remplace le contenu avec un `<p>`)

`$( 'div' ).text( 'Paragraphe' )` > Remplace le contenu du div avec le text 'Paragraphe'.

**N.B** `$( 'div' ).text( '<p>Paragraphe</p>' )` écrira `<p>Paragraphe</p>` à l'intérieur du div, les `<p></p>` étant affichés à l'écran en tant que text et pas du HTML

`$( 'input' ).val( 'contact@strasweb.fr' )` > remplace la valeur d'un champs input par contact@strasweb.fr

# .append() et .prepend()



Lorsque `.html()` remplace l'HTML d'un élément, ces deux méthodes permettent d'insérer du HTML (append après le contenu existant, prepend avant).





## .css()

`.css()`

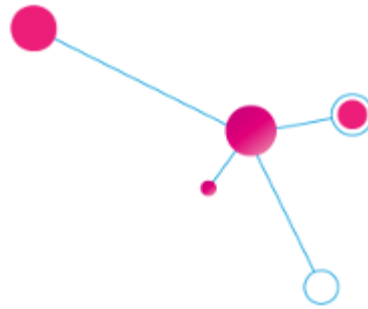
permet de récupérer ou définir la valeur d'une propriété CSS d'un élément

```
var backgroundColor = $('div').css('background-color')
```

Récupère la couleur de fond d'un élément

```
$('div').css('background-color', 'green')
```

défini la couleur de fond comme vert



## .attr()

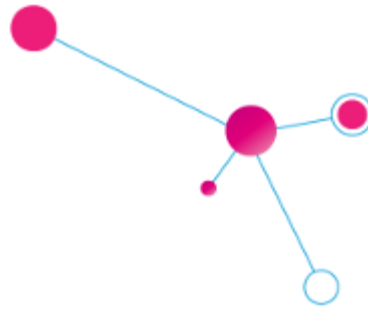
.attr() permet de récupérer ou définir un attribut HTML d'un élément.

```
var imgSrc = $('img').attr('src')
```

récupère l'URL de l'image

```
$('img').attr('src', 'http://mondomaine.com/
image.png')
```

définir l'URL de l'image (et change donc l'image affichée).



# `.addClass()`, `.removeClass()`, `.toggleClass()`



Permet d'ajouter, enlever ou toggle les classes d'un élément.

`.toggleClass()` fonctionne de la même manière que `.toggle()`



# Les gestionnaires d'événements jQuery



- jQuery permet aussi de gérer les événements Javascript, et d'y associer des fonctions :
- ```
$( '.button' ).click(function() {  
    // Do something on click here like change  
    the color  
    $( '.text' ).css( 'color', 'red' );  
} );
```



Les gestionnaires d'événements jQuery



- Le syntaxe suivant doit être respecté :

- selecteur fonction 'helper' qui associe à l'événement 'click' sur cet élément

```
$( '.button' ).click( function() {  
    $( '.text' ).css( 'color', 'red' );  
} ) ;
```

fonction qui contient les
instructions de ce qu'il faut faire
quand on clique

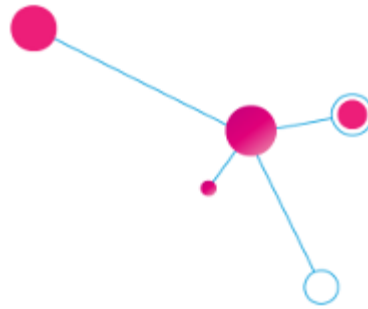


Les gestionnaires d'événements jQuery

- Quelques 'event binders' utiles :
- `.click()`
- `.hover()`
- `.ready()`
- `.load()`
- `.focus()`
- `.blur()`
- `.submit()`



Exercises jQuery



Quelques exercices jQuery : <http://www.w3resource.com/jquery-exercises/jquery-core-exercises.php> (en)

Quelques drills jQuery : <http://jqexercise.droppages.com/> (en)

Encore quelques méthodes utiles :

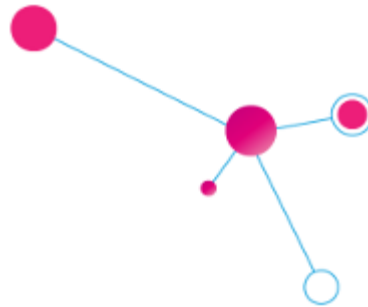
`.clone()`

`.insertAfter()`

`.insertBefore()`



Exercices jQuery



Quelques exercices à faire avec le template de l'Exercice 1 :

- Insérer l'HTML suivant dans le document :
`<button id="mybutton">Hop-là!</button>`
- Avec jQuery créer les fonctions qui permettent :
 - De changer le couleur de fond (de `<body>`) à `red` quand on clique sur le bouton.
 - De cacher tous les divs test quand on clique sur le bouton
 - De supprimer un div quand on clique sur le bouton (méthode `.remove()`)
 - D'ajouter la classe 'green' à tous les divs quand on clique
 - D'enlever toutes les classes quand on clique
 - D'ajouter la classe 'green' à un div quand on le survole (astuce : utilisez le mot-clé `$` (`this`) pour sélectionner l'élément qui a déclenché l'événement.)

