

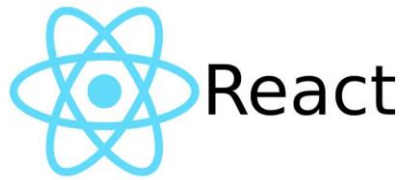


**Iset Djerba**

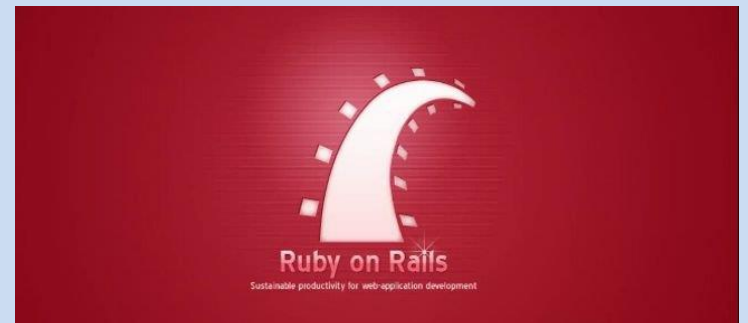
**semestre 01 année 2018-2019**

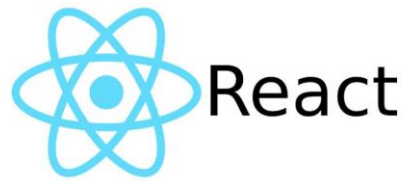
# **Atelier FRAMEWORK Coté Client**

**Enseignant : Béchir HAMIDI**



# Atelier 01 : Introduction





ATELIER 01

django



# Introduction



# I. Définitions



## Que signifie Framework ?

Pourquoi utilise-t-on un FramaWork ?

# FRAMEWORK ?

Des bibliothèques regroupant des fonctions/classes directement utilisables par les programmes afin de réaliser des tâches bien définies

– Objectifs :

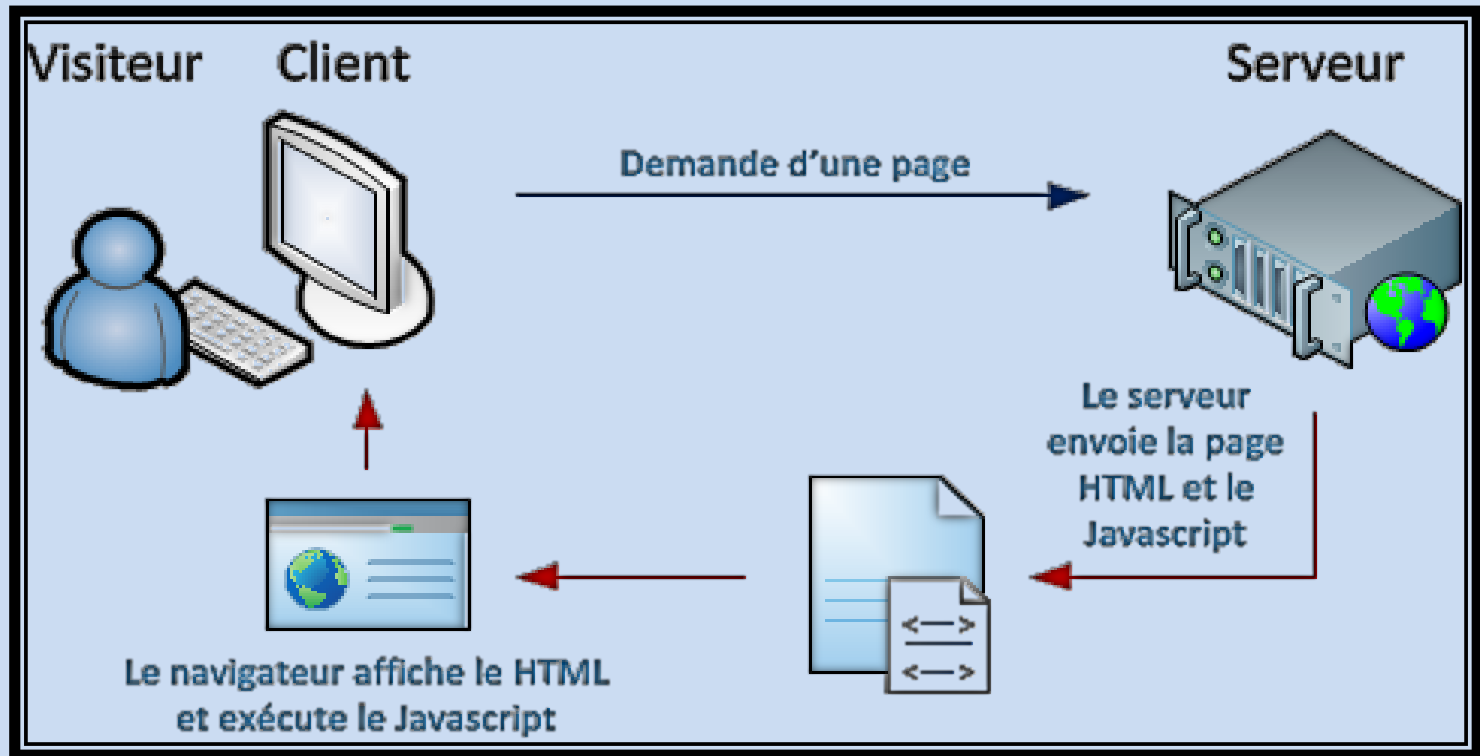
simplifier le travail des développeurs en leur offrant une architecture prête à l'emploi. et qui leur permette de ne pas repartir de zéro à chaque nouveau projet.

# I. Définitions



Programmation Coté Client  
?

# Programmation Coté Client ?





# I. Définitions

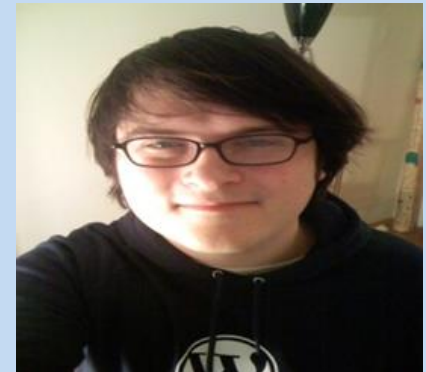


## jQuery?



# jQuery ?

- Bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture des scripts côté client dans le code HTML des pages web.
- Créer en 2006 par John Resig



# jQuery ?

– Permet de :

- Traverser et manipuler très facilement l'arbre DOM des pages web.
- Changer/ajouter une classe CSS, créer des animations, modifier des attributs, etc.
- Gérer les événements **javascript** .
- Envoyer et recevoir des données depuis un serveur grâce à AJAX (asynchronous JavaScript and XML).
- Une grande compatibilité avec les différents navigateurs.

# jQuery ?

## boîte d'outils de jQuery

- Un bon éditeur de texte
- Un navigateur web et des outils de développeur
- La documentation jQuery (<https://api.jquery.com/>)

# jQuery ?

## Inclure jQuery dans un projet

### En local :

Télécharger à partir de [www.jquery.com](http://www.jquery.com)

```
<script src="chemin/jquery.js"></script>
```

### En ligne :

CDN (Content Delivery Network)

```
<script src="http://code.jquery.com/jquery.js "></script>
```

CDN	Version non minimisée
jQuery	<a href="http://code.jquery.com/jquery.js">http://code.jquery.com/jquery.js</a>
Google	<a href="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.js">http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.js</a>
Microsoft	<a href="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.2.js">http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.2.js</a>

## Un document HTML type ressemble finalement à ceci :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page title</title>
    <link rel="stylesheet" href="css/styles.css" />
  </head>
  <body>
    <p>Page content here</p>
    <!-- on inclut jQuery via CDN -->
    <script src="//code.jquery.com/jquery-1.12.0.min.js"></script>
    <!-- votre propre fichier dans le dossier de votre projet -->
    <script src="js/fichier_ou_vous_ecrivez_votre_jquery.js"></script>
  </body>
</html>
```

# jQuery ?

## Syntaxe de base de jQuery

Syntaxe basée sur la fonction **jQuery()**, généralement écrite de manière abrégée **\$()**

**\$(selecteur)** : permet d'accéder aux éléments HTML.

**\$('p')** : permet d'accéder à tous les éléments (P)

# jQuery ?

## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
  </head>
  <body>
    <h1>Découverte de jQuery</h1>
    <p>Ce paragraphe n'apparaîtra pas</p>
    <p>Celui-ci non plus</p>

    <!--On inclut la librairie jQuery-->
    <script src='jquery-3.1.0.js'></script>

    <!--On écrit notre code JavaScript / jQuery-->
    <script>
      $('p').hide();
    </script>
  </body>
</html>
```





# jQuery ?

Attendre le chargement de la page pour  
exécuter le jQuery

# jQuery ?

Attendre le chargement de la page pour  
exécuter le jQuery

Généralement, lorsque l'on code en JavaScript, on attend la fin du chargement de la page pour exécuter les scripts.

**ready** : permet d'attendre la fin du chargement de la page avant l'exécution du script.

# jQuery ?

Attendre le chargement de la page pour  
exécuter le jQuery

```
jQuery(document).ready(function()  
{ // Ici, le DOM est défini });
```

```
$(document).ready(function()  
{// Ici, le DOM est défini });
```

```
$(function() { // Ici, le DOM est défini });
```



# Sélection d'éléments

sélecteurs et modifieurs

**ATELIER 02**

# Rappel du Principe de jQuery

1. Sélectionner une partie du document.
2. Agir dessus

- fonction **jQuery()**
- abrégée **\$()**

## Pour Attendre le chargement de la page

```
$(document).ready(function() { // Ici, le DOM est défini });  
$(function() { // Ici, le DOM est défini });
```

# Sélection d'éléments

Possibilité de sélectionner :

- par type de bloc
- par identifiant
- par classe
- en combinant les critères
- en filtrant sur les noms d'attributs
- en faisant référence aux positions relatives dans le DOM
- en ne récupérant qu'un seul élément parmi les objets sélectionnés
- en filtrant parmi les objets sélectionnés



# Sélection d'éléments

Sélection d'éléments par type de bloc, identifiant ou classe  
(sélecteurs CSS)

Sélecteur CSS	Explication
<code>\$('*');</code>	Cible tous les éléments (inclus <html>, <head> et <body>).
<code>\$('element');</code>	Cible tous les éléments étant de type correspondant.
<code>\$('#element');</code>	Cible l'élément ayant l'identifiant donné (obligatoirement unique).
<code>\$('.element');</code>	Cible tous les éléments ayant la classe donnée.
<code>\$('parent descendants');</code>	Cibles tous les enfants directs ou indirects de l'élément parent.

# Sélection d'éléments

Sélection d'éléments par type de bloc, identifiant ou classe  
(sélecteurs CSS)

# Sélection d'éléments

## Sélection d'éléments par combinaison de critère

<code>\$('element, element2');</code>	Cible d'abord le premier élément trouvé, puis le second, etc.
<code>\$('parent &gt; enfant');</code>	Cible tous les enfants directs de l'élément parent.
<code>\$('frere + element');</code>	Cible tous les éléments précédés directement d'un frère.
<code>\$('frere ~ element');</code>	Cible tous les éléments précédés directement ou indirectement d'un frère.

# Sélection d'éléments

## Sélection d'éléments par Filtre

Filtres	Explication
<code>\$(':animated');</code>	Cible tous les éléments actuellement animés.
<code>\$(':eq(N)');</code>	Cible tous les éléments dont l'index est égal à N.
<code>\$(':gt(N)')</code>	Cible tous les éléments dont l'index est supérieur à N.
<code>\$(':lt(N)')</code>	Cible tous les éléments dont l'index est entre 0 et N-1.
<code>\$(':even')</code>	Cible tous les éléments dont l'index est paire.
<code>\$(':odd')</code>	Cible tous les éléments dont l'index est impair.
<code>\$(':first');</code>	Cible le premier élément trouvé.
<code>\$(':last');</code>	Cible le dernier élément trouvé.
<code>\$(':focus');</code>	Cible tous les éléments ayant actuellement le focus.
<code>\$(':header');</code>	Cible tous les éléments de type header (<h1>, <h2>, <h3>, etc).
<code>\$(':not(selecteur)');</code>	Cible tous les éléments ne correspondant pas au sélecteur.

# Sélection d'éléments

## Sélection d'éléments pour formulaire

Sélecteurs pour formulaire	Explication
<code>\$(':input');</code>	Cible tous les éléments de formulaire.
<code>\$(':button');</code>	Cible tous les éléments de formulaire de type button.
<code>\$(':checkbox');</code>	Cible tous les éléments de formulaire de type checkbox.
<code>\$(':checked');</code>	Cible toutes les boîtes cochées.
<code>\$(':file');</code>	Cible tous les éléments de formulaire de type file.
<code>\$('password');</code>	Cible tous les éléments de formulaire de type password.
<code>\$(':radio');</code>	Cible tous les éléments de formulaire de type radio.
<code>\$(':submit');</code>	Cible tous les éléments de formulaire de type submit.
<code>\$(':text');</code>	Cible tous les éléments de formulaire de type text.

# modifieurs d'éléments

suite à la sélection d'un élément, il devient possible de lui appliquer un traitement en exécutant une méthode jQuery.

les méthodes « getters » (de « get », « obtenir ») et les méthodes « setters » (de « set », « définir »)

certaines méthodes jQuery ont été créées de manière à pouvoir servir à la fois de getter ou de setter selon le type d'arguments qu'on va leur passer.

# modifieurs d'éléments

## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
  </head>
  <body>
    <h1>Manipulation du DOM</h1>
    <span id='test'></span>

    <!--On inclut la librairie jQuery-->
    <script src='jquery-3.1.0.js'></script>

    <!--On écrit notre code JavaScript / jQuery-->
    <script>
      $(document).ready(function(){
        //On récupère la taille de notre h1
        var taille = $('h1').css('font-size');

        //On affiche cette taille dans notre span
        $('#test').text(taille);
      });
    </script>
  </body>
</html>
```

# modifieurs d'éléments

## Exemple





# modifieurs d'éléments

## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
  </head>
  <body>
    <h1>Manipulation du DOM</h1>
    <span id='test'></span>

    <!--On inclut la librairie jQuery-->
    <script src='jquery-3.1.0.js'></script>

    <!--On écrit notre code JavaScript / jQuery-->
    <script>
      $(document).ready(function(){
        //On modifie la taille de notre h1
        $('h1').css('font-size', '50px');

        //On récupère la taille de notre h1 définie ci-dessus
        var taille = $('h1').css('font-size');

        //On affiche cette taille dans notre span
        $('#test').text(taille);
      });
    </script>
  </body>
</html>
```

# modifieurs d'éléments

## Exemple



# modifieurs d'éléments

## Chainage des méthodes

```
$('.rouge').css('background','red');  
$('.rouge').css('color','yellow');
```

```
$('.rouge').css('background','red').css('color','yellow');
```

# modifieurs d'éléments

## Utilisation de la méthode **text()**

peut être utiliser

- en tant que **getter** pour lire le contenu textuel d'un élément.
- en tant que **setter** pour définir un nouveau contenu textuel pour un élément.

La méthode **Text()** permet de lire ou modifier que le contenu textuel d'un élément. Elle ne permet pas l'interprétation des balises contenues dans un élément.

# modifieurs d'éléments

## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
  </head>
  <body>
    <h1>Manipulation du <mark>DOM</mark></h1>
    <span id='test'></span>

    <!--On inclut la librairie jQuery-->
    <script src='jquery-3.1.0.js'></script>

    <!--On écrit notre code JavaScript / jQuery-->
    <script>
      $(document).ready(function(){
        //On récupère le contenu de h1
        var titre = $('h1').text();

        //On modifie ensuite le contenu de h1
        $('h1').text('<mark>DOM</mark> manipulé !');

        //On affiche le contenu de la variable titre
        $('#test').text(titre);
      });
    </script>
  </body>
</html>
```

# modifieurs d'éléments

## Utilisation de la méthode **text()**

### Exemple



**N.B :** à ne pas utiliser la méthode *text()* pour les Formulaire, on utilise la méthode *val()*

# modifieurs d'éléments

## Utilisation de la méthode **html()**

similaire à **text()**, à la différence, quelle permet de manipuler (lire, modifier et ajouter) tous le contenu des éléments y compris les balises HTML ainsi que l'interprétation de ces balais par le navigateur.

# modifieurs d'éléments

## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
  </head>
  <body>
    <h1>Manipulation du <mark>DOM</mark></h1>
    <span id='test'></span>

    <!--On inclut la librairie jQuery-->
    <script src='jquery-3.1.0.js'></script>

    <!--On écrit notre code JavaScript / jQuery-->
    <script>
      $(document).ready(function(){
        //On récupère le contenu de h1 avec html()
        var titre = $('h1').html();

        //On modifie ensuite le contenu de h1
        $('h1').html('<mark>DOM</mark> manipulé !');

        //On affiche le contenu de la variable titre
        $('#test').html(titre);
      });
    </script>
  </body>
</html>
```



# modifieurs d'éléments

## Exemple



# modifieurs d'éléments

## Utilisation de la méthode **val()**

souvent utilisée avec les méthodes relatives aux **formulaire**s ( **keyup()**, **change()** ...) pour récupérer le contenu des éléments de ces **formulaire**s tels que les éléments **input**, **textarea** ou **select**.

**Val()**, renvoie la valeur **null** si aucune option n'a été choisie ou un tableau si plusieurs options ont été choisies.

# modifieurs d'éléments

## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
  </head>
  <body>
    <h1>Manipulation du DOM</h1>
    <label for='prenom'>Entrez un prenom : </label>
    <input type='text' id='prenom' name='prenom'>
    <span id='sp1'></span><br>
    <select multiple='multiple'>
      <option>Option 1</option>
      <option>Option 2</option>
      <option>Option 3</option>
    </select>
    <span id='sp2'></span>

    <!--On inclut la librairie jQuery-->
    <script src='jquery-3.1.0.js'></script>

    <!--On écrit notre code JavaScript / jQuery-->
    <script>
      $(document).ready(function(){
        $('#prenom').keyup(function(){
          var valeurInput = $(this).val();
          $('#sp1').text(valeurInput);
        });

        $('select').change(function(){
          var valeurSelect = $('select').val() || [];
          $('#sp2').text(valeurSelect.join(', '));
        });
      });
    </script>
  </body>
</html>
```

# modifieurs d'éléments

## Exemple



# modifieurs d'éléments

## Utilisation de la méthode **attr()**

la méthode **attr()** peut être utiliser en tant que

- **getter** pour récupérer la valeur d'un attribut d'un élément. Dans ce cas, elle prendre en argument le nom de cette l'attribut.
- **setter**, pour définir des attributs et leurs attribuer une valeur, ou pour modifier les valeurs d'attributs déjà existants. Dans ce cas, la elle prendre en arguments le nom et la valeur de l'attribut. Notez que on peut définir plusieurs attributs à la fois.

```

<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
    <style>        .titre{color: blue;}        </style>
  </head>
  <body>
    <h1 class='titre'>Manipulation du DOM</h1>
    <h2 class='t'>Utilisation de attr()</h2>
    <a>Un lien</a><br>
    <span id='test'></span>
    <!--On inclut la librairie jQuery-->
    <script src='https://code.jquery.com/jquery-3.1.0.js'></script>
    <!--On écrit notre code JavaScript / jQuery-->
    <script>
      $(document).ready(function(){
        //On récupère la valeur de l'attribut class de h1
        var atth1 = $('h1').attr('class');
        //On affiche cette valeur dans notre span
        $('#test').text('Valeur de la class de h1 : ' + atth1);
        //On modifie la valeur associée à l'att class de h2
        $('h2').attr('class', 'titre');

        /*On définit 2 attributs href et target pour a et on leur attribue des valeurs*/
        $('a').attr({
          href: 'http://google.com',
          target: '_blank'
        });
      });
    </script>
  </body>
</html>

```

# modifieurs d'éléments

## Exemple



# modifieurs d'éléments

## Utilisation de la méthode **prop()**

**prop()** est utiliser pour tous les attributs dont la valeur peut être changée de façon dynamique, comme **checked**, **selected** ou **value**.



# modifieurs d'éléments

## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
  </head>
  <body>
    <h1>Manipulation du DOM</h1>
    <label for='c'>Cochez / décochez moi !</label>
    <input type='checkbox' id='c' checked='checked'><br>
    <span id='test'></span>

    <!--On inclut la librairie jQuery-->
    <script src='jquery-3.1.0.js'></script>

    <!--On écrit notre code JavaScript / jQuery-->
    <script>
      $(document).ready(function(){
        $('#c').change(function(){
          var v = $(this);
          $('#test').html(
            'Valeur attribut : ' + v.attr('checked') +
            '<br>Valeur propriété : ' + v.prop('checked')
          );
        });
      });
    </script>
  </body>
</html>
```

# modifieurs d'éléments

## Exemple





# ATELIER 03

## Manipulation du CSS

# Manipulation du CSS

Méthodes	Action
<b>css()</b>	Récupère ou modifie une ou plusieurs propriété(s) CSS.
<b>height()</b>	Récupère la hauteur de l'élément ciblé.
<b>width()</b>	Récupère la largeur de l'élément ciblé.
<b>innerHeight()</b>	Récupère la hauteur de l'élément ciblé en prenant en compte ses marges intérieures mais pas les bordures.
<b>innerWidth()</b>	Récupère la largeur de l'élément ciblé en prenant en compte ses marges intérieures mais pas les bordures.
<b>outerHeight()</b>	Récupère la hauteur de l'élément ciblé en prenant en compte ses marges intérieures, extérieures, et ses bordures.
<b>outerWidth()</b>	Récupère la largeur de l'élément ciblé en prenant en compte ses marges intérieures, extérieures, et ses bordures.

# Manipulation du CSS

Méthodes	Action
<b>offset()</b>	Récupère les coordonnées absolues de l'élément ciblé.
<b>position()</b>	Récupère les coordonnées relatives de l'élément ciblé.
<b>scrollTop()</b>	Récupère la position verticale de la barre de défilement par rapport à la page.
<b>scrollLeft()</b>	Récupère la position horizontale de la barre de défilement par rapport à la page.

Méthodes	Action
<b>addClass()</b>	Ajouter un ou plusieurs attributs <b>class</b>
<b>removeClass()</b>	Retirer un ou plusieurs attributs <b>class</b>
<b>toggleClass()</b>	Ajouter un attribut <b>class</b> , s'il n'existe pas, le supprimer s'il l'est.

# Manipulation du CSS

Exemple addClass(), removeClass() et toggleClass()

```
<!DOCTYPE html>
<html>
  <head>
    <title>style CSS</title>
    <meta charset='utf-8'>
    <style>
      div{border: 1px solid black;}
      .default{font-size: 30px}
      .centre{text-align: center;}
      .droite{text-align: right;}
      .style01{background-color: red;font-weight: bold;}
      .style02{background-color: blue;font-style: italic;};
    </style>
  </head>
```

# Manipulation du CSS

## Exemple --- suite ---

```
<body>
  <h1> Exemple addClass(), removeClass() et toggleClass()</h1>

  <div id='Div01' class='default'>
    <p>Premier paragraphe de Div01</p>
    <p>Deuxième paragraphe de Div01</p>
  </div>
  <p></p>
  <div id='Div02' class='default'>
    <p>Premier paragraphe de Div02</p>
    <p>Deuxième paragraphe de Div02</p>
  </div>

  <button id='b1'>Ajouter des attributs class au div01</button>
  <button id='b2'>Supprimer les attributs class du div01</button>
  <button id='b3'>Modifier les attributs class du div02</button>
```



# Manipulation du CSS

## Exemple --- suite ---

```
<!--On inclut la librairie jQuery-->
<script src='https://code.jquery.com/jquery-3.3.1.js'></script>

<!--On écrit notre code JavaScript / jQuery-->
<script>
$(document).ready(function(){
    $('#b1').click(function(){ $('#Div01').addClass('style01 centre'); });
    $('#b2').click(function(){ $('#Div01').removeClass('defaut style01 centre'); });
    $('#b3').click(function(){ $('#Div02').toggleClass('defaut style02 droite'); });
});
</script>
</body>
</html>
```

# Manipulation du CSS

## Explication de l'Exemple

Dans cet exemple, on va créer 4 class styles CSS (**default**, **centre**, **droite**, **styleo1** et **styleo2**), deux **Blocs** (DIV1 et div2 avec une bordure noire, solide d'une largeur de 1px avec la class style **default**) Et 3 boutons b1, b2 et b3 permettant de gérer les classes des style de DIV1 et div2, et ce, comme suite :

- **Bouton b1** : pour ajouter les attributs **class** 'styleo1 et centre' au DIV1 en utilisant la méthode **addClass**.
- **Bouton b2** : Pour supprimer les attributs **class** 'default, styleo1 et centre' de DIV1 en utilisant la méthode **removeClass**.
- **Bouton b3** : Modifier les attributs **class** 'default styleo2 et droite' du DIVo2 en utilisant la méthode **toggleClass** ce que supprime les attributs existants et ajoute ceux qui n'existe pas.

## Exemple02 CSS()

```
<!DOCTYPE html>
<html>
  <head>
    <title>Atelier 03 de jQuery</title>
    <meta charset='utf-8'>
    <style>
      p{ font-size: 16px; }
    </style>
  </head>
  <body>
    <h1>Manipulation du styles CSS </h1>
    <div id='Div01' class='defaut'>
      <p>Premier paragraphe</p>
      <p>Deuxième paragraphe</p>
    </div>
    <br><span id='msg'></span><br><br>
    <button id='b1'>Modification des attributs des paragraphes</button>
    <!--On inclut la librairie jQuery-->
    <script src='https://code.jquery.com/jquery-3.1.0.js'></script>
```

<!--On écrit le code JavaScript / jQuery-->

<script>

\$(document).ready(function(){

//utilisation de la methode css() comme getter

var taille = \$('p').css('font-size');

\$('#msg').text('Taille du texte : ' + taille);

// utilisation de la methode css() comme setter

\$('#b1').click(function(){

//Modification d'un seul attribut

\$('p').css('font-style','italic');

//Modification de deux attributs en même temps

\$('p').css({'color':'blue','font-size':'60px'});

});

});

</script>

</body>

</html>

# Manipulation du CSS

## Explication de l'Exemple

Dans cet exemple, on va créer un bloc de deux paragraphes de 16px de taille de police, puis on utilisera deux fois la méthode CSS() pour modifier (le style, la couleur et la taille de police). la première fois, modifier uniquement le style en italique (modification d'un seul attribut) et la deuxième fois, modifier la couleur et la taille de police (blue et 60px) pour distinguer la déférence entre la modification d'un seul attribut et la modification de plusieurs attributs.

# Gestion Des Événements

# Gestion Des Evénements

## Introduction

Un évènement correspond à un moment précis où une action se produit.

L'action peut venir :

- de l'utilisateur (exemple : clic sur un élément de la page).
- du navigateur ( exemple : lors du chargement d'une page).

# Gestion Des Événements

## gestion avancée des événements en jQuery

Pour gérer les événements, jQuery dispose de la méthode **on()** ainsi qu'un certain nombre de ses notations raccourcies.

Méthode	Événement géré
on()	Initialise un gestionnaire d'événements
off()	Supprimer un gestionnaire d'événements.
trigger()	Simule un événement

Avantage de la méthode on() :

- Ecouter plusieurs événements en même temps
- Exécuter des fonctions différentes pour chaque événement



# Gestion Des Evénements

## gestion avancée des évènements en jQuery

Avantage:

Ecouter plusieurs évènements en même temps

Exécuter des fonctions différentes pour chaque évènement

Syntaxe (une seule événement):

```
$(sélécteur).on('évènement',fonction({}));
```

Syntaxe (plusieurs événements) :

```
$(sélécteur).on({  
    évènement1 :fonction({}),  
    évènement2 :fonction({}),  
    ...  
});
```

# Gestion Des Événements

liste des événements peuvent être passés à la méthode on

événements	Se déclenchée
<code>.on('click', function() { ... })</code>	au clic
<code>.on('scroll', function() { ... })</code>	utiliser la roulette
<code>.on('hover', function() { ... })</code>	au Passage sur l'élément ciblé
<code>.on('mouseover', function() { ... })</code>	Au Début de survol de l'élément
<code>.on('mouseenter', function() { ... })</code>	à l'entrée du curseur sur l'élément ciblé.
<code>.on('mouseleave', function() { ... })</code>	à la sortie du curseur de l'élément ciblé.
<code>.on('keydown', function() { ... })</code>	Appui sur une touche du clavier
<code>.on('keyup', function() { ... })</code>	Relâchement d'une touche du clavier préalablement enfoncée
<code>.on('keypress', function() { ... })</code>	Maintien d'une touche du clavier enfoncée
<code>.on('focus', function() { ... })</code>	Réception de focus par l'élément
<code>.on('blur', function() { ... })</code>	Perte de focus par l'élément
<code>.on('resize', function() { ... })</code>	Redimensionnement d'un élément

# Gestion Des Événements

## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
    <style>
      div{ border: 2px solid black; }
    </style>
  </head>
  <body>
    <h1>Gestion d'évènements en jQuery</h1>
    <div class='div1'>
      <p>Les 10 plus belles fleurs rouges du jardin</p>
      <p>La rose rouge</p>
    </div>
    <span id='msg'></span>
    
    <!--On inclut la librairie jQuery-->
    <script src='https://code.jquery.com/jquery-3.3.1.js'></script>
```

# Gestion Des Evénements

## Exemple ----Suite

```
<script> <!--écriture du code JavaScript / jQuery-->
l=$('#img').attr('width') ;          h=$('#img').attr('height') ;
$(document).ready(function(){
    $('#div1').on({
        mouseenter: function(){
            $('#img').attr('width',l*5);
            $('#img').attr('height',h*5);
            $('#msg').text('Vous êtes entré dans le div');
            $(this).css("background-color", "yellow");          },
        mouseleave: function(){
            $('#msg').text('Vous êtes sorti du div');
            $(this).css("background-color", "pink");
            $('#img').attr('width',l);
            $('#img').attr('height',h);          }
    });
});
</script>
</body> </html>
```

# Gestion Des Evénements

## Explication de l'Exemple

Dans cet exemple, on a utilisé la méthode **on()** pour gérer deux événement **mouseenter** et **mouseleave** sur **DIVo1**

- lorsque la souris entre dans **DIVo1**, la **DIV** prend la couleur "jaune" et l'image de la fleur sera agrandie 5 fois. (**mouseenter**)

- lorsque la souris sort du **DIVo1**, la **DIV** prend la couleur "rose" et l'image de la fleur prend sa taille initiale. (**mouseleave**)

# Gestion Des Evénements

## Les méthodes raccourci du on()

# Gestion Des Evénements

La méthode jQuery **on()** permet de gérer tous les évènements, possède un nombre des méthodes **notations raccourcies**.

Ces **notations raccourcies** portant le nom des évènements auxquels on souhaite réagir (Exemple la méthode `click()` qui va nous permettre de gérer l'évènement `click`).

A noter que depuis la version 1.7 de jQuery, la documentation officielle recommande d'utiliser `on()` plutôt que les notations raccourcies.

ces méthodes sont :

# Gestion Des Evénements

## gestion des Evénements de la souris

Méthodes	Action
click()	Se déclenche au clic.
dblclick()	Se déclenche au double-clic
hover()	Se déclenche au Passage sur l'élément ciblé
mouseenter()	Se déclenche à l'entrée du curseur sur l'élément ciblé.
mouseleave()	Se déclenche à la sortie du curseur de l'élément ciblé.
mousedown()	Se déclenche à l'enfoncement du bouton de la souris.
mouseup()	Se déclenche au relâchement du bouton de la souris
mousemove()	Se déclenche au mouvement de la souris.
scroll()	Scroller (utiliser la roulette)
toggle()	Se déclenche à chaque clic, et alterne deux fonctions.



# Gestion Des Événements

## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
  </head>
  <body>
    <h1>Gestion d'évènements en jQuery</h1>
    <p>Les 10 plus belles fleurs rouges du jardin</p>
    <p>La rose rouge</p>
    

    <!--On inclut la librairie jQuery-->
    <script src='https://code.jquery.com/jquery-3.3.1.js'></script>

    <!--On écrit notre code JavaScript / jQuery-->
    <script>
      l=$('#img').attr('width') ;      h=$('#img').attr('height') ;
      var i=0;
```

# Gestion Des Événements

## Exemple

```
$(document).ready(function(){
    $('img').dblclick(function(){
        if(i==0)
        {
            $('img').attr('width',l*5);
            $('img').attr('height',h*5);
            i=1;
        }
        else
        {
            $('img').attr('width',l);
            $('img').attr('height',h);
            i=0;
        }
    });
});
</script>
</body>
</html>
```

# Gestion Des Evénements

## Explication de l'Exemple

Dans cet exemple, on a utilisé la méthode **dblclick ()** pour gérer deux événement double clic de la souris, lorsqu'on clique deux fois sur la fleur

- Si la fleur est petite, elle sera agrandie 5 fois.
- Si la fleur est grande prend sa taille initiale.

# Gestion Des Événements

## Exemple avec HOVER()

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<p id="p1">This is a paragraph.</p>
<script src='https://code.jquery.com/jquery-3.3.1.js'></script>
<script>
$(document).ready(function(){
    $("#p1").hover(function(){
        $(this).css("background-color", "yellow");
    },
    function(){
        $(this).css("background-color", "pink");
    });
});
</script>
</body>
</html>
```

# Gestion Des Événements

## gestion des Événements du Clavier

Méthode	Événement géré
keydown()	Appui sur une touche du clavier
keyup()	Relâchement d'une touche du clavier préalablement enfoncée
keypress()	Maintien d'une touche du clavier enfoncée
la propriété <b><u>keyCode</u></b> permet de récupérer le code de la touche enfoncée	

### Exemple avec keydown() et 'keyup()' ---suite ---

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
  </head>
  <body>
    <h1>Gestion d'évènements en jQuery</h1>

    <label for='prenom'>Entrez votre prénom</label>
    <input type='text' id='prenom' name='prenom'>

    <!--On inclut la librairie jQuery-->
    <script src='https://code.jquery.com/jquery-3.1.0.js'></script>
```

### Exemple avec keydown() et 'keyup()' ---suite ---

```
<!--On écrit notre code JavaScript / jQuery-->
<script>
  $(document).ready(function(){
    //Si une touche est enfoncée, le champ est bleu
    $('#prenom').on('keydown',function(){
      $(this).css('background-color', '#69C');
    });
    //Si on relâche la touche, le champ devient orange
    $('#prenom').on('keyup',function(){
      $(this).css('background-color', '#C72');
    })
  });
</script>
</body>
</html>
```

# Gestion Des Événements

## gestion des Événements du Formulaire

Méthode	Événement géré
focus()	Réception de focus par l'élément
blur()	Perte de focus par l'élément
focusin()	Réception de focus par l'élément ou un de ses enfants
focusout()	Perte de focus par l'élément ou un de ses enfants
resize()	Redimensionnement d'un élément
change()	Modification d'un élément





# Effets et Animations

## sélecteurs et modifieurs

ATELIER 04

# Effets et Animations

## modification du style des pages

### css()

#### Syntaxe :

//pour récupérer la valeur d'une propriété

– \$(selecteur).css('propriété')

// Pour affecter une valeur à une propriété

– \$(selecteur).css('propriété', 'valeur')

// affecte ses valeurs à plusieurs propriétés

– \$(selecteur).css({'pro1':'val1', 'prop2', 'val2', ...})

# Effets et Animations

## Apparition /disparition

### Syntaxe:

**`$(sélecteur).methode(vitesse,callback);`**

Méthodes	Description
<b><code>show() , hide() toggle()</code></b>	apparaître / disparaître les éléments.
<b><code>fadeIn(), fadeOut() fadeToggle</code></b>	apparaître/ disparaître les éléments avec une animation de fondu

- **Vitesse** : "slow", "normal", ou "fast" ou le nombre en millisecondes (600ms, 400ms ou 200ms)
- **Callback** : (optionnel) fonction à exécuter à la fin de l'animation.

## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
    <style>
      div{
        border: 2px solid black;}
    </style>
  </head>
  <body>
    <h1>Gestion d'évènements en jQuery</h1>
    <div class='div1'>
      <p>Les 10 plus belles fleurs rouges du jardin</p>
      <p>La rose rouge</p>
    </div>
    <span id='P01'></span>
    
```

## Exemple

```
<button id='b1'>Cacher l'image avec fadeOut()</button>
  <button id='b2'>Afficher l'image avec fadeIn()</button>
  <button id='b3'>Afficher/ Cacher l'image avec fadetoggle()</button>
  <!--On inclut la librairie jQuery-->
  <script src='https://code.jquery.com/jquery-3.3.1.js'></script>
  <!--écriture du code JavaScript / jQuery-->
  <script>
    $(document).ready(function(){
      $('#b1').click(function(){      $('img').fadeOut('fast');
    });
      $('#b2').click(function(){      $('img').fadeIn('fast');
    });
      $('#b3').click(function(){      $('img').fadeToggle('fast');
    });
    });
  </script>
</body>
</html>
```

# Gestion Des Evénements

## Explication de l'Exemple

Dans cet exemple, on a utilisé les méthode

- **fadeOut()** pour cacher la fleur.
- **fadeIn()** pour afficher la fleur si elle est cachée.
- **fadetoggle()** pour cacher la fleur si elle est affichée et l'affiché si elle est cachée.

# Effets et Animations

## Syntaxe:

`$(sélecteur').fadeTo(durée, opacité, callback);`

`$(sélecteur). methode(vitesse, callback);`

Méthodes	Description
<b>fadeTo</b>	modifier progressivement l'opacité d'un élément sans aller jusqu'à sa disparition ou sa complète opacité
<b>slideDown()</b>	augmente la hauteur et la largeur de la sélection jusqu'à atteindre la dimension « native » de chaque élément ;
<b>slideUp()</b>	diminue la hauteur et la largeur de la sélection jusqu'à ce qu'elle disparaisse ;
<b>slideToggle()</b>	inverse l'animation



## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
    <style>
      div{
        border: 2px solid black;}
    </style>
  </head>
  <body>
    <h1>Gestion d'évènements en jQuery</h1>
    <div class='div1'>
      <p>Les 10 plus belles fleurs rouges du jardin</p>
      <p>La rose rouge</p>
    </div>
    <span id='P01'></span>
    
```

## Exemple

```
<button id='b1'>Cacher l'image avec slideUp()</button>
<button id='b2'>Afficher l'image avec slideDown()</button>
<button id='b3'>Afficher/ Cacher l'image avec slideToggle()</button>
<button id='b4'>Afficher/ Cacher l'image avec fadeTo()</button>
<!--On inclut la librairie jQuery-->
<script src='https://code.jquery.com/jquery-3.3.1.js'></script>
  <!--écritre du code JavaScript / jQuery-->
<script>
  $(document).ready(function(){
    $('#b1').click(function(){      $('#img').slideUp('2000');      });
    $('#b2').click(function(){      $('#img').slideDown('2000');      });
    $('#b3').click(function(){      $('#img').slideToggle('2000');      });
    $('#b4').click(function(){      $('#img').fadeTo(3000,0.25);      });
  });
</script>
</body>
</html>
```

## Animation personnalisée : `animate()`

### Syntaxes

```
$('selecteur').animate({ prop1: val1, prop2: val2, etc. },  
durée, modèle, function() { // instructions });
```

modèle : modèle de progression de l'animation, (swing, linear ou un modèle issu d'un plugin).

## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
    <style>
      div{
        border: 2px solid black;}
    </style>
  </head>
  <body>
    <h1>Gestion d'évènements en jQuery</h1>

    <div class='div1'>
      <p>Les 10 plus belles fleurs rouges du jardin</p>
      <p>La rose rouge</p>
    </div>
    <span id='P01'></span>
    
```

## Exemple

```
<button id='b1'>Animation avec animate()</button>
  <button id='b2'>Animation avec fil d'attente et animate()</button>
  <!--On inclut la librairie jQuery-->
  <script src='https://code.jquery.com/jquery-3.3.1.js'></script>
  <!--écritre du code JavaScript / jQuery-->
  <script>
$(document).ready(function(){
    $('#b1').click(function(){
        $('#img').animate({
            width:'-=50px',
            height:'20px'},5000);          });
    $('#b2').click(function(){
        $('#img') .animate({width:'-=50px'},5000)
        .animate({height: '20px'},2000);    });
    });
  </script>
</body>
</html>
```

## Animation personnalisée : animate()

**.stop()** : Arrête l'animation

`$(selecteur).stop(stopAll, goToEnd);`

- **stopAll** spécifie si également la file d'attente d'animation doit être effacé ou non. Par défaut est faux.

- **goToEnd** paramètre spécifie si oui ou non pour terminer l'animation en cours immédiatement. Par défaut est faux.

**.delay(Durée)** : définir un temp savant lequel certaines animations ne doivent pas être lancées.

## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
    <style>
      div{
        border: 2px solid black;}
    </style>
  </head>
  <body>
    <h1>Gestion d'évènements en jQuery</h1>

    <div class='div1'>
      <p>Les 10 plus belles fleurs rouges du jardin</p>
      <p>La rose rouge</p>
    </div>
    <span id='P01'></span>
```

## Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Découverte de jQuery</title>
    <meta charset='utf-8'>
    <style>
      div{
        border: 2px solid black;}
    </style>
  </head>
  <body>
    <h1>Gestion d'évènements en jQuery</h1>

    <div class='div1'>
      <p>Les 10 plus belles fleurs rouges du jardin</p>
      <p>La rose rouge</p>
    </div>
    <span id='P01'></span>

    <button id='b1'>Animation avec animate()</button>
```



## Exemple

```
<button id='b1'>Animation avec animate()/</button>
<button id='b2'>Animation avec fil d'attente et animate()/</button>
<button id='b3'>Animation avec fil d'attente et animate() et delay</button>
<button id='b4'>Arrêter l'animation</button>
<div>
    
</div>
<!--On inclut la librairie jQuery-->
<script src='https://code.jquery.com/jquery-3.3.1.js'></script>

<!--écritre du code JavaScript / jQuery-->
<script>
    $(document).ready(function(){
        $('#b1').click(function(){
            $('img').animate({
                width:'-=200px',
                height:'100px'},5000);
        });
    });
</script>
```

## Exemple

```
$('#b2').click(function(){  
    $('#img')  
        .animate({width:'-=200px'},5000)  
        .animate({height: '100px'},2000);  
});
```

```
$('#b3').click(function(){  
    $('#img')  
        .animate({width:'-=200px'},5000)  
        .delay(2000)  
        .animate({height: '100px'},2000);  
});
```

```
$('#b4').click(function(){  
    $('#img').stop(true);  
});
```

```
    </script>  
</body>  
</html>
```

Fin d'Atelier

Fin du chapitre