

LA MAITRISE DU DOM

JQUERY, LES FONDAMENTAUX

LA MAITRISE DU DOM

#Installer jQuery ou utiliser le CDN	P.3
#Attendre le chargement du DOM	P.4
#Les sélecteurs en jQuery	P.5
#Les principaux événements en jQuery	P.6
#Modifier ou afficher le contenu d'une balise	P.7
#Activer ou désactiver une fonction	P.8
#Vérifier les champs d'un formulaire	P.9

INSTALLER JQUERY OU UTILISER LE CDN

jQuery est un ensemble de fonctions JavaScript simplifiées qui permettent une manipulation plus fluide du DOM. Tout comme un fichier de script normal, jQuery doit être installé pour pouvoir fonctionner et il faut le faire avant le script de la page Web car il fera appel à des fonctions jQuery.

jQuery.min
Afin d'alléger le contenu de la page il est recommandé d'utiliser le CDN minimiser

```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="utf-8" />
    <title>Bonjour le monde</title>
    <link rel="stylesheet" type="text/css" href="css/style.css">
  </head>

  <body>
    <script type="text/javascript"
      src="https://code.jquery.com/jquery-1.11.3.min.js">
    </script>
    <script type="text/javascript" src="js/script.js"></script>
  </body>

</html>
```

L'utilisation du CDN nécessite une connexion à Internet pour pouvoir être chargé, pour fonctionner en local il faut télécharger jquery est l'importé dans une page HTML comme un script Javascript normal.

ATTENDRE LE CHARGEMENT DU DOM

JavaScript est un langage principalement prévu pour ajouter de l'interactivité aux pages Web, c'est pourquoi il est important d'attendre le chargement du DOM avant d'exécuter les commandes JavaScript. En jQuery il faut sélectionner le document pour lui appliquer la fonction `.ready()`.

Fonction `ready()`

A l'appel de cette fonction, nous plaçons en paramètre une fonction globale qui englobe tous le code JavaScript à exécuter sur la page



```
//Commande à exécuter dès la première ligne du .js
$(document).ready(function() {
```

```
    //JavaScript exécuté un fois le DOM chargé
    alert('Le DOM est chargé');
```

```
});
```

Sans attendre

Le JavaScript hors de la fonction globale sera exécuté en premier, sans attendre le chargement du DOM



```
//Le JavaScript ci-dessous n'attend pas le chargement du DOM
alert('Le DOM n'est pas chargé');
```

Comme pour un fichier CSS, il est recommandé de lancer une commande pour s'assurer du bon chargement de la bibliothèque jQuery.

LES SELECTEURS EN JQUERY

jQuery permet une manipulation simplifiée du DOM, la sélection des balises HTML via les fonctions jQuery se fait sans avoir à préciser `document.getElementById` comme en JavaScript classique mais via la commande `$(".target")`. L'exemple ci-dessous illustre l'utilisation de cette commande.

Sélecteur d'éléments `$("...")`

Les éléments à sélectionner sont indiqués entre les guillemets : balises html, class ou id

```
//Sélecteurs jQuery
var allImg = $("img");
var allClass = $(".wailer");
var myId = $("#logo");
```

Trouver des éléments

Les sélecteurs propres à jQuery permettent une sélection encore plus précise des éléments du DOM

Le sélecteur **jQuery :first** correspond au sélecteur **CSS :first-of-type**

Les fonctions **.find()**, **.children()**, **.parent()**, **.prev()** et **.next()** peuvent prendre en argument d'autres balises

```
//Sélecteurs propres jQuery
var firstImgOfBob = $("#bob img:first");
var findBunnyImg = $("#bunny").find("img");
var childOfPeter = $("#peter").children();
var parentOfPeter = $("#peter").parent();
var beforeBunny = $("#bunny").prev();
var afterBunny = $("#bunny").next();
```

Sélecteur d'élément par index

La fonction **.eq()** permet de sélectionner la balise selon son ordre d'apparition dans le DOM **en commençant par zéro**

```
//Sélecteur eq()
var firstSpan = $("span").eq(0);
var secondeSpan = $("span").eq(1);
```

LES PRINCIPAUX EVENEMENTS EN JQUERY

Les événements sont utilisés pour déclencher des fonctions à différents moments d'interactions avec l'utilisateur. Le plus couramment utilisé en jQuery est .click() qui permet de déclencher une fonction lorsque l'utilisateur click sur une balise.

Événement focus()

<https://api.jquery.com/focus/>



```
$( 'input' ).focus( function() {  
    console.log( 'Le curseur est dans l\'input' );  
});
```

Événement blur()

<https://api.jquery.com/blur/>



```
$( 'input' ).blur( function() {  
    console.log( 'Le curseur est sorti de l\'input' );  
});
```

Événement hover()

<https://api.jquery.com/hover/>



```
$( 'input' ).hover( function() {  
    console.log( 'Le curseur est au-dessus de l\'input' );  
});
```

Événement scroll()

<https://api.jquery.com/scroll/>



```
$( window ).scroll( function() {  
    console.log( 'L\'utilisateur fait défiler la page ' );  
});
```

Événement load()

<https://api.jquery.com/load-event/>



```
$( 'img' ).load( function() {  
    console.log( 'Les images sont chargées' );  
});
```

MODIFIER OU AFFICHER LE CONTENU D'UNE BALISE

Une fois sélectionner en jQuery, il est possible d'éditer le contenu d'une balise HTML avec les fonctions `text()` et `html()`. La première permet d'afficher ou d'éditer le contenu d'une balise au format texte et la deuxième au format HTML.

Fonction `text()`

Pour éditer le contenu d'une balise au format texte



```
//Modifier le contenu avec text() et html()  
$('h1').text('Un nouveau titre');
```

Fonction `html()`

Pour éditer le contenu d'une balise au format HTML



```
$('p').html('Un texte <strong>gras</strong>');
```

```
//Afficher le contenu avec text()  
var showText = $('p').text();  
console.log(showText); //Les balises internes seront ignorées  
  
//Afficher le contenu avec html()  
var showTitle = $('h1').html();  
console.log(showText); //Les balises internes seront affichées
```

Les fonctions `text()` et `html()` permettent également d'afficher le contenu d'une balise parent (<section> par exemple) et de la modifier, il est préférable d'utiliser la fonction `html()` pour éditer le contenu d'une balise parent

ACTIVER OU DESACTIVER UNE FONCTION

Les fonction on() et off() de jQuery permettent de gérer l'activation d'une fonction. Il faut déclarer une fonction et l'intégrer en paramètres des fonctions on() et off() et la fonction déclarer sera active à l'instant voulu. Prenons pour exemple un formulaire qui n'est validable que si le champ firstName est renseigné.

Fonction preventDefault()

Fonction jQuery permettant d'annuler le fonctionnement normal du formulaire

```
//Fonction pour afficher #validation
var formSubmit = function(event){
    event.preventDefault();
    console.log($('#firstName').val());
};
```

Fonction on()

Les paramètre de la fonction active le click sur #validation

```
//Condition d'activation
if ($('#firstName').val() == 0) {
    on("click", "#validation", formSubmit);
};
```

Fonction off()

Les paramètre de la fonction désactive le click sur #validation

```
} else {
    off("click", "#validation", formSubmit);
};
```

Les fonctions on() et off() peuvent être utilisées sur tous types d'évènements.

VERIFIER LES CHAMPS D'UN FORMULAIRE

L'utilisation des formulaires sur le Web est très répandue et avec JavaScript et la bibliothèque jQuery il est devenu possible d'ajouter des systèmes de vérification complets qui aident l'internaute à se repérer dans les étapes du formulaire.

Récupération du contenu des inputs

La fonction val() permet de récupérer la valeur des inputs renseignés par l'utilisateur

Connaître la longueur d'un input

L'utilisation de la fonction length() sur la valeur du input permet de savoir combien de lettres ont été tapées par l'utilisateur

Guider l'utilisateur

La fonction html() sur une balise placé à côté des inputs permettent d'indiquer à l'utilisateur la marche à suivre

```
$("#validation").click(function(event){
    event.preventDefault();
    var firstName = $("#firstName").val();
    var lastName = $("#lastName").val();

    if (firstName.length >= 5 && lastName.length >= 5) {
        console.log('Le prénom est : ' + firstName);
        console.log('Le nom est : ' + lastName);

    } else {
        $("#helpForm").html("Vous devez remplir <b>tous les champs</b>");
    };
});
```

L'association de JavaScript à PHP pour la gestion des formulaires permet de ne pas forcer l'utilisateur à recharger la page pour sauvegarder les valeurs du formulaire.