



# Langages JavaScript et jQuery

Dr. Dellys Hahchemi

Cours Technologie et Développement Web (TDW) 2CS

2019-2020



2

# Introduction

# Pourquoi JavaScript ?

- HTML est un langage statique
- Les capacités dynamiques de CSS sont très limitées
- Ne répondent pas à certains besoins : répondre aux interactions, traitement, boucles,...
- JavaScript a été proposé dans ce contexte
- JS est un langage interprété par le navigateur
- JS a été créé par Netscape Communications

## Que peut faire JS ?

- Afficher des messages
- Valider les formulaires
- Faire des calculs
- Faire des animations
- Détecter le navigateur et son potentiel
- Interagir avec les serveur à travers les services web et AJAX

# Déclarations JavaScript

```
<!-- script interne -->  
  <script type="text/javascript">  
    function appelScriptInterne() {  
      alert('salam interne');  
    }  
  </script>  
<!-- script externe -->  
<script src="ScriptIntro.js"></script>
```

6

# Syntaxe

# Instructions

- La syntaxe JavaScript ressemble à la famille C, C++, Java ou C# avec des spécificités
- Les variables sont déclarées avec le mot clé **var**
- Le type est automatiquement déduit par l'interpréteur
- Quand c'est possible, le moteur JS effectue automatiquement les conversions
- Pour convertir une variable, utiliser `parseInt` et `parseFloat`

# Exemple

```
<script type="text/javascript">
    function somme(x, y) {
        return x + y;
    }
    /* cette fonction sert à faire l'addition de deux
    nombres */
    function calculer() {
        var nbr1 = parseInt(document.getElementById('txt1').value); // récupérer la valeur et la
convertir en int
        var nbr2 = parseInt(document.getElementById('txt2').value);
        var resultat = somme(nbr1, nbr2);
        alert(resultat);
    } </script>
```



# Fenêtre et DOM

# Introduction

- JS interagit avec la page à travers une structure appelée « DOM » accessible via une variable appelée « document »
- À travers le DOM, on peut accéder à un ou plusieurs éléments. Pour accéder à un élément par son « id », utiliser « `getElementById` »
- Si un élément HTML a un « id », alors, une variable est automatiquement créée pour lui

# Propriétés / Méthodes des nœuds

Fonction	Description
nodeName	Nom du nœud
nodeType	1 balise normale, 3 texte simple, 9 document
nodeValue	Valeur d'un nœud texte
innerHTML	HTML à l'intérieur du nœud
innerText	Texte brut à l'intérieur du nœud
id	Id de l'élément
className	Nom de la classe de l'élément
firstChild	Premier enfant
lastChild	Dernier enfant
style	Accède au style de l'enfant

## Propriétés / Méthodes des nœuds - suite

Fonction	Description
childNodes	Tableau contenant tous les enfants
previousSibling	Frère précédent
nextSibling	Frère suivant
appendChild(elt)	Ajoute un nouvel enfant au parent
insertBefore(elt, eltAvant)	Insère un élément avant un autre élément
replaceChild(nouveau, ancien)	Remplace un élément par un nouvel élément
removeChild(enfant)	Supprime l'enfant
hasChildren()	Indique si l'élément possède des enfants ou non
cloneNode()	Clone un élément
attributes	Tableau des attributs de l'élément

## Style d'un noeud

- Toutes les propriétés CSS sont accessibles via la propriété « **style** » d'un nœud
- Par exemple : `noeud.style.backgroundColor = 'yellow';`

# Propriétés / Méthodes de l'objet document

Fonction	Description
getElementById(id)	Renvoie un élément par son id
getElementsByTagName(balise)	Renvoie la liste des éléments du document dont la balise est « balise »
createTextNode(texte)	Crée une balise texte
createElement(balise)	Crée une balise HTML
getElementsByTagName(className)	Renvoie la liste des éléments par nom de classe

# Appel de Fonctions

- L'appel d'une fonction se fait en utilisant son nom et les paramètres entre parenthèses
- Si un paramètre n'est pas fourni, l'interpréteur lui donne la valeur « undefined »
- Utiliser le paramètre `||` pour affecter des valeurs par défaut

# Exemple

```
function calcul(liste, x, y) {  
    var elt = document.createElement('option');  
    x = x || 25; // si x n'est pas fourni, prendre 25 comme va par défaut  
    y = y || 30; // si y n'est pas fourni, prendre 30 comme val par défaut  
    elt.text = x + y;  
    elt.value = x + y;  
    liste.add(elt);  
}  
  
function testAppels() {  
    var liste = document.getElementById('liste');  
    calcul(liste, 11, 12); // fournit tous les paramètres  
    calcul(liste, 11); // 1er par par défaut  
    calcul(liste); // prend les deux paramètre par défaut  
    calcul(liste, 1, 2, 6); // dernier appel ignoré  
}
```



# Objets (bibliothèques) prédéfinis

- L'objet « Math » fournit des utilitaires mathématiques
- L'objet « String » gère les chaînes de caractères
- L'objet « Date » gère la date et l'heure

# Evènements et interactions

# Introduction aux évènements

- Les évènements représentent des fonctions appelées en réaction à une action de l'utilisateur : souris, clavier,....
- La fonction répondant à un évènement est appelée « gestionnaire d'évènement »
- Les évènements sont par convention préfixés par « on ». Par exemple : « onclick »

# Affectation de Gestionnaire - Exemple

```
<div class="vignette" onmouseover="survoler(this)"  
onmouseout="sortir(this)">Survolez-moi</div>
```

## L'objet « event » - Propriétés

Propriété	Description
modifiers	Touches spéciales (Par exemple Ctrl ou Alt)
button	Bouton gauche ou droit de la souris (0 ou 2)
pageX, pageY	Coordonnées
cancelable	Indique si l'évènement peut être annulé
target	Objet relatif à l'évènement
charCode	Relatif aux évènements clavier, code caractère
keyCode	Relatif aux évènements clavier, code touche
altKey	Indique si la touche « alt » est pressée
ctrlKey	Indique si la touche « ctrl » est touchée
preventDefault()	Annule l'évènement s'il est annulable

# Evènements de la souris

Evènement	Description
onclick	Clic simple de la souris
ondblclick	Clic double
onmousedown	Clic de la souris sur un élément
onmouseup	Quand l'utilisateur relache la souris
onmousemove	Quand la souris est déplacée sur un élément
onmouseover	Quand la souris entre dans l'espace d'un élément
onmouseout	Quand la souris sort de l'espace d'un élément

# Evènements des formulaires

Evènement	Description
onchange	Est provoqué au fur et à mesure que le contenu d'un contrôle est changé
oninput	Provoqué au fur et à mesure du changement du contenu dans une zone de texte
onfocus	Provoqué lorsqu'un contrôle reçoit le focus
onblur	Provoqué lorsqu'un contrôle perd le focus
onsubmit	Lorsque le formulaire est envoyé au serveur

# Evènements du clavier

Evènement	Description
onkeydown	Une touche est entrain d'être appuyée
onkeypress	Une touche est appuyée
onkeyup	Une touche est relachée



# Evènements de la fenêtre

Evènement	Description
onload	Appelé lorsque la page est chargée
onresize	Lorsque la fenêtre est redimensionnée
onerror	Lorsqu'une erreur se produit lors du chargement d'une image
onabort	Lorsque le chargement d'une image est annulé

# Interaction avec les fenêtres

Méthode	Description
open()	Ouvre une nouvelle fenêtre
close()	Ferme la fenêtre
moveTo()	Déplace la fenêtre
resizeTo()	Redimensionne la fenêtre
moveBy()	Déplace la fenêtre par rapport à la position en cours
resizeBy()	Redimensionne par rapport à la taille actuelle

# Boîtes de dialogue

Méthode	Description
<code>alert(msg)</code>	Affiche un message sous forme de dialogue
<code>confirm(msg)</code>	Message de confirmation. Renvoie true or false selon la réponse de l'utilisateur (oui / non)
<code>prompt(msg,val_defaut)</code>	Demande à l'utilisateur d'entrer une valeur



# Exercice

- Dans le site web de l'école, rajoutez une colonne au tableau des formations qui spécifie le taux de la taxe qui est de 10% pour les deux première formations, 0% pour la troisième formation et 20% pour les deux dernières.
- Le prix TTC ne doit plus être rajouter au code mais calculé à partir du prix HT et de la taxe. Programmez une fonction en JavaScript seulement pour calculé et affiché le prix TTC pour chaque formation.

## Section 8 : JQuery

# Introduction

- Bibliothèque JavaScript adoptée par la majorité des industriels : IBM, Netflix, Amazon, Dell, Twitter et Microsoft
- JQuery facilite la manipulation du DOM en appliquant des connaissances HTML, CSS ou JS
- JQuery permet de créer des scripts non obstrusifs
- Référence : <http://api.jquery.com/>
- Téléchargement dernière version : <http://jquery.com/>

# Avantage

- Economie de code (écrire moins, faire plus)
- Code non obstrusif (séparation de la structure du traitement);
- Compatibilité avec tous les navigateurs;
- Taille insignifiante du code;
- Manipulation HTML, CSS, évènements, animations, AJAX...etc.

# Ajout JQuery aux pages HTML

```
<head>  
<script type="text/javascript" src="jquery.js"></script>  
</head>
```

```
<head>  
<script type="text/javascript" src="jquery.min.js"></script>  
</head>
```





# Syntaxe

- La syntaxe de base de JQuery est sous la forme : **\$ ("selecteur"). action ()**.
  - un signe \$ pour définir JQuery
  - un (sélecteur) afin de trouver les éléments HTML
  - une action() JQuery a effectuer



## Syntaxe-exemple

- ➡ `$("a").hide()` : pour cacher les liens dans une pages HTML.
- ➡ `$("#ID").hide()` : pour cacher l'élément HTML dont l'identifiant est « ID ».
- ➡ `$(".MaClass").hide()` ; pour cacher tous les éléments appartenant à la classe « MaClass »
- ➡ `$(" this").hide()` : pour cacher l'élément HTML courant.



# Fonction Ready

- Permet d'éviter d'exécuter du code JQuery avant le chargement complet de la page (erreurs de sélection d'éléments non encore chargés)

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

```
// Toutes les fonctions jQuery devront se trouver ici
```

```
});
```



## Example

```
$(document).ready(function(){  
  $("button").click(function(){  
    $("p").hide();  
  });  
});
```

# Conflit d'Alias

- Plusieurs librairies JavaScript utilisent l'alias \$ pour leurs fonctions.
- La fonction noConflict() spécifie une variable pour remplacer le \$.

```
Var JQ=jQuery.noConflict();  
JQ(document).ready(function(){  
JQ("button").click(function(){  
JQ("p").hide();  
});  
});
```

# Sélecteurs JQuery

# Syntaxe des sélecteurs JQuery

- La sélection des éléments se fait selon la syntaxe du langage CSS.

Exemple:

- `$("p")` sélectionne tous les éléments p
- `$(" h1.titre")` sélectionne les éléments h1 appartenant à la classe titre
- `$(" span#ID")` sélectionne le span ayant pour identifiant « ID »

# Syntaxe des sélecteurs JQuery

- ➡ La sélection des attributs se fait selon la syntaxe XPATH.

Exemple:

- `$("[name='par']")` sélectionne tous les éléments possédant un attribut href ayant pour valeur « par »
  - `$("[src$='.jpg']")` sélectionne tous les éléments possédant un attribut src dont la valeur se termine par « .jpg ».
- 
- ➡ La sélection HTML se fait par la fonction `html()` et celle de CSS par la fonction `CSS()`.



# Sélecteurs JQuery

Sélecteur	Exemple	Description
*	<code>\$(" * ")</code>	Tous les éléments
Élément	<code>\$(" p ")</code>	Tous les éléments « p »
#ID	<code>\$(" #ID ")</code>	L'éléments avec l'identifiant « ID »
.Class	<code>\$(" .MaClass ")</code>	Les éléments appartenant à la classe « MaClass »
Div p	<code>\$(" div p ")</code>	Les « p » qui descendent de div
Div>p	<code>\$(" div-&gt;p ")</code>	Les « p » qui descendent directement de « div »
:eq(index)	<code>\$(" a:eq(2) ")</code>	Le deuxième élément « a »
:gt(index)	<code>\$(" p:gt(4) ")</code>	Les « p » dont la position est supérieure strictement à 4
:lt(index)	<code>\$(" li:lt(4) ")</code>	Les « li » dont la position est inférieure strictement à 4
:even	<code>\$(" div:even ")</code>	Les « div » en position paire
:odd	<code>\$(" li:odd ")</code>	Les « li » en position impaire
:not(s)	<code>\$(" not(p) ")</code>	Tous les éléments sauf les « p »

# Sélecteurs JQuery

sélecteur	Exemple	Description
:first	<code>\$("tr:first")</code>	Le premier élément « tr »
:last	<code>\$("td:last")</code>	Le dernier élément « td »
:animated	<code>\$("a:animated")</code>	Les éléments « a » animés
:empty	<code>\$("input:empty")</code>	Les « input » qui sont vides
:hidden	<code>\$("h2:hidden")</code>	Les « h2 » qui sont cachés
:visible	<code>\$("img:visible")</code>	Les « img » qui sont visibles
:contains(v)	<code>\$(":contains(.dz)")</code>	Tous les éléments contenant la chaîne « .dz »
:input	<code>\$(":input")</code>	Les éléments « input »
:text	<code>\$(":text")</code>	Les éléments « text »
:radio	<code>\$(":radio")</code>	Les éléments « radio »

# Sélecteurs JQuery

Sélecteur	Exemple	Description
:selected	\$("# :selected ")	Tous les éléments ayant l'attribut selected à vrai
:checked	\$("# :checked ")	Tous les éléments ayant l'attribut checked à vrai
:enabled	\$("# :enabled ")	Tous les éléments ayant l'attribut enabled à vrai
[attribute]	\$("# [src] ")	Les éléments ayant un attribut « src »
[attribute=val]	\$("# [src=alger.jpg] ")	Les éléments ayant un attribut « src » qui a une valeur « alger.jpg »
[attribute!=val]	\$("# [src!=alger.jpg] ")	Les éléments ayant un attribut « src » qui a une valeur différente de « alger.jpg »
[attribute\$=val]	\$("# [src\$=.jpg] ")	Les éléments ayant un attribut « src » qui a une valeur qui se termine par « .jpg »

## Exercice

Donner la signification des sélecteurs suivant:

- `$("div p:even, p:odd")`
- `$("img[src='fig1.jpg']+p:contains('définition')")`
- `$("ul.ing > li:last")`
- `$("a[href$='.dz']:hidden")`
- `$("img['src!=fig1.jpg'] a[target='_blank']")`

# Solution

- `$(" div p:even, p:odd ")` : tous les paragraphes d'ordre impair descendants d'un élément « div » et tous les paragraphes d'ordre pair
- `$(" img[src="fig1.jpg"]+p:contains("définition") ")`: Les paragraphes contenant le mot « definition » qui suivent directement les images dont le nom de fichier (source) est « fig1.jpg »
- `$(" ul.ing > li:last")` : le dernier élément descendant direct d'une liste « ul » qui appartient à la classe « ing »
- `$(" a[href$=".dz"]:hidden ")` : tous les liens cachés dont la valeur de l'attribut « href » se termine par « .dz »
- `$(" img["src!=fig1.jpg"] a[target="_blank"] ")`: sélecteur sémantiquement incorrecte, un lien ne peut descendre d'une image.



## Exercice

Écrivez en langage JQuery les sélecteurs qui permette d'avoir les éléments suivant:

- Tous les paragraphes de couleur rouge qui sont à l'intérieur de section de classe « important ».
- Toutes les images du type « GIF » et de taille de « 100px » qui sont à l'intérieur d'un div appartenant à la classe « imganim » qui à son tour est un descendant direct d'un article appartenant à la classe « typeimages ».
- Toutes les méta-données qui permettent d'indexer la page selon un mot clés.

# Solution

- Tous les paragraphes de couleur rouge qui sont à l'intérieur de section de classe « important ».

```
$(" section.important p[color='red']")
```

- Toutes les images du type « GIF » et de taille de « 100px » qui sont à l'intérieur d'un div appartenant à la classe « imganim » qui à son tour est un descendant direct d'un article appartenant à la classe « typeimages ».

```
$(" article.typeimages>div.imganim img[ src$='GIF' width='100px']")
```

- Toutes les méta-données qui permettent d'indexer la page selon un mot clés

```
$(" meta[ name='keywords']")
```

# Évènements et effets sous JQuery



# Introduction

- JQuery permet de simplifier l'affectation des évènements aux éléments et élimine tous les problèmes d'incompatibilité
- JQuery permet d'affecter un évènement de plusieurs façons et de le désaffecter
- JQuery peut même affecter des évènements à des éléments qui n'existent pas encore
- Pour plus de détail :

[http://w3schools.com/jquery/jquery\\_ref\\_events.asp](http://w3schools.com/jquery/jquery_ref_events.asp)



# Syntaxe

```
$(document).ready(function(){  
  $("button").click(function(){  
    $("p").hide();  
  });  
});
```

# Fonctions de gestion des événements

Fonction	Description
click,	Clic de la souris
hover	Survol de la souris
bind(nom, fonction)	Associe une fonction à un événement
unbind(nom)	Supprime un gestionnaire d'évènement
one(nom, fonction)	Gestionnaire d'évènement qui ne s'exécute qu'une seule fois
on(evt,selecteur,fonction)	Attache un évènements aux éléments sélectionnés (s'applique aussi aux éléments créés dynamiquement)
trigger(evt,donnees)	Déclenche manuellement un événement
off(evt)	Supprime le gestionnaire des éléments

# Effets jQuery

Fonction	Description
show()	Affiche les éléments sélectionnés
hide()	Cache les éléments sélectionnés
fadeTo(val)	Change l'opacité des éléments jusqu'à une certaine valeur
fadeIn()	Rendre les éléments plus visible
fadeOut()	Rendre les éléments plus invisible
animate()	Anime les éléments à l'aide de propriétés CSS
delay()	Définit un delai
stop()	Arrête une animation
toggle()	Donne un effet clignotant
slideUp()	Diminue la hauteur des éléments jusqu'à les rendre invisible
slideDown()	Augmente la hauteur des éléments

# Manipulation HTML

- La fonction `html()` permet de changer le contenu d'un élément qui est l'équivalent au `innerHTML` de JavaScript.
- La syntaxe est la suivante : **`$(selector).html(contenu)`**
- Plus d'information sur:  
[http://w3schools.com/jquery/jquery\\_ref\\_html.asp](http://w3schools.com/jquery/jquery_ref_html.asp)

Sélecteur	Description
<code>appendTo(elt)</code>	Ajoute des éléments à un ensemble
<code>insertAfter(elt)</code>	Insère un élément après un sélecteur
<code>insertBefore(elt)</code>	Insère un élément avant un sélecteur
<code>html(elt)</code>	Change le HTML interne d'un élément

# Manipulation CSS

- La fonction `CSS()` permet de manipuler les propriétés CSS à l'aide de jQuery
- `.css(name,value)` : fixe la valeur d'une propriété identifiée par name
- `.css({properties})` : pour en fixer plusieurs propriétés.

Exemple:

- `$("p").css("color", "black");`
- `$("div").css({"background-color":"beige","font-size":"150%"});`

# Fonction callback

- Une ligne de code peut être exécuté par le navigateur même si l'action liée à la ligne précédente n'est pas encore terminée. Cela peut créer des erreurs. Pour éviter cela, vous pouvez créer une fonction dite Callback.
- Une fonction Callback est une fonction qui est lancée après la fin de l'action en cours d'exécution.
- La syntaxe est la suivante : **`$(selector).action(valeur,callback).`**

## Exemple:

```
$("#div").hide(1000, function(){  
  alert(" les paragraphes sont cachés ");  
});
```

# AJAX sous JQuery



# Introduction à AJAX

- AJAX ou Asynchronous Javascript And XML permet d'interroger le serveur sans avoir à rafraîchir toute la page
- Le web 2,0 est techniquement basé sur AJAX
- Écrire du code AJAX directement est un peu fastidieux

## AJAX avec JQuery

- JQuery facilite l'intégration d'AJAX dans les applications web
- La fonction AJAX est une fonction utilitaire (\$.ajax)
- Les deux alias \$.get et \$.post permettent de faire des requêtes AJAX en mode GET ou POST

# Fonctions ajax

Fonction	Description
<code>\$(selecteur).load(url,data,callback)</code>	Requête de chargement partiel d'éléments distant à l'intérieur des éléments sélectionnés.
<code>\$.ajax(options)</code>	Fonction détaillée qui gère tous les paramètres AJAX
<code>\$.get(url,data,callback,type)</code>	Requête AJAX en mode GET
<code>\$.post(url, data,callback,type)</code>	Requête AJAX en mode POST
<code>\$.getJSON(url, data, callback)</code>	Requête de données en format JSON à parti du serveur

- (url) : l'URL des éléments à charger
- (data) : les données à envoyer au serveur sous la forme clé/valeur
- (callback) : fonction Callback à exécuter lorsque les données sont totalement chargées
- (type) : valeur de retour de la requête (html, xml, json, script,text)
- (options) : toutes les options (sous la forme clés/valeur)

# Exemple

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("button").click(function(){
par=$.ajax({url:« exemple.html",async:false});
$("p").html(par.responseText);
});
});
</script>
</head>
<body>
<p>Ce texte change quand on click sur le bouton</p>
<button>Changer le texte</button>
</body>
</html>
```



# Exercice 1

- Reprogrammez la fonctionnalité de calcul de la taxe avec la bibliothèque JQuery.
- 



## Exercice 2

- Mettez en place un mécanisme pour ajouter une nouvelle formation offerte par l'école dans le menu à l'aide de JQuery.
- 

# Bibliographie

- ▶ Java Script Reference, W3C Schools
- ▶ Sam's Teach Yourself, HTML, CSS, and JavaScript, 2012
- ▶ JQuery in Action, Bear Bibeault, Yehuda Katz, Mann, 2010
- ▶ JQuery reference ([www.JQuery.com](http://www.JQuery.com))