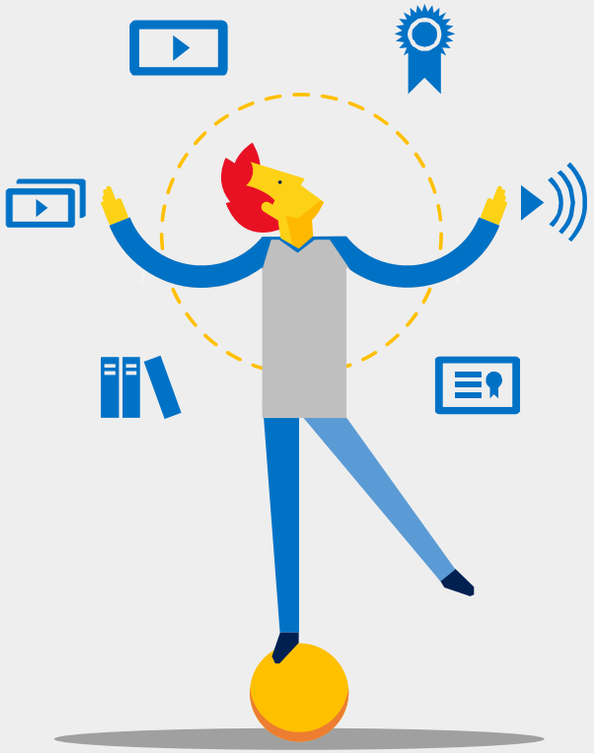
JavaScript, jQuery

Ajout de la bibliothèques externes Jquery pour améliorer les applications HTML



Microsoft

jQuery

Plan

- Inroduction
- Getter/setter
- JQuery et événements
- JQuery et Ajax

jQuery : qu'est ce que c'est ?

jQuery est une bibliothèque JavaScript rapide, petite et riche en fonctionnalités

- Bibliothèque Javascript libre et open-source
- Simplifie les interactions DOM/HTML/CSS/Javascript
- Requête à l'aide de sélecteurs CSS
- Événements des éléments de poignée
- Méthodes de chaînes
- Standardiser les requêtes API
- Développée par John Resig (auteur également du plugin CCK)

Ressources

- Téléchargeable sur le site officiel : https://jquery.com/

- Documentation et tutoriels :
  - https://learn.jquery.com/
  - https://api.jquery.com/
  - https://speckyboy.com/free-jquery-plugins/
  - https://fr.wikipedia.org/wiki/JQuery

- **Refcard** disponible sur le moodle **(**consultez la !**)**

Utilisation

- Téléchargez et décompressez le fichier .js :
  - Placez le fichier JQuery.js dans votre repertoire racine du projet
- Chargez la librairie puis commencez à coder !:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Le titre du document</title>
        <link rel="stylesheet" type="text/css">
    </head>
<body>
        <!-- code HTML pour structurer le contenu du document -->
        <script src="jquery.js"></script>
        <script src="autre_script.js"></script>
</body>
</html>
```

- Pour optimser le chargement, il est préférable de faire appel aux scripts à la fin du document

Sélection d'éléments

- JQuery est basée sur une seule et unique fonction JQuery() ou son alias $().:

Cette fonction peut prendre comme paramètres

- Une fonction, exécutée dès que le DOM est disponible

```
$(function() {
// Insérer le code jQuery ici
});
```

- Un sélecteur CSS ou une balise HTML

```
$('a'); // sélectionne tout les liens hypertextes
$('#id'); // sélectionne l'élement d'identifiant id
$('.important'); // sélectionne tous les élements de classe important
```

- Elle retourne un objet JQuery assimilable à un tableau contenant tous les éléments sélectionnés

Objet JQuery

Cet objet dispose de très nombreuses propriétés et méthodes (getters et setters)

- Pour accéder au tableau ainsi qu'à ses propriétés

```
$('a').length // Retourne le nombre de lien dans le document
$('a')[3] // Retourne, si il existe, le quatrieme lien du document
```

- Appliquer une méthode à la séléction

```
$('elem').action() // Applique action() à tous les éléments elem
$('a').css('color') // retourne la couleur des liens hypertextes
```

- Une liste exhaustive des méthodes est disponible sur https://api.jquery.com/

Sélecteurs : types

- Sélecteurs css/html

```
$('div') // tous les éléments correspondants aux balises div
$('.class') // tous les éléments de classe 'class'
```

- Sélecteurs par attributs

```
$('[nom]') // éléments ayant un attribut 'nom'
$('[nom:val]') // éléments ayant un attribut 'nom' de valeur val
```

- Sélecteurs hiérarchiques

```
$('parent > enfant') // éléments enfant descendants des éléments parents
$('enfant1 > enfant2') // éléments enfant2 précédés d'un élément enfant1
$('li:first-child') // premier élément descendant de l'élément li
```

Sélecteurs : types (suite)

- Pseudo sélecteurs

```
$('elem:even') // retourne les éléments elem d'indice pair
$('elem:gt(3)') // retourne les éléments elem d'indice superieur à 3
```

- Sélécteurs spécifiques

```
$('elem:hidden') // éléments elem qui sont cachés
$(':visible') // tous les éléments visibles
```

- Sélecteurs formulaires

```
$(':input') // tous les éléments de type input, textarea, select et button
$(':submit') // tous les éléments de type submit
$(':checkbox') // tous éléments de type checkbox
```

Parcourir la séléction

Le résultat d'une séléction est donné sous la forme d'un tableau qu'il est possible de parcourir à l'aide de la méthode each()

```
$('elem').each(function(index){
//Une ou plusieurs instructions JavaScript
});
```

```
<body>
        <h1>Introduction</h1>
        <h1>Corps du texte</h1>
        <h1>Conclusion</h1>
        <script src="jquery.js"></script>
        <script>
        $(function() {
                $('h1').each(function(index){
                this.text('Titre'+ (index+1));
                });
        });
        </script>
</body>
```

Accès et modification des éléments

- Accès/modification des propriétés CSS avec la méthode `css()`

```
var couleur = $('.titre').css('color');
$('.titre').css('color','red');
```

- Accès/modification des attributs avec la méthode `attr()`

```
$('#image').attr('src');
$('#image').attr('src','logo.gif');
$('#image').attr({src: 'logo.gif', alt: 'Logo de la société'});
$("a").attr({target:function(){…}});
```

- Accès/modification des champs de formulaires avec la méthode `val()`

```
$('champ').val();
$('#date').val('Today')
```

- Accès/modification du contenu des balises avec méthodes `text(),html()`

```
$('p').text();
$('p').text('ceci est le nouveau paragraphe');
```

Modification du DOM avec JQuery

JQuery permet également d'interagir avec le DOM, en particulier

- rajouter/supprimer des attributs, des classes…

```
addClass() //ajoute une classe dans les éléments sélectionnés ;
removeClass() //supprime une classe des éléments sélectionnés ;
toggleClass() //ajoute la classe si elle n'existe pas et la
supprime sinon
removeAttr() //supprime un attribut des balises le contenant
```

- rajouter/remplacer/supprimer des éléments du DOM

```
remove(); // supprime un élément du DOM
replaceWith(); // remplace l'élément sélectionné par un autre
append(); // insère du contenu à la fin de la sélection
prepend(); // insère du contenu au début de la sélection
before(); // insère du contenu avant la sélection
```

Effets

JQuery permet également d'interagir avec le DOM, en particulier

- Cacher ou rendre visibles des éléments

```
show(speed,callback) // Affiche progressivement un/des élements
hide(speed,callback) // Dissimule progressivement un/des élements
toggle(speed,callback) // Inverser une propriété
```

- Animer des éléments

```
animate(params,option) // Permet de définir une animation personnalisée
scrollTop(val)
```

Associer des données aux éléments

On peut stocker des données textuelles dans des éléments du DOM avec la méthode data()

```
$.data(el, 'nom', nom_don: don);
$.data('#id', 'Var_id', donnee1: valeur1);
var v = $.data('#id', 'Var_id').donnee1;
```

Méthode utile par exemple lors de l'édition d'un élément pour stocker dans celui-ci son état initial et le récupérer a posteriori:

```
var nextElmt = $("<textarea>" + contenuInitial +
"</textarea>").data("initValue",contenuInitial);
$(this).replaceWith(nextElmt);

$("textarea").each(function (){
// $(this) le textarea courant !
var initValue = $(this).data("initValue");
$(this).replaceWith('<p class="editable">' + initValue + "</p>");
});
```

JQuery : événements

- Méthode principale

```
$('elem').event(function() {
// gestion de l'événement lorsqu'il se produit
}
```

```
<script src="jquery.js"></script>
<script>
	$(function() {
		$('h1').mousemove(function() {
			alert('passage sur la balise <h1>');
		});
		$(window).scroll(function() {
			alert('Utilisation de la roulette dans le document');
			});
	});
</script>
```

- Permet de purifier le code HMTL !

JQuery : événements (suite)

- Evénements de la souris

```
click() // Clic gauche
dblclick() // Double-clic
mousedown() // Appui sur le bouton gauche ou droit de la souris
mouseenter() ou mouseover() // Début de survol de l'élément
mouseleave() ou mouseout() // Arrêt de survol de l'élément
mousemove() // Déplacement du pointeur au-dessus de l'élément
mouseup() // Relâchement du bouton gauche ou droit
scroll() //Utilisation de la roulette
```

- Evénements du clavier:

```
keydown() // Appui sur une touche du clavier
keyup() // Relâchement d'une touche du clavier préalablement enfoncée
keypress() // Maintien d'une touche du clavier enfoncée
```

JQuery : événements (suite)

- Evénement éléments

```
focus() // Réception de focus par l'élément
blur() // Perte de focus par l'élément
change() // Modification d'un élément
load(), unload() // Chargement complet de la page ou passage à une autre
```

- Les deux instructions suivantes sont équivalentes:

```
- $(elem).click(function() {…})
- $(elem).on('click', function() {…})
```

JQuery : événements (suite)

- La méthode on() permet la gestion de plusieurs événements

```
// méthodes standard
$('img').mouseenter(function() { … }
$('img').mousemove(function() { … }
// méthode on()
$('img').on('mouseenter mousemove', function() { … })
```

- Possibilité de création d'événements personnalisés:

```
$('element').on('monEvenement', function() {
        alert('Ceci est mon propre evenement !');
});
// declanchement de l'evenement
$('element').trigger('monEvenement');
```

Exercice

Scénario

It is almost time for your new web page to go live and you would like to modernize the web application. In testing, you found that some articles were very short and did not need to take up the entire width of the page. You would like to lay out the articles in a horizontal fashion. Also, your carousel works but you need to add more features to the carousel in the future and this may prove to be difficult. You have elected to leverage the third-party Bootstrap library to add these modern features to your web page.