

Développement Web: Introduction à JavaScript

Abdelraouf Hecham - IUT de Montpellier - AS

hecham@lirmm.fr (www.lirmm.fr/~hecham)

8 Janvier 2018

Outline

1 Le DOM

- Manipulation du DOM
- Window onLoad
- Ajouter des Éléments HTML avec Javascript

2 JQuery

DOM (Document Object Model)

- Le DOM (Document Object Model) est une représentation d'un document HTML (XML) sous forme d'arbre où chaque nœud est un élément (balise).
- Le DOM permet de manipuler les éléments de la page HTML (cacher/afficher une balise, changer son contenu, etc.)
- Chaque élément de la page HTML peut être identifié de manière unique par son attribut id:

```
<p id="monP">...</p>
```

```
var p = document.getElementById("monP"); // Retourne l'objet <p>  
p.innerHTML = "Ce texte a été changé par JS";
```

Exercices DOM

Exercice 1:

- Créez une page HTML qui contient une balise `<p>` qui contient "Ceci est un Texte" et un bouton "Changer Couleur".
- Au click sur le bouton, la couleur du texte du paragraphe devient rouge. (pour changer la couleur d'un élément du DOM: `element.style.color = "red"`).
- Ajouter un autre bouton "Changer Contenu" a la page HTML. Au click sur ce bouton le texte de la balise `<p>` change en "Ce texte a été changé par JS".

Exercice 2:

- Créez une page HTML qui contient deux balises `<p>` qui contiennent un texte quelconque. Ajouter un bouton "Compter les balises `<p>`".
- Au click sur le bouton, on affiche a la console le nombre d'element `<p>` dans la page HTML. (La fonction `document.getElementsByTagName(balise)` retourne un tableau de tout les éléments du type balise qui sont dans le document HTML).

Exercices DOM

Exercice 3:

- Créez une page HTML qui contient une balise `<input type="text">` et un bouton 'afficher'.
- Au clique sur le bouton, le contenu de input est affiché dans une boite d'alerte. (pour avoir la valeur d'un input, on utilise son attribut value: `var input = document.getElementById("monInput"); input.value`)

Exercice 4:

- Créez une page HTML qui contient une balise `<input type="text">` et un bouton 'Ajouter 2'.
- Au clique sur le bouton, si le contenu de input est un chiffre (`parseInt(valeur) != NaN`) lui ajoutez 2 et le mettre a jour dans le input. Sinon affichez dans une boite d'alerte le contenu de input suivi de " n'est pas un chiffre".

Il faut attendre que le DOM se crée!

Exercice 5:

- Créez une page HTML qui contient trois balises `<p>` qui contiennent un texte quelconque.
- Dans un document JS, afficher dans la console le nombre d'éléments `<p>` dans la page HTML.
- Quel est le chiffre affiché? pourquoi?

Solution:

- Avant de manipuler le DOM il faut s'assurer que tout a été bien créé, une manière de le faire est de mettre le code Javascript indépendant des événements (comme onclick, etc.) dans l'événement `window.onload`:

```
window.onload = function() {  
    // code Javascript  
}
```

Il faut attendre que le DOM se crée!

Exercice 6:

- Créez une page HTML qui contient une balise `<p>` avec un id "monP" et un texte quelconque.
- Dans un document JS, afficher directement (au chargement de la page) dans la console le contenu de `<p>`.
- Que ce passe-t-il si on ne fait pas le `window.onload`? pourquoi?

Ajouter des Éléments HTML avec Javascript

- Pour Créer un élément HTML on utilise la fonction `document.createElement(balise)` . e.g.
`var nouveauP = document.createElement("p");` // Crée un élément `<p>`
- pour l'ajouter a element parent on utilise `appendChild()` sur l'élément parent:
`var p = document.getElementById("monP");` // Retourne l'objet `<p>`
`p.appendChild(nouveauP);`

Exercice 7:

- Créez une page HTML avec une liste vide (`<ul id="maList">`).
- Dans Javascript, créez un tableau qui contient "Lait", "6 Oeufs", "1kg Oranges", "Un pack Coca".
- Ajoutez avec le Javascript les éléments du tableau dans la liste du document HTML.

Ajouter des Éléments HTML avec Javascript

Exercice 8:

- On veut convertir un nombre entre 0 et 999 en toutes lettres.
- Créez une page HTML avec un input, un paragraphe et un bouton "convertir".
- Au clique sur le bouton, le contenu de input est transformé en lettre et affichée dans le paragraphe. Par exemple, le nombre 33 est traduit en "trente trois", 0 en "zero", etc.
- Pour séparer les centaines, les dizaines et les unités, utiliser le modulo: $\text{nombre} \% 10$ vous donne les unités, $(\text{nombre} \% 100) - (\text{nombre} \% 10)$ vous donne les dizaines, etc.
- Utiliser un tableau pour faire le lien entre les chiffre et les lettres, par exemple:
unites = ['zero', 'un', 'deux', 'trois', 'quatre', 'cinq', 'six', 'sept', 'huit', 'neuf', 'dix', 'onze', 'douze', 'treize', 'quatorze', 'quinze', 'seize', 'dix-sept', 'dix-huit', 'dix-neuf'];
dizaines = ['', 'dix', 'vingt', 'trente', 'quarante', 'cinquante', 'soixante', 'soixante', 'quatre-vingt', 'quatre-vingt-dix'];
- Testez votre code avec: 0, 10, "98", 135, 999, 1000, 20.1, "text".

La bibliothèque JQuery

- Une **bibliothèque** est une collection de classes, objets, et fonction qui regroupe un ensemble de fonctionnalités.
- **jQuery** est une bibliothèque JavaScript qui a pour but de faciliter la gestion du DOM.
- Une fonction incontournable de cette bibliothèque est la fonction **\$**.
- Pour utiliser une bibliothèque dans Javascript il faut l'ajouter au document HTML:
Local: `<script src="jquery-3.2.1.min.js"></script>`

Ou en ligne:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```

La bibliothèque JQuery: Sélection d'élément

- Exécuter du code après le chargement du DOM: `window.onload` devient `$(document).ready(function(){...})`
- Sélectionner un élément avec son id: `document.getElementById("id")` devient `$("#id")`
- Sélectionner les éléments du même tag: e.g. `$("p")` sélectionne tout les `<p>`.
- Sélectionner les éléments auxquels on a attribué une classe donnée: e.g. `$(".maclass")` sélectionne tout les element qui ont la class CSS "maclass".

La bibliothèque JQuery: Gestion des événements

- JQuery attache une fonction a un événement on utilisant

`élément.on(événement, fonction)` : e.g.

```
$("#p").on("click", function() {  
    alert("Vous avez cliquez sur un paragraphe");  
});
```

Exercice 9:

- Créez une page HTML avec une liste vide (li).
- Dans Javascript, créez un tableau qui contient "Lait", "6 Oeufs", "1kg Oranges", "Un pack Coca".
- Ajoutez avec le Javascript les éléments du tableau dans la liste du document HTML en utilisant JQuery.

Effets avec JQuery

- Pour ajouter ou supprimer une classe CSS d'un élément on utilise `addClass(class)` et `removeClass(class)`. Exemple: ajouter une classe a tous les paragraphes
`$("p").addClass("maClassCSS")`
- `show()`, `hide()` et `toggle()` permettent de montrer et cacher des éléments: e.g.
`$("p").hide()`
- `slideDown()` et `slideUp()` permettent de faire apparaître (respectivement disparaître) un élément à la manière d'un store se déroulant ou s'enroulant.
- `fadeIn(vitesse)` et `fadeOut(vitesse)` permettent de faire progressivement apparaître (ou disparaître) un élément en jouant sur sa transparence ("slow", "normal" ou "fast").
- ... (voir documentation JQuery)

Manipulation du DOM

- Création d'un élément en suspens (sans parent): e.g. `$("<p id=\"monP\">...</p>")`
- Lecture du contenu HTML d'un élément: `$("#monP").html()`
- Modification du contenu HTML d'un élément: `$("#monP").html("blablabla")`
- Lecture du contenu valeur d'un élément (input): `$("#monInput").val()`
- Modification de la valeur d'un élément (input): `$("#monInput").val("blablabla")`
- Insertion d'un élément à l'intérieur d'un autre élément donné: **append(contenu)** et **prepend(contenu)** (à la fin ou début de l'élément sélectionné.)
- Insertion à l'extérieur d'un élément donné: **after(contenu)** et **before(contenu)** ajoutent contenu respectivement après et avant l'élément sélectionné.
- Insertion autour d'un élément donné: **wrap(élément)** e.g. `$("p").wrap($("<div></div>"));`
- Remplacement et suppression: **replaceWith(contenu)** permet de remplacer l'élément sélectionné par contenu. Pour vider un élément sélectionné de son contenu, on fait appel à la méthode **empty()**.

jQuery

- jQuery permet d'exécuter une fonction sur l'ensemble des éléments d'un Tableau.

En utilisant `$(this)` on a accès à chaque élément au fur et à mesure. Exemple: afficher le contenu de chaque paragraphe quand on clique dessus:

```
$("#p").on("click", function() {  
    alert("Vous avez cliqué sur le paragraphe qui contient: " + $(this).html());  
});
```

Exemple: Ajouter une classe au liens qu'on clique dessus: `$("#a").on("click", function() {`

```
    $(this).removeClass("maClassCSS");  
});
```

JQuery

Exercice 10:

- On veut convertir créer une calculatrice.
- Créez une page HTML avec deux input (operande1 et operande2), un paragraphe et quatres boutons (+, -, /, %).
- Au clique sur un des boutons, exécutez l'opération demandée avec les opérandes indiqués dans les inputs et affichez le résultat dans le paragraphe.
- Testez votre code avec: $10 + 5$, $49 - 99$, $50 / 0$, "text" % 10.

Exercice 11: Voir TP JQuery