

JS avancé

Inspiré du livre « TOUT
JavaScript » de
Olivier Hondermarck
aux éditions
DUNOD
et de OCR

SK

Les différentes version de JS

- EcmaScript 6 ou 2015
- EcmaScript 7 ou 2016
- EcmaScript 8 ou 2017

- Depuis 2016, ECMA international édite annuellement une itération de la norme javascript

- 1. Le navigateur envoie une requête HTTP au serveur
- 2. Le serveur fabrique le code HTML de la page demandée et le compresse
- 3. Le serveur envoie le code HTML dans une réponse à la requête HTTP du navigateur
- 4. Le navigateur reçoit en réponse le code HTML et le décompresse
- 5. Le navigateur interprète le code HTML ligne par ligne et commence l'affichage de la page.

- 6. Le navigateur envoie de nouvelles requêtes vers le serveur pour chacun des éléments à charger (images, styles, scripts).
- 7. Le rendu de la page est terminé

- Ouvrir un dossier intelligent
- Visualisation en direct du code: Cmd/Ctrl + Alt + p
- Édition dynamique du CSS: Cmd/Ctrl + E
- Intègre LESS et SASS
- Mise en surbrillance du HTML/CSS // Une sélection dans la page sélectionne le code lié
- Extensions
- Les raccourcis classiques: recherche, sélection multiple (Cmd/Ctrl + clic), mise en commentaire (Cmd/Ctrl + Maj + /), sélections occurrence identique équivalente (b)

- La fonction `alert()`; affiche une boîte de dialogue. Elle prend en paramètre le message à afficher.
- Exercice 1: faites un « Hello world » sous forme d'une boîte de dialogue

- La fonction `log()`; de l'objet console affiche un message dans la console. Elle prend en paramètre le message à afficher.
- La console s'affiche dans le navigateur Chrome avec la touche « F12 ».
- C'est cette fonction que nous utiliserons pour déboguer nos scripts.

- Exercice 2: faites un « Hello world » sous forme d'une message dans la console

- La méthode `window.document.write()`
 - Méthode `write` de l'objet `document` appartenant à l'objet maître `window`. Elle prends en paramètre le texte à afficher.
- Exercice 3: faites un hello world en utilisant cette méthode
- Cette dernière peut également s'écrire sans le `window` qui est implicite.
- Attention: en `nodeJS`, langage coté serveur, l'objet `window` n'est pas disponible

➤ La méthode

```
document.getElementById("monId")
```

➤ La propriété associée à un élément de type HTML `.innerHTML` permet d'écrire du HTML dans le DOM. Attention à sa syntaxe:

<https://developer.mozilla.org/fr/docs/Web/API/Element/innerHTML>

➤ Exercice 4: faites un Hello world en utilisant cette méthode et cette propriété.

➤ Comment aurions nous pu écrire également cet exemple ?

➤ La méthode

```
document.createElement("baliseHtml");
```

➤ La méthode `.appendChild()`; associé à un élément de type HTML et ayant pour valeur une méthode `createElement` est appelée pour ajouter dans le document « baliseHtml ».

➤ Exercice 5: faites un « Hello world » en utilisant ces méthodes

- Sous Chrome:
 - L'onglet « inspecteur d'éléments » accessible avec la flèche tout à gauche qui affiche l'élément sélectionné du DOM
 - L'onglet « responsive » à droite de l'inspecteur d'éléments qui affiche le rendu sous différents types d'appareils
 - L'onglet « Network » qui mesure les temps d'appels aux différents scripts

➤ Téléchargeable à cette adresse:

<https://chrispederick.com/work/web-developer/>

➤ Permet de désactiver javascript

➤ S'utilise dans le serveur local car l'outil de prévisualisation de pages sous Brackets n'active pas les extensions.

La balise <noscript>

- Le contenu de la balise <noscript> s'affiche uniquement pour les visiteurs ayant désactivé javascript.
- Exercice 6: reprenez l'exercice 5 et affichez le message « JS désactivé, navigation impossible » pour les visiteurs ayant désactivé javascript. Essayez le script en activant / désactivant javascript.

- Travaillez sous différents environnements:
 - L'environnement de **développement** qui est votre code hébergé dans un localhost comme MAMP, WAMP ou LAMP lors du dev.
 - L'environnement de **recette - preprod**, est hébergé sur la même plateforme que le site final. Il permet de partager le code avec les autres développeurs et de tester son app dans le même contexte que le site final: même version de PHP, MySQL..
 - L'environnement de **production**: le site

- L'indentation, les espaces, les ; et les règles de nommage
 - A partir de l'exercice 7base, indentiez correctement le code
- Les commentaires, pour expliquer le pourquoi et non le comment et comprendre son code dans le futur
 - En reprenant votre page, ajoutez des commentaires qui vous semblent utiles.
- Ré-utilisabilité et cohérence (garder la même logique: façon d'indenter, nommage etc.

- Voir le cours consacré à GIT sur www.je-code.com

SK

Déclaration de variables (ES<6)

- Une variable se déclare avec le mot clef var
- Elle peut être de type number, string, object, boolean et undefined (dans le cas d'une variable déclarée mais non initialisée)
- On peut obtenir le type d'une variable avec la fonction typeof. Exemple:
 - `var a=1;`
 - `console.log(typeof a);`
- Exo 8: créez une variable ayant pour valeur JS et affichez son type

Déclaration de variables (ES>6)

- Depuis ES6, on peut déclarer des variables avec `const` et `let`
- `Const` est une constante dont la valeur ne peut plus être modifiée une fois définie
- `Let` est une variable dont la portée est restreinte au bloc d'instruction dans lequel il est défini
- Exo 9: définissez une variable à l'aide de `const` et tentez de changer sa valeur

- Le regroupement d'instructions se fait à l'intérieur d'un bloc commençant par une accolade ouvrante { et se terminant par une accolade fermante }
- Un script doit donc contenir autant d'accolade ouvrante que fermante.
- Exercice 10:
 - que vont afficher les console.log ?

➤ Il n'y a pas de `elseif` en javascript. Pour des tests multiples, on utilise le `else if` ou le `switch`.

➤ Syntaxe du `switch`:

```
switch (expression) {  
  case valeur1:  
    blocInstruction1  
  break;  
  case valeurN:  
    blocInstructionN  
  break;  
  default:  
    blocInstructionDefault  
}
```

L'opérateur ternaire

➤ Doit son nom aux 3 parties qui compose son écriture:

Condition ? valeurSiVrai : valeurSiFaux;

➤ Exemple:

```
var port = montant > 100 ? 0 : 6.9;
```

OU

```
nb > 1 ? 's' : '';
```

➤ La négation d'une expression consiste à inverser les ET et les OU. Ainsi, ce code:

```
if ( (a!=1) || (b!=2))
```

Sera équivalent à celui-ci:

```
if ( (a==1) && (b==2))
```

- Les fonctions peuvent s'écrire de 2 façons:

```
function nomFonction(param1, ... paramN){  
    // instruction  
}
```

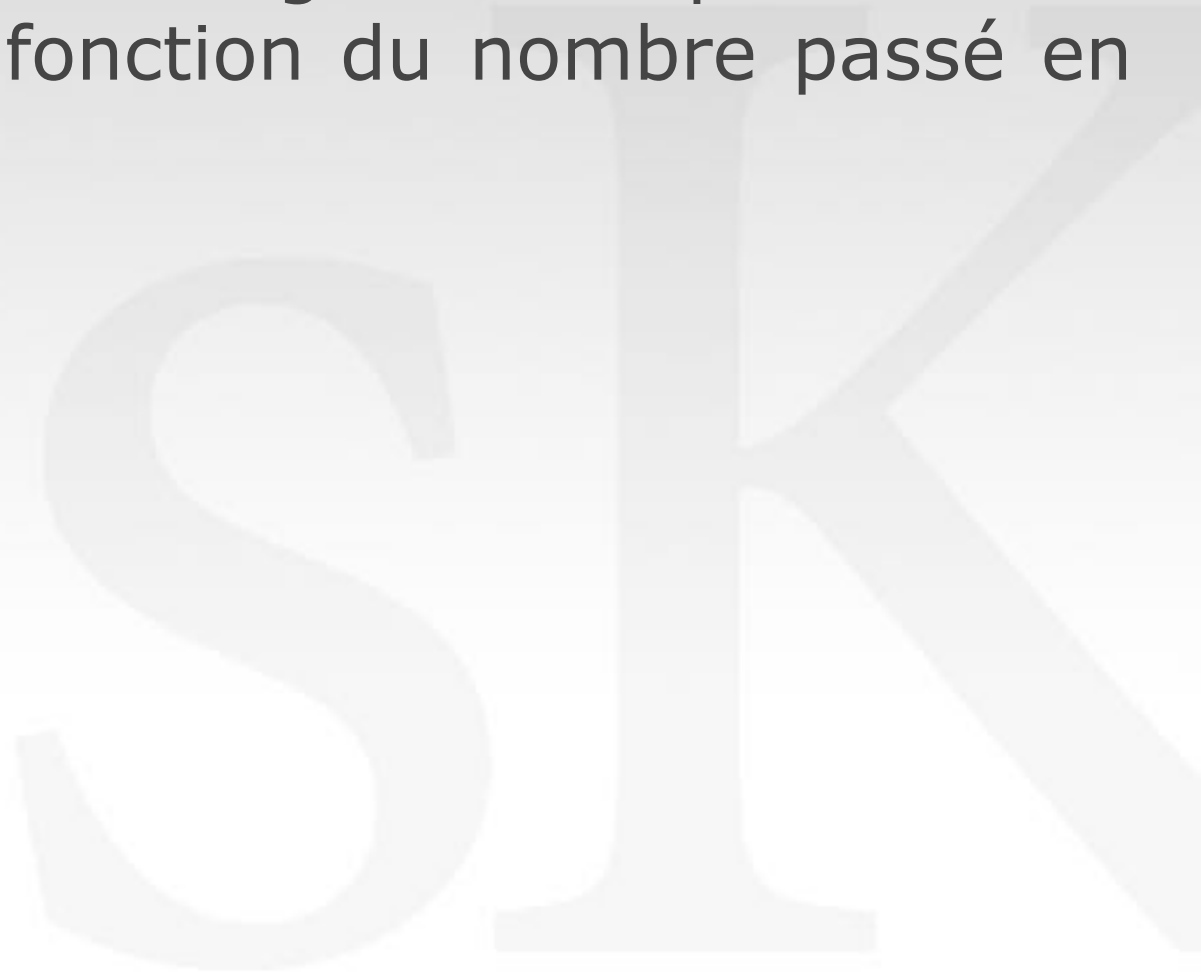
Ou de manière équivalente avec le constructeur function:

```
nomFonction = function(param1, ... paramN){  
    // instruction  
}
```

- La fonction est appelée de la mm manière

Exercice 11

- Ecrivez une fonction `getPluriel` qui retourne un `s` ou `non` en fonction du nombre passé en paramètre.



➤ Syntaxe:

```
(function(param1, ..., paramN){  
    // traitement  
})(p1, ..., pN);
```

➤ Synthétise la déclaration et l'appel en une seule syntaxe

- Les fonctions anonymes:

```
var nombres = [4,8,1000,1500];  
console.log(nombres.every(function(valeur){  
    return (valeur%2==0);  
}));
```

- Peuvent s'écrire sous la forme de fonctions fléchées depuis ES6:

```
console.log(nombres.every(valeur =>  
valeur%2==0));
```

- Cf exemple 12

- Les objets sont constitués de propriétés et de méthodes comme en PHP
- Ils se définissent à l'aide de classe depuis ES6

Le JS orienté objet (exemple)

```
function creerAnimal(nom, age){  
  // propriétés  
  this.nom = nom;  
  this.age = age;  
  // méthode  
  this.log = function(){  
    console.log(this.nom + " a " + this.age + "  
ans.")  
  }  
}
```

```
var monChien = new creerAnimal("Princesse",2);  
monChien.log();
```

Le JS orienté objet (exemple)

Peut s'écrire:

```
class Animal{
  constructor(nom, age){
    // propriétés
    this.nom = nom;
    this.age = age;
    // méthode
    this.log = function(){
      console.log(this.nom + " a " + this.age + " ans.")
    }
  }
}
```

```
var monChat = new Animal("Tigrou",3);
monChat.log();
```

- Exercice 13:
 - Créez une classe véhicule avec pour propriété la marque et le nombre de roues et un log spécifique à la classe
 - Avec les fonctions classiques puis avec les class
 - Instanciez une voiture Renault avec la fonction classique
 - Instanciez une moto Honda avec la class

- Le framework le plus aboutie pour développer des applications en ligne. Seul inconvénient: la courbe d'apprentissage
- Développé par Google. A ne pas confondre avec AngularJS, la première version d'Angular, qui possède des concepts différents.
- Utilise Ionic pour le développement d'application mobile. Aussi, ReactJS, développé par Facebook, est plus performant sur ce dernier point car il génère des composants natifs alors que Ionic génère des webviews. Voir ReactNative.

- Utilise Typescript. Rappel sur Typescript:
 - Utilise un typage strict des variables
 - Transcompilation nécessaire pour transformer le Typescript en JS. Dans ce contexte, angular se charge de l'effectuer.
 - Notions d'interface
 - Propriétés et méthodes en public / private

Angular 8: le CLI

- Angular s'installe en ligne de commande
- Il faut d'abord installer NPM: `npm install -g npm@latest`
- Ensuite installer angular `npm install -g @angular/cli`
- Ensuite créez votre premier projet: `ng new mon-premier-projet`
- Puis lancer le sous chrome sous un système unix (mac, linux): `cd mon-premier-projet`
`ng serve — open`

Angular 8: initialiser un projet

- Créez votre premier projet: `ng new mon-premier-projet --style=scss --skip-tests=true` pour choisir son style en sass (cf cours sur je-code.com) et ne pas avoir de fichiers de tests
- Installez bootstrap: `npm install bootstrap@3.3.7 --save`
- Ouvrez le fichier `angular.json` à la racine de votre projet. Dans `"architect/build/options"`, modifiez l'array `styles` comme suit :
`"styles": [" ./node_modules/bootstrap/dist/css/bootstrap.css", "styles.scss"]`

- Créez votre premier composant:
ng generate component mon-premier



Angular 8: ressources

- Le tutoriel officiel (en anglais): <https://angular.io/tutorial>
- Un cours sur Udemy: <https://www.udemy.com/course/angular-developper-tutoriel-application-typescript/>
- Comprendre la programmation réactive: <https://www.technologies-ebusiness.com/enjeux-et-tendances/rxjs-pour-les-humains>
- Le blog du pape de Angular: <https://johnpapa.net/>
- Et si nécessaire, reprendre JS: <https://code.tutsplus.com/fr/tutorials/the-best-way-to-learn-javascript--net-21954>