

Initiation en JavaScript (Les fonction en javascript)

Training the best tech talent in Africa



PLAN DU WORKSHOP



1. Qu'est-ce que JavaScript ?
2. Présentation
3. Les variables
4. Structures de contrôles
5. Opérateurs
6. Navigation dans l'arbre DOM

Outils d'utilisation



Sublime text



JavaScript



Chrome

1. Qu'est-ce que JavaScript ?

JavaScript

JavaScript est un langage de programmation dit interprété. Il a la particularité de s'activer côté client. Cela signifie que lorsque l'utilisateur ouvre une page web, le navigateur active le code afin d'afficher la page.

JavaScript permet d'**intégrer des scripts entre les éléments HTML de la page, généralement pour ajouter une animation ou un effet visuel**. C'est un langage utile pour améliorer l'expérience utilisateur, car **il permet d'ajouter de l'interactivité** sans pour autant ralentir la page.

Avantages

Le principal avantage de JavaScript réside dans sa sécurité. En effet, si votre navigateur peut récupérer un script dans une page HTML, il n'y a en revanche aucun risque qu'il accède à vos fichiers.

Inconvénients

En tant que langage interprété par les navigateurs, JavaScript peut justement être interprété de différentes manières, d'un navigateur à un autre. Certains navigateurs, trop anciens, ne suivent pas les évolutions du langage et peuvent ne pas être en mesure d'afficher les scripts.

Langages de programmation pour un site web dynamique

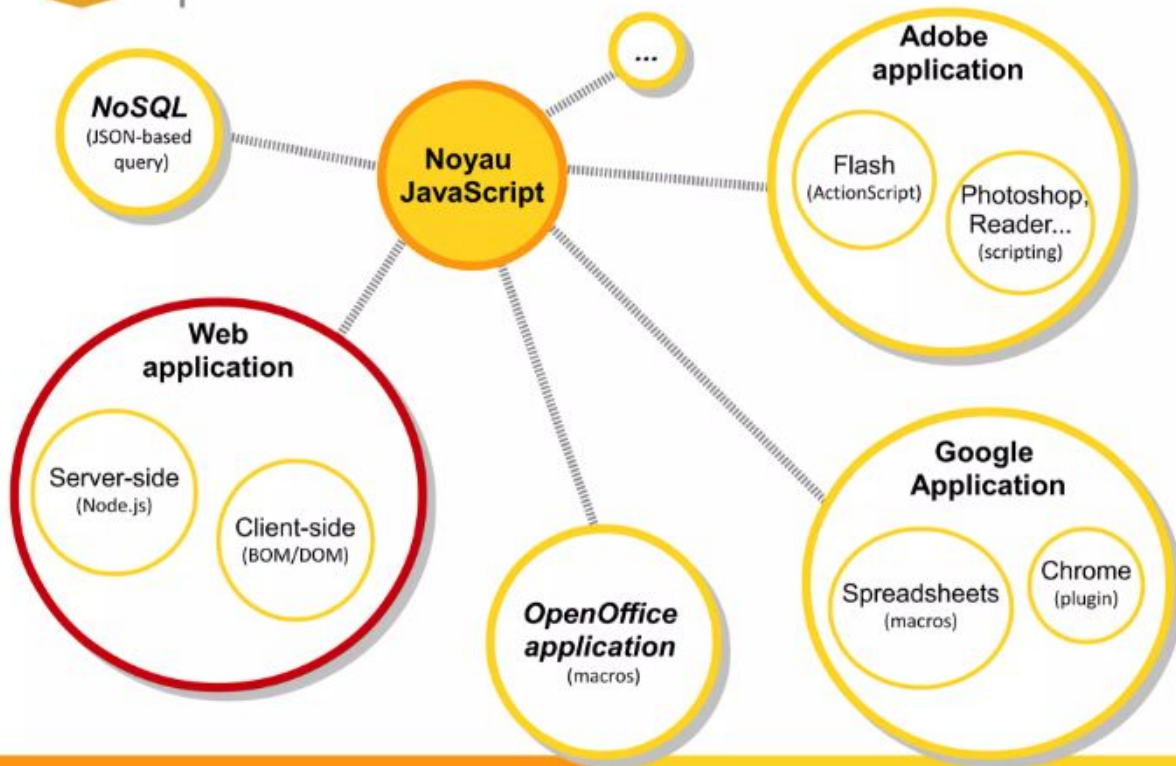
Un site web dit « dynamique » est généralement un site qui gère des données. Celui-ci est alors connecté à un serveur, avec lequel il communique via un langage dédié. Tout comme pour les langages appelés « clients », il existe de nombreux langages de programmation dits « serveurs ». Ce type de développement est appelé *backend*.

2. Présentation

Training the best tech talent in Africa



La galaxie JavaScript





JS, une technologie client

Client Web : Chrome



Execution sur le client
(HTML, CSS, **JavaScript**, ...)

4

1 Requête http



3 Réponse http



Execution sur le serveur
(ASP, PHP, JSP, ...)

2



Aspects lexicaux

- Chaque instruction est séparée par un ;
 - Mais pas obligatoire. Le retour à la ligne suffit.
- Commentaires
 - Par ligne : `// Ceci est un commentaire`
 - Par bloc : `/* Ceci est un commentaire */`
- Déclaration et typage
 - Déclaration avec le mot clé var ou const
 - `var` année; `const` PI = 3.14;
 - Typage dynamique
 - Implicite et peut changer en cours d'exécution...



Sous-programmes JS

- Le code des sous-programmes se situe au niveau de l'entête d'un document Html
- Soit directement

```
<head>
  <script type="text/javascript">
    //ici vos définitions de fonctions/procédures JS
    //...
  </script>
</head>
```

- Soit par inclusion d'un fichier externe

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



Programme principal JS

- Le code du programme principal se situe dans le corps d'un document Html
 - C-a-d les portions de code qui ont vocation à appeler les sous-programmes
- Soit dans une balise spéciale `<script></script>`

```
<body>
  <script type="text/javascript">
    var price = 6.78;
    doSomething(price); //Appel d'un sous-programme
  </script>
</body>
```

- Soit via des événements html



Programmation Objet ?

- Orientation objet
 - Beaucoup de chose dans le langage est considéré comme objet (dont les fonctions !)
- Paradigme objet
 - Un objet est décrit par ses propriétés et ses méthodes
 - Accès aux propriétés et méthodes par un point '.'
 - Auto-référence avec this
 - Visibilité des propriétés et méthodes : public/privé
- Absence du concept de classe
 - On ne peut pas définir de classes d'objets

3. Les variables

Training the best tech talent in Africa



Types primitifs

- Entier

- `var annee = 2014;`

- Réel

- `var prix_ttc = 45.789;` //il s'agit d'un point, pas d'une virgule

- Chaîne de caractère

- `var message="Gangnam style";`

- `var message='Gangnam style';`

- Booléen

- `var estSympa=true;`



Portée et vie des variables

- Variable locale
 - Une variable déclarée à l'intérieur d'une fonction est seulement accessible dans cette fonction
- Variable globale
 - Une variable déclarée en dehors de toute fonction est accessible en tout point du script et de la page
- Cycle de vie d'une variable
 - Une variable locale est supprimée lorsque la fonction se termine
 - Une variable globale est supprimée à la fermeture de la page



Objets primitifs

`typeof(nom_variable)`
permet de connaître
le type de la variable

- Boolean

- `var estSympa = new Boolean(true);`

- Number (entiers et réels)

- `var annee = new Number(2014);`

- Array

- `var saveurs_yop = new Array[2];`

- `saveurs_yop[0] = "banane"; saveurs_yop[1] = "kiwi";`

- String

- `var message = new String("Gangnam style");`

4. Structures de contrôles 1



Structures de contrôles

- Condition

- `if (expr) { ... } else { ... }`

- Boucle

- `while (expr) { ... }`

- `do { ... } while (expr);`

- Iteration

- `for (expr1; expr2; expr3) { ... }`

- `for each (key in object) { ... }`

- `for (value in object) { ... }`

Spécifiques
aux objets

5. Les opérateurs

Training the best tech talent in Africa



Opérateurs arithmétiques

• Opérateurs binaires

Symbole	Fonction	Champs d'application
+	addition	entiers ou réels
+	concaténation	chaînes de caractères
-	soustraction	entiers ou réels
*	multiplication	entiers ou réels
/	division réelle (jamais entière)	entiers ou réels
%	reste de la division entière	entiers

• Opérateurs unaires

Symbole	Fonction	Champs d'application
+	plus	entiers ou réels
-	moins	entiers ou réels
++	pré- ou post-incrémentation	entiers ou réels
--	pré- ou post-décrémentation	entiers ou réels



Opérateurs logiques

Opérateurs binaires

Symbole	Fonction
&&	ET logique
	OU logique

Opérateurs unaires

Symbole	Fonction
!	NON logique

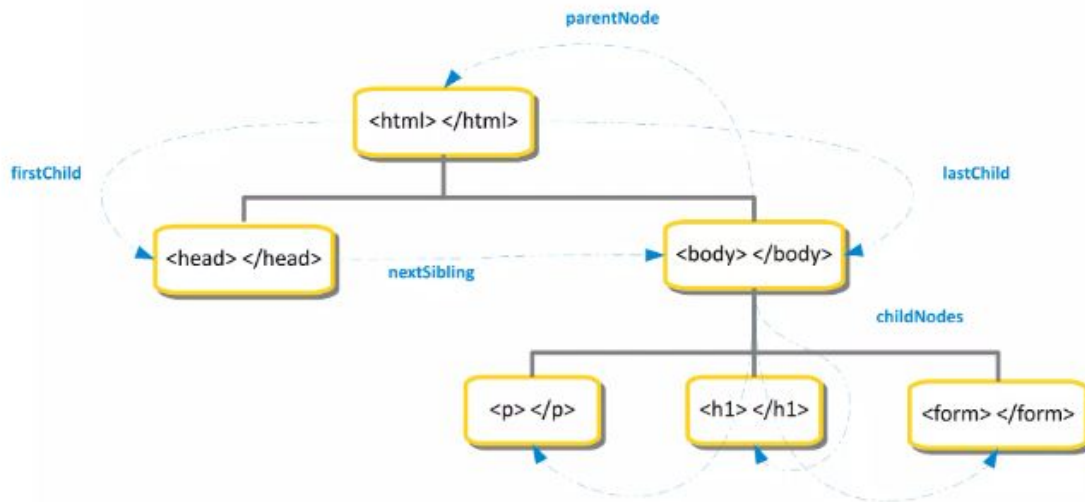
Opérateurs relationnels

Symbole	Fonction
>	supérieur
>=	Supérieur ou égal
<	inférieur
<=	Inférieur ou égal
==	égal
===	identique
!=	différent
!==	non identique

6. **Navigation dans l'arbre DOM**



Navigation dans l'arbre DOM





Méthodes d'un nœud

Méthodes	Commentaires
<code>createElement()</code>	Méthode pour créer un nouvel élément HTML dans le document (div, p, span, a, form, input, etc...).
<code>createTextNode()</code>	Méthode pour créer un nœud texte.
<code>appendChild()</code>	Pour ajouter l'élément créé dans le document. L'élément sera ajouté comme étant le dernier nœud enfant d'un élément parent.
<code>insertBefore()</code>	Pour ajouter l'élément créé avant un autre nœud.
<code>removeChild()</code>	Pour supprimer un nœud.



Accès direct aux nœuds

- Par la valeur de l'attribut id (si il existe)
 - `var result = document.getElementById("intro") ;`
 - Renverra 0 ou 1 résultat
- Par le nom de balise (*Tag* en anglais)
 - `var result = document.getElementsByTagName("input") ;`
 - Renverra 0 ou n résultats
- Par la valeur de l'attribut name (si il existe)
 - `var result = document.getElementsByName("news_email") ;`
 - Renverra 0 ou n résultats



**COACH EN
PROGRAMMATION**

**MOUHAMED
AHMADOU BA**

Training the best tech talent in Africa



