الكلية متعددة التخصصات-ورزازات
ⵜⴰⵢⵜⵍⴰⵢⵜ ⵜⴰⵣⵉⵣⴰⵡⵜ – ⵍⵓⴰⵣⵓⵔⵜ
FACULTÉ POLYDISCIPLINAIRE DE OUARZAZATE

Technologies de web

JavaScript

Prof. Ilyas El jaafari

SMI/IGE 2018

Introduction

- JavaScript est un langage de programmation interprété formalisé dans le standard de langage ECMAScript.
- Javascript permet de créer du contenu mis à jour de façon dynamique.
- Les moteurs (navigateur) JavaScript interprètent et exécutent JavaScript.
- JavaScript n'a aucune relation avec Java
- Script: est une portion de code qui vient s'insérer dans une page HTML
- Le code du script n'est toutefois pas visible dans la fenêtre du navigateur car il est compris entre des balises spécifiques qui signalent au navigateur qu'il s'agit d'un script écrit en langage JavaScript
- Interprété du coté client
- Quand le navigateur rencontre un bloc de JavaScript, il l'exécute généralement dans l'ordre, de haut en bas

Que peut-il vraiment faire?

- Le cœur de JavaScript est constitué de fonctionnalités communes de programmation permettant de :
  - stocker des valeurs utiles dans des variables.
  - faire des opérations sur des morceaux de texte.
  - exécuter du code en réponse à certains évènements se produisant sur une page web.
  - …
- Propose des APIs (code JavaScript déjà prêts)
- Comme l'API DOM (Document Object Model) permet de manipuler du HTML et du CSS
  - créer, supprimer et modifier du HTML
  - appliquer de nouveaux styles à la page de façon dynamique, etc...

Comment ajouter du JavaScript à votre page?

- JavaScript interne

```
<script>
   //mettre ici votre code javascript
</script>
```

- JavaScript externe

1) Créer une page HTML et ajouter :
<script src='monFichier.js'></script>

2) Créer le fichier « monFichier.js » et mettre le code JavaScript dedans .

- les éléments situés dans l'en-tete (<head>) se comportent comme des déclarations, ils ne s'exécutent pas directement
- les éléments situés dans le corps s'exécutent au fur et à mesure du chargement de la page

- Utilisation des attributs de balise pour la gestion évènementielle :

  <balise onEvenement=' instruction javascipt'>…</balise>

Les bases

❑ Les 4 types de base

entier : 127 (base 10), 0755 (base 8), 0xFA15 (base 16)
flottant : 0.123, -0.4e5, .67E-89
booléen : true, false
chaine de caractères : "chaine" ou ’chaine’

- Pas de déclaration des variables:

```
nbr = 10;
fl = 3.141;
str1 = "L’étoile";
str2 = ’brille’;
lien = ’ <a href=“page.html” >click ici</a>’;
```

- Portée des variables:

-locale (uniquement dans le script ou la fonction)

```
var vloc = 0 ;
```

-globale (en tout point du document)

```
vglob = 0 ;
```

Les bases

❑ Expressions

- Arithmétique:
  b=(3+a)*(2.5/30);

- Chaine de caractères:
  ch1= ‘text1’+ a + ‘text2’;

- Logique:
  temp == 37
  h2o = (temp < 100) ? ”eau” : ”vapeur”;

❑ Operateurs

- Affectation:
  +=, -=, *=, /=, %=, …

- Comparaison:
  ==, !=, <, >, <=, >=

- Arithmétique:
  +, -, *, /, %, ++, --

- Logique:
  &&, ||, !

Les bases

Structures de contrôle

- if else
- switch case
- for
- for each in
- while
- do while
- break
- continue

```
for each (var item in obj) {
        somme += item;
}

for (var i = 0; i < 9; i++) {
        str = str + i;
}


if (a > 0) {
        return "positive";
} else {
        return "NOT positive";
}
```

Les fonctions

- Une fonction est une portion de code identifiée par un nom.
- C’est une action à va être exécuter pour atteindre un objectif ou une tâche.
- Les fonctions vous permettent de diviser un problème complexe en tâches plus simples, ce qui facilite la gestion et la maintenance des scripts.
- Un paramètre ou un argument est une donnée qui est transmise à une fonction pour effectuer l'action sur c’est données.
- Les fonctions peuvent recevoir zéro argument ou plus.
- Une fonction est exécutée lorsqu'un appel à cette fonction est effectué n'importe où dans le script, la page, une page externe ou par un événement.
- Les fonctions ont toujours la garantie de renvoyer une valeur lorsqu’elles sont exécutées.
- Les données transmises à une fonction lorsqu'elle est exécutée sont appelées entrée de la fonction et la valeur renvoyée par une fonction exécutée est appelée sortie de la fonction.

Les fonctions

Pour utiliser (ou appeler) une fonction, il faut tout d'abord la définir:

- Definition d'une fonction:

```
function nomfonction(param1, ..., paramN) {
        return param1*…*paramN;
}
```

- Appel d'une fonction:

```
nomVariable = nomfonction(exp1, ..., expN);
```

```
function somme() {
        var argv = somme.arguments;
        var argc = somme.arguments.length;
        var result = 0;
        for (var i = 0 ; i < argc ; i++) {
                result += argv[i];
        }
        return result;
}
somme(1,2,3); // retourne 6
somme(2);     // retourne 2
```

Les évènements

Les événements JavaScript permettent d'intercepter les changements d'états de l'environnement provoqués par le document HTML, les scripts ou l'interaction du client.

- onclick : un clic du bouton gauche de la souris sur une cible
- onMouseOver : passage du pointeur de la souris sur une cible
- onblur : une perte de focus d’une cible
- onfocus : une activation d’une cible
- onselect : une sélection d’une cible
- onchange : une modification du contenue d’une cible
- onsubmit : une soumission d’un formulaire
- onload : un chargement d’une page
- onunload : la fermeture d’une fenêtre ou le chargement d’une page autre que la courante

…

Les évènements

Exemple:

```
<FORM>
        …
        …
        <INPUT type="button" id="bt" onclick="confirm('voulez vous vraiment envoyer ces données');">
</FORM>
```

Explication:

Dans cet exemple, onclick permet de demander au navigateur d'exécuter le code javascript (la fonction confirm()) le moment ou l'utilisateur clique sur le boutton.

- les évènements sont utilisés pour déclencher des fonctions selon les actions de l'utilisateur .

Les objets prédéfinit

- En JavaScript les objets sont aussi des variables. Mais les objets peuvent contenir:
  - plusieurs valeurs (propriétés): chaque valeur est identifiée par un nom,
  - des fonctions (méthodes).

Exemple:

```
<script>
        document.bgColor="blue";
        document.write(document.URL);
</script>
```

- Dans cet exemple:
  - j'ai utilisé l'objet prédéfinit « document », cet objet contient plusieurs propriétés.
  - et la propriété « bgColor » de l'objet document pour modifier la couleur de l'arrière plan de page web.
  - et la propriété « URL » de l'objet document pour récupéré l'url de page web.
  - et la méthode « write() » de l'objet document pour écrire l'url dans la page web

Les objets prédéfinis

- String

Propriétés:

length : nombre de caractères de chaine de caractère.

Méthodes:

charAt() : renvoie le caractère (en fait l'unité de code) situé dans la chaîne à l'offset indiqué, ou en son absence, la chaîne vide.

concat() : concaténer des chaînes de caractères

indexOf() : cherche une sous chaîne à partir d'une position, et renvoie l'offset où se trouve la sous-c et -1 lorsque la sous-chaîne n'est pas trouvée.

……..

Exemple:

```
<script>
        var email = ‘test@test.com’ ;
        var valid= email.indexOf (‘@’) ;
        if(valid != -1){
                alert(‘email valide’);
        } else {alert(‘invalide’);}
</script>
```

Les objets prédéfinis

- Math

  Propriétés:

  PI : 3.14

  Méthodes:

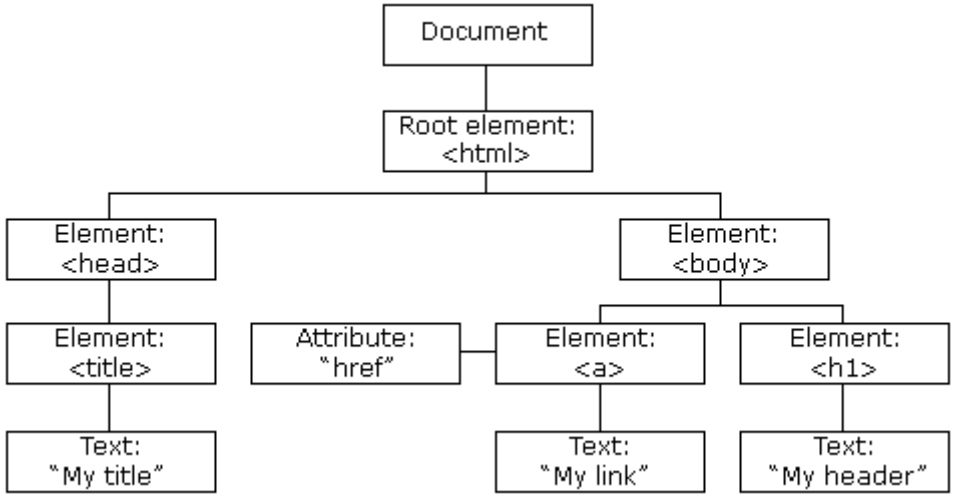
  sqrt() : renvoie le racine carré.
  pow() : puissance
  …

Exemple:

```
<script>
        var rayon = 13;
        var surface = Math.PI * Math.pow(rayon,2);
        alert('la surface est:' + surface);
</script>
```

DOM (Document Object Model)

- DOM: Modèle d'objet de document. Le modèle DOM est une plate-forme et une interface indépendante du langage qui permettent aux programmes et aux scripts d'accéder et de mettre à jour de manière dynamique le contenu, la structure et le style d'un document.
- Lorsqu'une page Web est chargée, le navigateur crée un modèle des objets de document pour cette page.



DOM (Document Object Model)

Avec le modèle objet, JavaScript obtient toute la puissance nécessaire pour créer du code HTML dynamique:

- JavaScript peut changer tous les éléments HTML de la page
- JavaScript peut changer tous les attributs HTML de la page.
- JavaScript peut changer tous les styles CSS de la page
- JavaScript peut supprimer des éléments et attributs HTML existants
- JavaScript peut ajouter de nouveaux éléments et attributs HTML
- JavaScript peut réagir à tous les événements HTML existants dans la page.
- JavaScript peut créer de nouveaux événements HTML dans la page.

API DOM

- Dans le DOM, tous les éléments HTML sont définis en tant qu'objets.
- L'API DOM de JavaScript est constituée des propriétés et méthodes de chaque objet.
- Une propriété est une valeur que vous pouvez obtenir ou définir (par exemple, modifier le contenu d'un élément HTML).
- Une méthode est une action que vous pouvez effectuer (comme ajouter ou supprimer un élément HTML).

Exemple:

```
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

- L'exemple ci-dessus, modifie le contenu (le innerHTML) de l'élément <p> avec id = "demo":
- Dans l'exemple ci-dessus, getElementById est une méthode, alors que innerHTML est une propriété.

API DOM

❑ Trouver des éléments HTML

Méthode	Description
document.getElementById(*id*)	Trouver un élément par id
document.getElementsByTagName(*name*)	Trouver un élément par nom de balise
document.getElementsByClassName(*name*)	Trouver un élément par nom de la classe

❑ Changer les éléments HTML

Méthode	Description
element.innerHTML = *nouveau contenu*	Changer le contenu d'un élément
element.attribute = nouvelle valeur	Changer la valeur de l'attribut d'un élément HTML
element.setAttribute*(attribut, valeur)*	Changer la valeur de l'attribut d'un élément HTML
element.style.*property = nouveau style*	Change le style d'un élément html

API DOM

❑ Ajout et suppression d'éléments

Méthode	Description
document.createElement(*élément*)	Créer un élément HTML
document.removeChild(*élément*)	Supprimer un élément HTML
document.appendChild(*élément*)	Ajout d'un élément HTML
document.replaceChild(nouveau *élément, ancien élément*)	remplacer an HTML élément
document.write(*texte*)	Ecrire dans le document

❑ Ajout de gestionnaires d'événements

document.getElementById(*id*).onclick = function(){*code*}	Ajout de code de gestionnaire d'événement à un événement (onclick comme exemple ici)

API DOM

- Exemple:

```
var myElement = document.getElementById(“monId");

var x = document.getElementsByTagName("p");

var x = document.getElementById("main");
var y = x.getElementsByTagName("p");

document.getElementById("demo").innerHTML = “mon text" ;

document.getElementById("p2").style.color =  “blue" ;

document.getElementById(“img").src =  “image.jpg" ;

var btn = document.createElement("BUTTON");        // creer un element button
btn.innerHTML="clik awa"                           // ajouter du texte au <button>
document.body.appendChild(btn);                    // ajouter <button> à<body>

document.getElementById("myBtn").onclick = displayDate();
function displayDate() {
        document.getElementById("demo").innerHTML = Date();
}
```

Les tableaux

- Les tableaux classiques

```
var cars = ["Saab", "Volvo", "BMW"]; // creer une tableau
```

var cars = new Array("Saab", "Volvo", "BMW"); // 2eme method pour creer un tableau

var name = cars[0]; // recuperer la 1ere valeur du tableau

cars[4] = "Opel"; // ajouter un valeur au tableau a une position

`Cars.push(“fiat”)` // ajouter un valeur au tableau a la fin du tableau

delete fruits[0]; // supprimer l’element 0 du tableau

…

Les tableaux

- Les tableaux classiques : Exemple

```
<script>
var fruits, text, fLen, i;

fruits = ["Banana", "Orange", "Apple", "Mango"];

fLen = fruits.length;

text = "<ul>";

for (i = 0; i < fLen; i++) {
    text += "<li>" + fruits[i] + "</li>";
}

text += "</ul>";
document.getElementById("demo").innerHTML = text;
</script>
```

Les tableaux

- Les tableaux associatifs

```
var person = [];
person["firstName"] = "John";
person["lastName"] = "Doe";
person["age"] = 46;
```

- Exemple:

```
var person = [];
person["firstName"] = "John";
person["lastName"] = "Doe";
person["age"] = 46;
for (ele in person) {
document.write(person[ele]);
}
```