JAVASCRIPT

Introduction

- Evolution du Web statique, dynamique, sémantique, collaboratif, mobile..
- Langages Web
  - **HTML** pour définir le contenu des pages Web (statique)
  - **CSS** pour spécifier la disposition des pages Web
  - **JavaScript** pour programmer le comportement des pages Web (dynamique)

Rappels HTML5

- délimitent les zones du document possédant une valeur sémantique chaque section possède un rôle
- section : Section générique, regroupant une même thématique de contenu, de préférence avec un en-tête.
- article : Section de contenu dans un document ou une application web, dont la composition peut être indépendante du reste de la page et extraite individuellement.
- nav : Section abritant des liens de navigation majeurs, permettant de naviguer au sein du document ou vers d'autres pages.
- aside : Section dont le contenu est tangentiellement lié à ce qui l'entoure, et qui peut être considérée comme séparé de ce contenu.
- header : Section d'introduction

Rappels HTML5

- recueillir les informations d'une entrée effectuée par l'utilisateur
  - text : Champ de texte
  - password : Champ mot de passe
  - hidden : Champ caché (invisible)
  - radio : Bouton radio (un seul choix) checkbox : Case à cocher (choix multiples)
  - button : Bouton
  - reset : Remise à zéro du formulaire
  - submit : Bouton de validation du formulaire
  - image : Image cliquable
  - file : Fichier

Rappels HTML5

- Les nouveaux types
  - tel : Numéro de téléphone
  - url : Adresse URL/URI
  - email : Adresse e-mail
  - search : Champ de recherche date : Date
  - time : Heure
  - datetime : Date et heure
  - datetime-local : Date et heure (locales)
  - month : Mois
  - week : Semaine
  - number : Valeur numérique
  - range : Valeur numérique d'un intervalle, sans grande précision
  - color : Une couleur RVB (3 composantes de 8 bits)

Introduction Javascript

- Mettre le code entre <script> et </script>
- L'attribut type n'est pas requis. JavaScript est le langage de script par défaut en HTML
- Les scripts peuvent être placés dans le <body>, ou dans la section <head> d'une page HTML
- La fonction est appelé lors d'un événement (ex : clic)
- Fichier externe : <script src="myScript.js"></script>
  - Séparer le html et le code
  - Faciliter la lecture
  - Accélérer le chargement avec la mise en cache des scripts

Affichage

- **innerHTML :** Écrire dans un élément HTML .
- **document.write ()** : Écrire dans la sortie HTML
- **window.alert () :** Ecrire dans une boîte d'alerte
- **console.log () :** Écrire dans la console du navigateur

Syntaxe javascript

- La syntaxe du langage Javascript s'appuie sur le modèle de Java et C
  1. L'insertion des espaces peut s'effectuer n'importe où dans le script
  2. Chaque commande doit être terminée par un point-virgule (;).
  3. Un nombre à virgule est séparé par un point (.) et non par une virgule
  4. Le langage Javascript est sensible à la casse
  5. Il existe deux méthodes permettant d'intégrer des commentaires à vos scripts.
     - Placer un double slash (//) devant le texte
     - Encadrer le texte par un slash suivi d'une étoile (/*) et la même séquence inversée (*/)

Variables

- Le mot-clé var permet de déclarer une ou plusieurs variables.
- Après la déclaration de la variable, il est possible de lui affecter une valeur par l'intermédiaire du signe d'égalité (=).
- Si une valeur est affectée à une variable sans que cette dernière ne soit déclarée, alors Javascript la déclare automatiquement.

Variables

- **Déclaration et affectation**
  - La lecture d'une variable non déclarée provoque une erreur
  - Une variable correctement déclarée mais dont aucune valeur n'est affectée, est indéfinie (undefined).
- **La portée**
  - les variables peuvent être globales ou locales.
  - Une variable globale est déclarée en début de script et est accessible à n'importe quel endroit du programme.
  - Une variable locale est déclarée à l'intérieur d'une fonction et n'est utilisable que dans la fonction elle-même.

Variables

- Les noms peuvent contenir des lettres, des chiffres, des traits de soulignement et des signes dollar.
- Les noms doivent commencer par une lettre
- Les noms peuvent aussi commencer par $ et _
- Les noms sont sensibles à la casse (y et Y sont des variables différentes)
- Les mots réservés (comme les mots-clés JavaScript) ne peuvent pas être utilisés comme noms

Variables

- JavaScript inclut aussi deux types de données spéciaux :
  - Null : possède une seule valeur, null, qui signifie l'absence e données dans une variable
  - Undefined : possède une seule valeur, undifined. Une variable dont le contenu n'est pas clair car elle n'a jamais stocké de valeur, pas même null est dite non définie (undifined).

Opérateurs

Operator	Description	Example	Result
+	Addition	x=2 y=2 x+y	4
-	Subtraction	x=5 y=2 x-y	3
*	Multiplication	x=5 y=4 x*y	20
/	Division	15/5 5/2	3 2,5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

Opérateurs

Operateur	Exemple	Est égal à
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

Opérateurs

Operator	Description	Example
==	Est égal à	5==8 faux
===	Vérifie valeur et type	x=5 y="5" x==y vrai x===y faux
!=	différent	5!=8 vrai
>	superieur	5>8 faux
<	inferieur	5<8 vrai
>=	Superieur ou égal	5>=8 faux
<=	Inferieur ou egal	5<=8 vrai

Opérateurs

Operateur	Description	Exemple
&&	and	x=6 y=3 (x < 10 && y > 1) vrai
||	or	x=6 y=3 (x==5 || y==5) faux
!	not	x=6 y=3 !(x==y) faux

Structures conditionnelles

```
if (condition)
{
 instructions ;
}
[ else
{
 instructions ;
}
```

Structures conditionnelles

```
switch (expression)
{
  case étiquette :
    instructions ;
    break ;
  case étiquette :
    instructions ;
    break ;
  default :
    instructions ;
}
```

Structures itératives

```
while (condition)
{
 instructions ;
}

do
{
 instructions ;
}
while (condition) ;
```

Structures itératives

```
for (instr ; condition ; instr)
{
 instructions ;
}

for (variable in objet)
{
 instructions ;
}
```

Commentaires

```
// Commentaire ligne

/* Commentaire multi-lignes */
```

Fonctions

- Le but des fonctions est de regrouper des instructions pour pouvoir les réemployer à plusieurs endroit sans avoir à les réécrire.
- On peut passer à une fonction des paramètres, ceux-ci serviront à moduler l'action de la fonction.
- La fonction peut retourner un résultat, celui-ci étant alors directement exploitable dans une expression.

Fonctions

```
function nom (paramètre){
 // exécuté a chaque appel de la fonction
 return (résultat)
 }
```

- nom : au choix de l'utilisateur
- paramètre : optionnel
- return : renvoie un résultat

Fonctions

- function carre(i) { return (i*i) }
- document.write("La fonction a retourne ",carre(4),".")

Les objets

Les Objets

- Les objets de JavaScript, sont soit des entités pré définies du langage, soit créés par le programmeur.
  - Par exemple, le navigateur est un objet qui s'appelle "**navigator**".
  - La fenêtre du navigateur se nomme "**window**"
  - La page HTML est un autre objet, que l'on appelle "**document**".
  - Un formulaire à l'intérieur d'un "**document**", est aussi un objet.
  - Un lien hypertexte dans une page HTML, est encore un autre objet. Il s'appelle "**link**". etc...
- Les objets javascript peuvent réagir à des "Evénements".
- Tous les navigateurs ne supportent pas les mêmes objets
- Accès aux propriétés d'un objet
  - voiture.couleur.value
  - voiture.couleur.value = verte

Les objets

- **L'opérateur New**
  - L'opérateur *new* est utilisé pour créer une nouvelle instance ou un nouveau type d'objet défini par l'utilisateur ou de l'un des types d'objets prédéfinis, *Array*, *Boolean*, *Date*, *Function*, *Image*, *Number*, *Object*, ou *String*.
  - nouvel_objet = new type_objet(parametres)

```
texte = new String("Une chaîne de caractère");
```

- **L'opérateur Typeof**
  - L'opérateur *typeof* renvoie une chaîne de caractères indiquant quel est le type de l'opérande.

L'objet String

- Propriété :
  - *length* : retourne la longueur de la chaîne de caractères;
- Méthodes :
  - *anchor( )* : formate la chaîne avec la balise <A> nommée;
  - *b( )* : formate la chaîne avec la balise <B>;
  - *big( )* : formate la chaîne avec la balise <BIG>;
  - *charAt( )* : renvoie le caractère se trouvant à une certaine position;
  - *charCodeAt( )* : renvoie le code du caractère se trouvant à une certaine position;
  - *concat( )* : permet de concaténer 2 chaînes de caractères;
  - *fromCharCode( )* : renvoie le caractère associé au code;
  - *indexOf( )* : permet de trouver l'indice d'occurrence d'un caractère dans une chaîne;

L'objet String

- *italics( )* : formate la chaîne avec la balise <I>;
- *lastIndexOf( )* : permet de trouver le dernier indice d'occurrence d'un caractère;
- *link( )* : formate la chaîne avec la balise <A> pour permettre de faire un lien;
- *slice( )* : retourne une portion de la chaîne;
- *substr( )* : retourne une portion de la chaîne;
- *substring( )* : retourne une portion de la chaîne;
- *toLowerCase( )* : permet de passer toute la chaîne en minuscule;
- *toUpperCase( )* : permet de passer toute la chaîne en majuscules;

L'objet Array

- Propriété :
  - *length* : retourne le nombre d'éléments du tableau;
- Méthodes :
  - *concat( )* : permet de concaténer 2 tableaux;
  - *join( )* : converti un tableau en chaîne de caractères;
  - *reverse( )* : inverse le classement des éléménts du tableau;
  - *slice( )* : retourne une section du tableau;
  - *sort( )* : permet le classement des éléments du tableau;

L'objet Math

- Propriétés :
  - Math.PI // retourne PI
  - Math.SQRT2 // racine carrée
  - Math.LN2 // retourne l'algorithme naturel
  - Math.LOG2E
  - Math.LOG10E

L'objet Math

- Méthodes :
  - *abs( ), exp( ), log(), sin( ), cos( ), tan( ), asin( ), acos( ), atan( ), max( ), min( ), sqrt( )* sont les opérations mathématiques habituelles;
  - *atan2( )* : retourne la valeur radian de l'angle entre l'axe des abscisses et un point;
  - *ceil( )* : retourne le plus petit entier supérieur à un nombre;
  - *floor( )* : retourne le plus grand entier inférieur à un nombre;
  - *pow( )* : retourne le résultat d'un nombre mis à une certaine puissance;
  - *random( )* : retourne un nombre aléatoire entre 0 et 1;
  - *round( )* : arrondi un nombre à l'entier le plus proche.

L'objet Date

- Propriété : aucune;
- Méthodes :
  - *getFullYear( ), getYear( ), getMonth( ), getDay( ), getDate(), getHours( ), getMinutes( ), getSeconds( ), getMilliseconds( )*: retournent respectivement l'année complète, l'année (2chiffres), le mois, le jour de la semaine, le jour du mois, l'heure, les minutes, les secondes et les millisecondes stockés dans l'objet *Date*;
  - *getUTCFullYear( ), getUTCYear( ), …* retournent respectivement l'année complète, l'année (2chiffres), … stockés dans l'objet *Date* en temps universel;
  - *setFullYear( ), setYear( ), …* remplacent respectivement l'année complète, l'année (2chiffres), … dans l'objet *Date*;

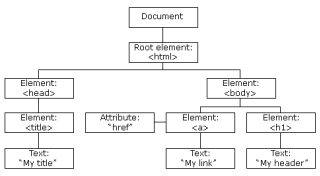
L'objet Date

- *setUTCFullYear( ), setUTCYear( ), …* remplacent l'année complète, l'année (2chiffres), … dans l'objet *Date* en temps universel;
- *getTime( )* : retourne le temps stocké dans l'objet *Date*;
- *getTimezoneOffset( )* : retourne la différence entre l'heure du client et le temps universel;
- *toGMTString( ), toLocaleString( ), toUTCString( )* : convertissent la date en chaîne de caractère selon la convention GMT, selon la convention locale ou en temps universel;

Le DOM HTML (Document Object Model)

Introduction

- Chargement de page Web : le navigateur crée un **D**ocument **O**bjet **M**odèle de la page
- Modèle DOM HTML : arbre d'objets



Javascript et DOM

- 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 les 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

DOM

- Norme W3C : une norme d'accès aux documents
- 3 parties :
  - Core DOM - modèle standard pour tous les types de documents
  - XML DOM - modèle standard pour les documents XML
  - HTML DOM - modèle standard pour les documents HTML

DOM HTML

- Modèle **objet** standard et **une interface de programmation** pour HTML. Il définit:
- Les éléments HTML en tant **qu'objets**
- Les **propriétés** de tous les éléments HTML
- Les **méthodes** pour accéder à tous les éléments HTML
- Les **événements** pour tous les éléments HTML

Méthodes DOM HTML

- Le DOM HTML est accessible avec JavaScript Dans le DOM, tous les éléments HTML sont définis en tant **qu'objets** .
- L'interface de programmation est : les propriétés et les méthodes de chaque objet.
  - Une **propriété** est une valeur que vous pouvez obtenir ou définir (comme changer le contenu d'un élément HTML).
  - Une **méthode** est une action que vous pouvez effectuer (comme ajouter ou supprimer un élément HTML).

Méthodes DOM HTML

- La méthode getElementById
  - accéder à un élément HTML : utiliser l'identifiant de l'élément
- La propriété innerHTML
  - obtenir ou remplacer le contenu des éléments HTML
- Exemple 1

document DOM HTML

- L'objet document : page Web
- Éléments html :
  - document.getElementById(*id*) : Sa valeur est unique dans tout le document
  - document.getElementsByTagName : Elle retourne un tableau qui renferme tous les nœuds du type désigné.
  - document.getElementsByClassName : par le nom de la classe

document DOM HTML

- Modification des éléments HTML
  - element.innerHTML = nouveau contenu html
  - element.getAttribute : permet de récupérer la valeur de l'attribut passé en paramètre.
    - alert(document.getElementById("publicite").getAttribute("src"))
  - element.setAttribute(attribut, valeur) : = nouvelle valeur : changer dynamiquement la valeur d'un attribut d'une balise
    - document.getElementsByTagName("div").item(0).setAttribute("align","center");
  - element.style.property = nouveau style

document DOM HTML

- Ajout et suppression d'éléments
  - document.createElement(*element*)
  - document.removeChild(*element*)
  - document.appendChild(*element*)
  - document.replaceChild(*element*)
  - document.write(*text*)
- Gestionnaire d'évènements
  - document.getElementById(*id*).onclick = function(){*code*}

DOM CSS

- changer le style d'un élément HTML
  - document.getElementById(*id*).style.*property = new style*
  - *Exemple 5*
- Le DOM HTML vous permet d'exécuter du code lorsqu'un événement se produit.
  - Cliquer
  - Chargement de la page
  - Les champs de saisie sont modifiés

DOM : objet Window

- Propriétés
  - frames[ ] : tableau de frames
  - frames.length : nombre de frames
  - self : fenêtre courante
  - opener : la fenêtre (si elle existe) qui a ouvert la fenêtre courante
  - parent : parent de la fenêtre courante, si la fenêtre courante est une sous-partie d'un frameset
  - top : fenêtre principale (qui a crée toutes les fenêtres)
  - status : message dans la barre de statut
  - defaultstatus : message par défaut de la barre de statut
  - name : nom de la fenêtre

46

DOM : objet Window

- Méthodes
  - alert(string) : ouvre une boîte de dialogue avec le message passé en paramètre
  - confirm : ouvre une boîte de dialogue avec les boutons OK et cancel
  - blur() : enlève le focus de la fenêtre
  - focus() : donne le focus à la fenêtre
  - prompt(string) : affiche une fenêtre de saisie
  - scroll(int x, int y) : positionnement aux coordonnées (x,y)
  - open(URL, string name, string options) : ouvre une nouvelle fenêtre contenant le document identifié par l'URL
  - close() : ferme la fenêtre

47

DOM : objet Document

- Propriétés
  - title : titre du document
  - location : URL du document
  - lastModified : date de dernière modification
  - referrer : URL de la page d'où arrive l'utilisateur
  - bgColor : couleur de fond
  - fgColor : couleur du texte
  - linkColor vlinkColor alinkColor : couleurs utilisées pour les liens hypertextes

48

DOM : objet Document

- Propriétés
  - forms[ ] : tableau des formulaires de la page
  - forms.length : nombre de formulaire(s) de la page
  - links[ ] : tableau des liens de la page
  - links.length : nombre de lien(s) de la page
  - anchors[ ] : tableau des ancres internes (<A NAME= …>)
  - anchors.length : nombre de d'ancre(s) interne(s)
  - images[ ]
  - applets[ ] tableaux des images, applets et plug-ins
  - embeds[ ]
  - *Remarque : les tableaux contiennent les éléments dans l'ordre de leur apparition dans le code HTML*

49

DOM : objet Document

- Méthodes
  - write(string) : écrit une chaîne dans le document
  - writeln(string) : idem + caractère de fin de ligne
  - clear() : efface le document
  - close() : ferme le document

50

DOM : objet Form

- Propriétés
  - name : nom (unique) du formulaire
  - method : méthode de soumission (0=GET, 1=POST)
  - action : action déclenchée par la validation du formulaire
  - target : fenêtre de destination de la réponse (si elle existe)
  - elements[ ] : tableau des éléments du formulaires
  - length : nombre d'éléments du formulaire

51

DOM : objet Form

- Méthodes
  - submit() : soumet le formulaire
  - reset() : ré-initialise le formulaire
- Événements
  - onSubmit(method) : action à réaliser lorsque le formulaire est soumis
  - onReset(method) : action à réaliser lorsque le formulaire est ré-initialisé

52

DOM : objet Navigator

- Propriétés
  - appCodeName : nom de code interne du navigateur
  - appName : nom réel du navigateur
  - appVersion : version du navigateur
  - userAgent : objet complexe contenant des détails sur :
    - l'appCodeName,
    - l'appVersion
    - le système d'exploitation utilisé
  - plugins[] : tableau des plugins installés chez le client
  - mimeType[] : tableau des types MIME supportés par le navigateur

53

DOM : objet Navigator

- Méthodes
  - javaEnabled : retourne TRUE si le navigateur supporte Java (et que l'exploitation de Java est actif)

54

Evenements javascript

- 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

DOM EventListener

- La fonction attachée à un évènement est appelée fonction «gestionnaire d'événement » – event handler – ou « d'écoute » – event listener
- La méthode addEventListener réalise l'abonnement d'une fonction à un évènement donnée pour l'objet sur lequel elle est invoquée . objet.addEventListener(eventType, listenerFunction)
  - objet : l'objet ciblé : window, ou un élément de la page
  - eventType : une chaîne de caractèeres désignant l'événement concerné "click", "load", "change", "mouseover", "keypress" etc.
  - listenerFunction : la fonction listener qui est appelée lorsque l'événement se produit

DOM EventListener

- Plusieurs gestionnaires d'événements à un élément (Exemple)
- Exemple

- Passer des paramètres
- Exemple

- La méthode removeEventListener ()

Réferences

- www.w3schools.com/

- Cours javascript (dupont)