ESI
مدرسة علوم المعلومات
ⵜⵉⵎⵔⵔⴰⵜ ⵏ ⵜⵓⵙⵙⵏⵉ ⵏ ⵜⵣⵔⴼⵜ
ECOLE DES SCIENCES DE L'INFORMATION

Technologies Web

Pr. Fatima MOURCHID

Plan

- Instructions de base du langage Javascript (Variables, itérations, tableaux, fonctions, objets, etc…)
- DOM (Document Object Model)
- Framework Javascript (jQuery, Ajax, Angular.js, Node.js, etc…)
- Format JSON (JavaScript Object Notation)

Langage Javascript

Introduction

- Developpemnt Web: Statique / dynamique
- HTML
  - Statique
  - Images
  - CSS
  - Design créatif afin de développer des pages Web attractives
- HTMl n'est pas un lagnage de programamtion
  - Markup language: utilise des balises pour définir des pages Web

Introduction

- Langage de programmation
  - Réaliser des actions étape par étape
  - Réaliser des actions répétitives
  - Réaliser des actions comme résultat des données introduites par l'utilisateur
- Javascript est une langage de programmation
  - Développer des pages web interactives
  - Réagir suite aux actions/données des utilisateurs

C’est quoi Javascript?

- Javascript développé par Netscape 1995
- JavaScript: the core language (ECMAScript)
  - Standardisé par ECMA TC39 committee en 1997
  - Utilisé dans des environments autre que le Web tel que node.js
  - Versions ECMAScript: ES1, ES2, ES3, ES5, and ES6
  - Depuis 2016, les versions sont nommées: ECMAScript 2016 / 2017 / 2018
- Langage de scripts (programmes interprétés) exécutés sur la machine du client
  - Une page Web devient réactive à un événement si à cet événement est associée une fonction javascript
- Javascript #Java (même si tous les 2 sont orientés-objet)
  - JavaScript est toujours interprété côté client, alors que Java peut être compilé ou interprété côté serveur
  - Code JavaScript est appelé dans une page Web différemment d'une applet Java
  - Il est nécessaire de déclarer les variables en Java, alors que cette déclaration peut être implicite en JavaScript

- Ecma(European Computer Manufacturers Association): Industry association for standardizing information and communication systems
- Ecma International's TC39 is a group of JavaScript developers, implementers, academics, and more, collaborating with the community to maintain and evolve the definition of JavaScript

C’est quoi Javascript?

- Javascript est basé sur les objets
  - Un objet est une entité avec des propriétés/attributs et un comportement défini par des méthodes
- Objets Javascript
  - objets issus du document html
  - objets prédéfinis dans Javascript
  - objets créés par l’utilisateur
- JavaScript “epmrunte” sa syntaxe depuis Java, C, and C++. Il est aussi influencé aussi par Awk, Perl et Python.

Insérer du code Javascript

- Utilisation des attributs prévus dans les recommandations (X)HTML : onclick, onmouseover, etc.

```
<a href="#top" onclick="alert('Bonjour!');">lien</a>
```

- Insérer le code JavaScript entre les balises <script> et </script> (dans l'entête du fichier HTML<head> ou au cœur de la page HTMl <body>)

```
<script type="text/javascript"  >(...)</script>
```

- Insérer le code JavaScript qui se trouve dans un fichier externe

```
<script src="../scripts/monfichier.js">
```

Syntaxe générale

- Convention d'écriture
  - Chaque instruction de code doit se terminer par « ; »
  - JavaScript est sensible à la casse : variabletest et Variabletest désignent deux variables différentes
  - Noms réservés: var, function, for, if...

- Commentaire JavaScript
  - Quand le commentaire s'étend sur une seule ligne, on peut commencer cette ligne par //
  - Quand le commentaire s'étend sur plusieurs lignes, il doit commencer par /* et se terminer par */
    - //Une ligne
    - /* plusieurs

      …

      lignes*/

Variables Javascript

- Types de données
  - Chaîne de caractères: délimitée par des " ou ‘
    - Caractères échappés : \b correspond au retour arrière, \f au saut de page, \n au saut de ligne, \r au retour chariot, \t à la tabulation horizontale, \\ au caractère \ lui-même
  - Nombre: entier ou nombre à virgule flottante
  - Type boolean: « true » ou « false »
  - Type object : déclarer les objets en JavaScript
  - undefined: une variable ne peut pas être définie ou ne l'a pas encore été
  - null: une variable qui ne contient aucune donnée
  - Constantes
    - positive Infinity ou +Infinity
    - negative Infinity ou –Infinity
    - NaN (Not A Number) qui est obtenu lorsque l'on essaie de réaliser une opération interdite (comme par exemple division par zéro)

Variables Javascript

- Conversion de type
  - parseInt() et parseFloat():transformer une chaîne de caractères en nombre
  - Number(): transformer un objet en nombre pour les chaînes de caractères
  - string() : transformer un objet en chaîne de caractères

- Tests sur les types
  - typeof : renvoie une chaîne de caractères donnant le type d'une variable
  - isFinite(): tester si la variable passée en paramètre est bien un nombre fini
  - isNaN(): tester si le paramètre n'est pas un nombre

Variables Javascript

- Déclaration et affectation de valeurs
  - =: affectation
  - a=b=c=d=5: plusieurs affectations à la fois
  - Déclaration d'une variable sans affectation de valeur
    - var X: déclarer une variable sans lui avoir affecté de valeur, alors elle est de type undefined
- Opérateur d’affectation complexe
  - += ; permet de réaliser une addition
  - -= : permet de réaliser une soustraction
  - *= : permet de réaliser une multiplication
  - /= : permet de réaliser une division

Variables Javascript

- Opérateur ternaire: permet d'affecter une valeur à une variable en fonction du résultat à un test

  variable=TestARéaliser ?valeurSiTestVrai:ValeurSiTestFaux

  - Exemple: a=b>3/?0:1 affecte la valeur 0 à a si b>3 sinon la valeur 1
- Portée
  - Si une variable est déclarée sans le mot-clé var: variable globale
  - Si une variable est déclarée avec le mot-clé var: variable locale
  - let: permet de limiter la portée d'une variable à un bloc (compris entre deux accolades)
  - const: on ne peut pas affecter une nouvelle valeur à une variable déclarée const

Opérateurs

- Opérateurs mathématiques
  - +:addition
  - -:soustraction
  - /: division
  - %: reste de la division
- Opérateurs de comparaison
  - < et <=: inférieur et inférieur ou égal
  - > et >=: supérieur et supérieur ou égal
  - == et !=: égalité

Opérateurs

- Opérateurs logiques
  - &&: AND
  - || : OR
  - ! : négation
- Opérateurs unaires
  - !: négation
  - ++: incrémentation en préfixe ou postfixe
  - --: décrémentation en préfixe ou postfixe

Tableaux

- Déclaration

```
var tableau1 = new Array(4) ;
var tableau2 = new Array() ;
```

- Affectation
  - Affectation des valeurs une par une

```
var tableau1 = new Array(4) ;
tableau1[0]="Beurre" ; tableau1[1]="Confiture" ;tableau1[2]="Pain" ; tableau1[3]="Lait"
```

Tableaux

- Affectation par une seule instruction

```
var tableau1 = new Array(4) ;
tableau1=["Beurre", "Confiture", "Pain", "Lait"] ;
```

- Déclaration et affectation en une seule instruction

```
var tableau1 = new Array('Beurre', 'Confiture', 'Pain', ‘Lait’) ;
```

- Fonction Length(): taille du tableau

```
alert(tableau1.length) ;
```

Tableaux

- Tableaux à plusieurs dimensions
  - Déclaration suivie de l'affectation

```
var monTableau = new Array(10);
for (var i = 0; i < 10; i++)
{
 monTableau[i] = new Array(20);
}
monTableau[0][0] = 'choix1'; //On affecte une valeur de la même façon qu'on l'affiche
```

  - Déclaration et affectation

```
var monTableau = [ [Paul, Jean], [Pierre, Marc]];
console.log(items[0][0]); // Affiche Paul
```

Collections: Set

- Set: une collection de valeurs et une valeur est unique dans la collection

```
let mySet = new Set();
mySet.add(1);
mySet.add('some text');
mySet.add('foo’);
(…)              // instructions
mySet.has(1); // true
mySet.delete('foo');
mySet.size; //   la taille de l’ensemble devient  2
```

Itérations

- Boucle for
  - instruction de départ
  - condition de fin
  - instruction à exécuter à chaque fin d'itération

```
var tableau1 = new Array(4) ;
 for (i=0;i<3;i++)
  {
        tableau1[i]=i ;
  }
```

Itérations

- Boucle for… in

```
var monTableau = new Array(3) ;
for (element in monTableau)
 {
   element="test" ;
 }
```

Itérations

- do… while et while
  - var compteur = 1 ;

```
do
 {compteur++ ;
   alert(compteur) ;  }
while (compteur<3)
```

  - var compteur = 1 ;

```
while (compteur<3)
 { compteur++ ;
   alert(compteur) ;
 }
```

Tests logiques

- Instruction If

```
if (choix==1)
{ alert("Vous avez fait le premier choix") ;}
Else if (choix==2)
{ alert("Vous avez fait le deuxième choix") ;}
Else if (choix==3)
{ alert("Vous avez fait le troisième choix") ;}
Else {alert("Vous n'avez pas fait de choix") ;}
```

Tests logiques

- Instruction switch

```
switch (choix)
{
case 1: alert("Vous avez fait le premier choix") ;
break ;
case 2: alert("Vous avez fait le deuxième choix") ;
break ;
case 3: alert("Vous avez fait le troisième choix") ;
break ;
default : alert(« Vous n’avez pas fait de choix") ;
}
```

Fonctions Javascript

- Déclaration

  function nomFonction(liste éventuelle des arguments)

  {

  (…)

  }

- Appel
  - Une fonction est appelée par: nomFonction (arguments)
  - Une fonction peut éventuellement « retourner » une valeur à l'aide de l'instruction return

Fonctions Javascript

- Valeur de paramètres par défaut

```
function nomFonction(a=30, b=20)
 {
   return a*b ;
 }
```

- Obtenir la liste des arguments

```
var moyenne = function(){
  let somme = 0 ;
  for (let i=0;i<arguments.length;i++) somme+=Number(arguments[i]) ;
  console.log (somme/arguments.length) ;
}
```

Objets prédéfinis

- Objet String
  - Propriété: length
  - Méthodes

Méthode	Rôle
indexOf	position de la première occurrence d'un caractère donné dans une chaîne.
toUpperCase	convertit la chaîne en majuscules.
toLowerCase	convertit la chaîne en minuscules
substring	extrait une sous-chaîne de la chaîne à laquelle elle s'applique
concat	concatène la chaîne avec la ou les chaînes passées en argument
includes	teste si une chaîne de caractères donnée est incluse dans une autre, et renvoie un booléen

Objets prédéfinis

- Objet Math

Propriété	Rôle
E	constante d'Euler
PI	nombre pi
LN2	logarithme naturel de 2
LN10	logarithme naturel de 10
LOG2E	logarithme en base 2 de e
LOG10E	logarithme en base 10 de e
SQRT2	racine carrée de 2
SQRT1_2	racine carrée de 1/2

Objets prédéfinis

- Objet Math

Méthode	Rôle
abs	valeur absolue du paramètre
round	arrondi du nombre passé en paramètre
random	nombre pseudo-aléatoire compris entre 0 et 1
pow	élève à la puissance du second paramètre le nombre passé en premier paramètre

Objets prédéfinis

- Objet Date
  - permet de gérer les dates, les heures et durées

```
var nouvelleDate = new Date(annee, mois, jour [, heure, minute, seconde,
                                                milliseconde]) ;
```

  - Aujourdhui=new Date(); //la date renvoyée est la date courante

Méthode	Rôle
getDate	quantième dans le mois pour la date spécifiée
getTime	nombre en millisecondes écoulées depuis le 1er janvier 1970 jusqu'à la date spécifiée
getDay	jour de la semaine pour la date spécifiée (0=dimanche, 6=samedi)

Objets prédéfinis

- Objet window

Méthode	Rôle
alert(chaine)	affiche à l'écran le message d'information contenu dans chaine. Dans la boîte ne se trouve qu'un bouton OK
confirm(chaine)	affiche la chaîne de caractères chaine accompagnée de deux boutons OK et Annuler
prompt(chaine)	affiche la chaîne de caractères « chaine » et une champ de saisie. La méthode retourne sous la forme d'une chaîne de caractères le résultat de la saisie de l'utilisateur

Objets prédéfinis

- Ajouter des méthodes et propriétés aux objets prédéfinis

```
//On vérifie que la méthode n'existe pas déjà
if (!String..prototype..capitalize)
 {
   Object.defineProperty(String..prototype,'capitalize',
    {
      value : function()
       {  return this..charAt(0)..toUpperCase()+this..slice(1)..toLowerCase() ; // mettre en majuscule la première lettre d'une chaîne

}
}) ;
 }
(…)
var test="Essai";
alert(test.capitalize());
```

Classes dans Javscript

- Classes: templates pour la création d'objets. Elles encapsulant les données et le code pour manipuler ces données
  - class Polygon {

```
 constructor(height, width) {
  this.area = height * width;  }
}
(…)   // instructions
console.log(new Polygon(4, 3).area);// 12
```

  - class ClassWithStaticInitializationBlock {

```
static staticProperty1 = 'Property 1';
static staticProperty2;
static {    this.staticProperty2 = 'Property 2';  }}    // bloc d'initialisation statique
console.log(ClassWithStaticInitializationBlock.staticProperty1);// output: "Property 1 "
console.log(ClassWithStaticInitializationBlock.staticProperty2);// output: "Property 2"
```

Temporisation

- Fixer un délai pour une exécution : setTimeout et clearTimeout

```
function fonctionTest(i)
 {
   while (i>0)
    { alert(i) ;
      setTimeout("fonctionTest(i)", 1000) ;
    }
   i-- ;
 }
(…)
 clearTimeout(identifiant);
```

- Répéter une action: setInterval et clearInterval

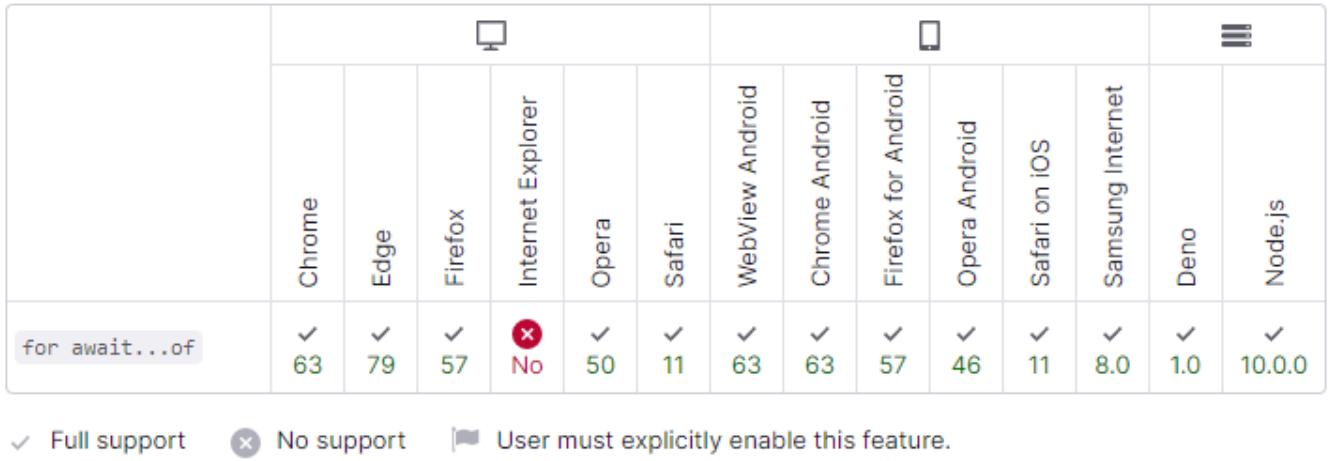
```
identifiant=setInterval("fonctionARepeter()", 1000) ;
(…)
clearInterval(identifiant);
```

Gestion des exceptions

- Throw et try … catch : gestion des exceptions au moment de l'exécution

```
function doSomethingErrorProne() {
  if (ourCodeMakesAMistake()) {
    throw (new Error('The message'));
  } else {
    doSomethingToGetAJavascriptError();
  }
}
(…) // instructions
try {
  doSomethingErrorProne();
} catch (e) {
  console.error(e.name);    // affiche le code de l'erreur
  console.error(e.message); // affiche le message de l'erreur
}
```

Compatibilité pour les navigateurs



	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	WebView Android	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	Deno	Node.js
for await...of	✓ 63	✓ 79	✓ 57	✗ No	✓ 50	✓ 11	✓ 63	✓ 63	✓ 57	✓ 46	✓ 11	✓ 8.0	✓ 1.0	✓ 10.0.0

✓ Full support ✗ No support User must explicitly enable this feature.

Web browser compatibility for « for await...of »

https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Statements/for-await...of

TPs: enoncés

- TP1: Écrire un script qui affiche le message "Hello World !" sur la page
- TP2: Écrire un script qui calcule la factorielle d’un nombre entier positif n
- TP3: Écrire un script qui permet d'afficher la table de multiplication d'un nombre entier n
- TP4: Écrire un script qui génère un nombre aléatoire
- TP5: Écrire un script qui programme un chronomètre

Bibliographie

- JavaScript: https://developer.mozilla.org/en-US/docs/Web/JavaScript
- Le Modèle Objet de Document (DOM: Document object Model): https://developer.mozilla.org/fr/docs/Web/API/Document_Object_Model/Introduction
- Document Object Model (DOM) Specification: http://www.w3.org/TR/dom/
- Le site officiel de jQuery: https://jquery.com/
- Introduction à jQuery: https://openclassrooms.com/fr/courses/3504441-introduction-a-jquery
- Le site officiel de Angular: https://angularjs.org/
- Introduction to Web Development: https://www.coursera.org/learn/web-development
- Introduction to Web Development with HTML, CSS, JavaScript: https://www.coursera.org/learn/introduction-to-web-development-with-html-css-javacript
- HTML Tutorial: https://www.w3schools.com/html/default.asp
- JavaScript Tutorial: https://www.w3schools.com/js/default.asp
- JSON Tutorial: https://www.w3schools.com/js/js_json_intro.asp