



L'école supérieure de la statistique et de l'analyse de l'information

JavaScript

Enseignante : **MANSOUR Sihem**

Contact: sihemmansour@yahoo.fr



Qu'est ce que JavaScript

- JavaScript est un langage de programmation créé en 1995 par Brendan Eich et intégré au navigateur web Netscape Navigator 2.0.

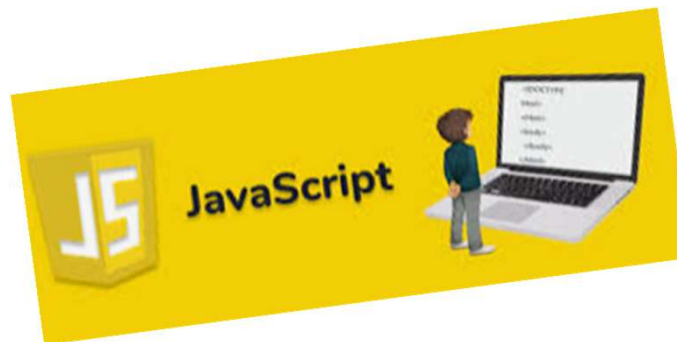


- JavaScript (« JS » en abrégé) est un langage de programmation dynamique complet qui, appliqué à un document HTML, peut fournir une interactivité dynamique sur les sites Web.



Qu'est ce que JavaScript

- JavaScript est d'une incroyable flexibilité. Vous pouvez commencer petit, avec des carrousels, des galeries d'images, des variations de mises en page et des réponses aux clics de boutons.
- Avec plus d'expérience, vous serez en mesure de créer des jeux, des graphiques 2D et 3D animés, des applications complètes fondées sur des bases de données et bien plus encore !





Premier contact avec JavaScript

- Vous pouvez utiliser deux manières pour injecter le JavaScript dans votre page HTML. Dans La première, vous écrivez le script dans la page HTML en utilisant la balise script.
- Cette injection peut être faite dans la partie Head ou la partie body. Voir l'exemple:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Où écrire le JavaScript ?</title>
    <meta charset="utf-8">
    <script>
      alert('Ceci est affiché en JavaScript !');
    </script>
  </head>
  <body>
    <h1>On peut écrire le JavaScript dans...</h1>
    <ul>
      <li>L'élément head d'un fichier HTML</li>
      <li>L'élément body d'un fichier HTML</li>
      <li>Un fichier ".js" séparé</li>
    </ul>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>Où écrire le JavaScript ?</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>On peut écrire le JavaScript dans...</h1>
    <ul>
      <li>L'élément head d'un fichier HTML</li>
      <li>L'élément body d'un fichier HTML</li>
      <li>Un fichier ".js" séparé</li>
    </ul>
    <script>
      alert('Ceci est affiché en JavaScript !');
    </script>
  </body>
</html>
```



Appeler un script JavaScript depuis un autre fichier dans une page HTML

- La deuxième manière consiste à écrire le script dans un fichier à part puis l'appeler dans la page HTML. Cette manière est plus conseillée
- L'appel du fichier doit se faire avant la fermeture de la balise body.

• Voir l'exemple:

```
javascript > JS app.js
1
2 alert("Hello");|
```

```
<body>
  <h1 id="t1">titre</h1>
  <p class="t2">texte1</p>
  <p class="t2">texte 2</p>

  <script src="javascript/app.js"></script>
</body>
```

Terminaison des instructions et commentaires en JavaScript

- Les instructions en JavaScript se terminent par ;

```
alert('hello world');
```

- Les commentaires sont comme le langage C++ et java:
- `/*.....*/` pour un commentaire multilignes
- `//` pour un commentaire sur une seule ligne,

```
<script>
//Je suis un commentaire mono-ligne

/*Je
 suis
 un
 commentaire
 multi-lignes*/

/*On peut utiliser cette syntaxe sur une seule ligne*/
</script>
```


Déclaration des variables en JavaScript

- ✿ Il faut suivre certaines règles de nomenclature des variables dans ce langage:
- ✿ Le nom des variables doit commencer par une lettre ;
- ✿ Le nom ne peut contenir que des lettres (non accentuées), des chiffres ou les signes « _ » et « \$ » ;
- ✿ Le nom des variables est sensible à la casse (« a » est différent de « A ») ;
- ✿ Le JavaScript possède des mots « réservés » qu'on ne peut pas utiliser pour créer une variable. Nous verrons ces mots au fil de ce cours. Par exemple, le mot « var » est un mot réservé.



Déclaration des variables en JavaScript

- 🌻 Déclaration des variables, Voir l'exemple:

```
let toto
```

- 🌻 Les types prédéfinis en JavaScript

```
let toto = 42;      // toto est un nombre  
toto = 'truc';     // toto est désormais une chaîne de caractères  
toto = true;       // toto est désormais un booléen
```

- 🌻 Les chaînes de caractères constantes peuvent être entourées par des guillemets ou apostrophes.

```
let chaine='Bonjour tout le monde';  
let chaine1="Hello World";
```




Déclaration des variables en JavaScript

🌻 Les variables objets:

```
let myVariable = document.querySelector('h1');  
console.log(myVariable);|
```





Déclaration des variables en JavaScript

Les expression régulière:

- 🌻 les expressions régulières permet de fixer un modèle puis tester si les chaines de caractères fournies respectent ce modèle ou non
- 🌻 On utilise les expressions régulières (regex), généralement pour valider la saisie de données des formulaires
- 🌻 Pour déclarer un regex, utiliser cette syntaxe:

```
let pattern =/ regex/identificateur;
```



Déclaration des variables en JavaScript

- L'existence de l'identificateur est optionnelle, vous pouvez le mettre comme vous pouvez l'enlever.

- Exemple d'expression régulière (regex):

```
let pattern =/abc/
```

- Pour tester que ce regex est respecté par la chaîne fournie, voir cet exemple:

```
let pattern =/abc/  
let ch='abc abc dd';  
alert (pattern.test(ch));
```

avec cet exemple: on a déclaré un regex pattern et une chaîne ch puis on a testé si la chaîne respecte le regex en utilisant la fonction test qui retourne un booléen. Dans notre cas le résultat est vrai.



Déclaration des constantes en JavaScript

- 🌻 Déclaration des constantes en JavaScript:

```
const number = 42;
```

- 🌻 Avec cette déclaration vous ne pouvez plus initialiser le contenu number avec une autre valeur, voir l'exemple:

```
const number = 42; |  
number = 50;
```

```
✖ ▶ Uncaught TypeError: Assignment to constant variable.  
    at app.js:3:8
```

Les opérateurs en JavaScript

🌻 Les opérateurs numériques:

Opérateur	Signe
addition	+
soustraction	-
multiplication	*
division	/
modulo	%

🌻 Les opérateurs de concaténation des chaînes

```
var hi = 'Bonjour', name = 'toi', result;  
result = hi + name;  
alert(result); // Affiche : « Bonjourtoi »
```

l'objet console en JavaScript

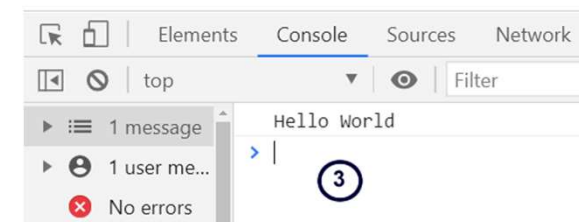
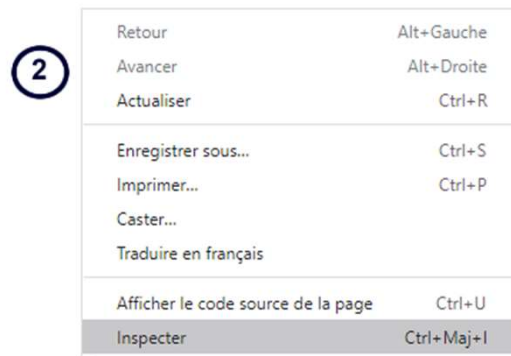
C'est quoi console log ?

🌻 console.log() est une méthode de l'objet console en JavaScript qui est utilisée pour afficher tout type de variables ou bien des messages devant être affichés à l'utilisateur dans la console d'un navigateur ou dans le terminal.

🌻 Voir l'exemple I

```
x='Hello World';  
console.log(x);
```

1





l'objet console en JavaScript

🌻 Voir l'exemple 2:

```
console.log()

console.log() permet
d'afficher des messages
dans la console

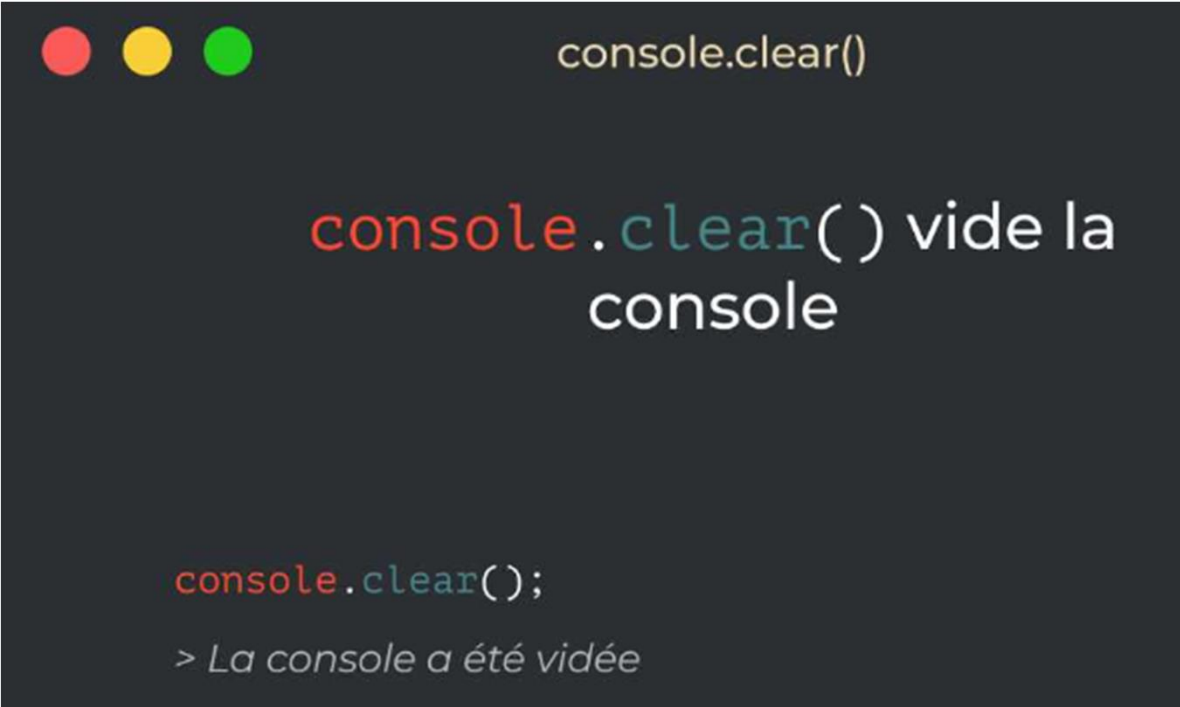
console.log("Hello, World!");
> Hello, World!

console.log("Joyeux anniversaire %s, vous avez
%d ans.", "Alice", 42);
> Joyeux anniversaire Alice, vous avez 42 ans.
```



l'objet console en JavaScript

- ☀ `console.clear()` est une méthode de l'objet console en JavaScript qui est utilisée pour vider le terminal
- ☀ Voir l'exemple;

A dark-themed terminal window with three colored window control buttons (red, yellow, green) in the top-left corner. The text `console.clear()` is shown in a light green monospace font. Below it, the text `console.clear()` is shown in red and blue, followed by the text "vide la console" in white. At the bottom, the command `console.clear();` is shown in red and blue, followed by the output `> La console a été vidée` in a light gray monospace font.

```
console.clear()

console.clear() vide la
console

console.clear();
> La console a été vidée
```



Les tableaux en JavaScript

- ✿ Pour déclarer et créer un tableau, utiliser la syntaxe suivante:

```
let shopping = ['pain', 'lait', 'fromage', 'houmous', 'nouilles'];  
console.log(shopping);
```

- ✿ Les indices des tableaux commence par 0, voir l'exemple d'accès et de modification d'une case du tableau:

```
let shopping = ['pain', 'lait', 'fromage', 'houmous', 'nouilles'];  
console.log(shopping[0]); ➡ Accès  
shopping[2]='chocolat'; ➡ Modification  
console.log(shopping[2]);
```

- ✿ Pour avoir la longueur du tableau, utiliser la propriété length

```
console.log(shopping.length);
```



Les chaines de caractères en JavaScript

- Comme on a dit avant dans ce cours, les chaines de caractères constantes sont délimitées par des guillemets ou des apostrophes.

- Exemple:

```
let str="Hello World";  
let str1='Welcome World';
```

- Pour concaténer des chaines de caractères, on utilise le symbole: +. Exemple:

```
let str="Hello World";  
let str1='Welcome World';  
let res = str+' '+str1;  
console.log(res);
```



Quelques fonctions prédéfinies sur les chaînes de caractères en JavaScript

Rq: Changer le terme **var** pour la déclaration des variables par **let**

🌻 La longueur de la chaîne:

```
var myString = "Ceci est une chaîne de caractères";  
alert(myString.length);
```

🌻 Convertir une chaîne en majuscule ou en minuscule

```
var myString = "Ceci est une chaîne de caractères";  
  
var motEnMinuscules = myString.toLowerCase();  
alert(motEnMinuscules);  
  
var motEnMajuscules = myString.toUpperCase();  
alert(motEnMajuscules); |
```

🌻 Comparer deux chaînes:

```
var myString = "Ceci est une chaîne de caractères";  
var myString1 = "Ceci est une chaîne de caractères";  
  
alert(myString===myString1); |
```




Quelques fonctions prédéfinies sur les chaînes de caractères en JavaScript

🌻 Accéder à un caractère de la chaîne:

l'indice du premier caractère est 0 et l'indice du dernier est `length-1`

```
var myString = "Ceci est une chaîne de caractères";  
alert(myString.charAt(0)); // afficher le premier caractère
```




Les fonctions de conversion en JavaScript

- ✿ Pour convertir une chaîne en un entier, utiliser la fonction

parseInt:

```
var text = '1337', number;  
number = parseInt(text);
```

- ✿ Pour convertir une chaîne en un float, utiliser la fonction

parseFloat

- ✿ Pour convertir des valeurs numériques en une chaînes de caractères, utiliser l'opérateurs de concaténation

```
var text, number1 = 4, number2 = 2;  
text = number1 + '' + number2;  
alert(text); // Affiche : « 42 »
```



Les objets en java Script

- Dans ce cours, nous allons voir la forme la plus simple pour déclarer un objet, voir le graphique ci-dessous:

```
let etudiant={  
  mat: 10,  
  nom: 'ali'  
};  
  
console.log(etudiant.mat);
```

Déclaration de l'objet

Affichage sur la console l'attribut mat d'etudiant

- Vous pouvez créer et manipuler un tableau d'objets, voir le graphique ci-dessous

```
let etudiants=[  
  {  
    mat: 10,  
    nom: 'ali'  
  },  
  {  
    mat: 11,  
    nom: 'mohamed'  
  },  
  {  
    mat: 13,  
    nom: 'maha'  
  }  
];
```

```
console.log(etudiants[0].mat);
```



Les conditions en JavaScript

🌻 Les opérateurs de comparaison:

Opérateur	Signification
==	égal à
!=	différent de
===	contenu et type égal à
!==	contenu ou type différent de
>	supérieur à
>=	supérieur ou égal à
<	inférieur à
<=	inférieur ou égal à

🌻 Les opérateurs logiques:

Opérateur	Type de logique	Utilisation
&&	ET	valeur1 && valeur2
	OU	valeur1 valeur2
!	NON	!valeur



Les conditions en JavaScript

- Les conditions en JavaScript sont les mêmes que ceux du langage C. voir les exemples suivants:

- If else

```
let valeur=Math.floor(Math.random()*10);
if (valeur%2==0) {
    alert(valeur+" est pair ");
} else {
    alert(valeur+" est impair ");
}
```

- Structure if généralisée:

```
let moy=Math.random()*20;
if (moy<10) {
    alert('redouble avec la moyenne '+ moy);
} else if(moy==10){
    alert('réussit passablement avec la moyenne '+moy);
}
else{
    alert('réussit avec mention avec la moyenne '+moy);
}
```



Les conditions en JavaScript

☀ Switch:

```
javascript
1 var drawer = parseInt(prompt('Choisissez le tiroir à ouvrir (1 à 4) :'));
2
3 switch (drawer) {
4   case 1:
5     alert('Contient divers outils pour dessiner : du papier, des crayons, etc.');
```

```
6     break;
7
8     case 2:
9       alert('Contient du matériel informatique : des câbles, des composants, etc.');
```

```
10      break;
11
12      case 3:
13        alert('Ah ? Ce tiroir est fermé à clé ! Dommage !');
```

```
14        break;
15
16        case 4:
17          alert('Contient des vêtements : des chemises, des pantalons, etc.');
```

```
18          break;
19
20          default:
21            alert("Info du jour : le meuble ne contient que 4 tiroirs et, jusqu'à preuve du contraire, les
22            tiroirs négatifs n'existent pas.");
23 }
```

☀ Changer le terme var par le terme let pour déclarer les variables



Les boucles en JavaScript

- Les boucles en JavaScript sont les mêmes que ceux du langage C.
voir les exemples:

- Boucle While:

```
var number = 1;

while (number < 10) {
    number++;
}

alert(number); // Affiche : « 10 »
```

- Boucle for:

```
const array1 = ['a', 'b', 'c'];
for(let i=0;i<array1.length;i++){
    console.log(array1[i]);
}
```




Les boucles en JavaScript

🌻 Boucle do While

```
let val=0;
let somme=0;
let ite=0;
do{
    somme=somme+val;
    val=Math.floor(Math.random()*10);
    ite++;
}while(somme<=20);
console.log('on est arrivé à la somme '+somme+' au bout de '+ite+' itérations');
```



Les boucles en JavaScript

🌻 Boucle for..each

```
let sum = 0;
const numbers = [65, 44, 12, 4];
numbers.forEach(myFunction);

function myFunction(item) {
  sum += item;
  console.log(sum);
}
```

Ecrire et exécuter sa propre fonction en JavaScript

- ✿ Pour écrire une fonction en JavaScript, suivre la syntaxe ci-dessous:

```
function myFunction(arguments) {  
    // Le code que la fonction va devoir exécuter  
}
```

- ✿ Exemple:

```
/** Definition de fonction */  
function double(val){  
    return 2*val;  
}  
  
/**Exécution de fonction */  
let val=Math.round(Math.random()*10);  
console.log('le double de ',val,' est ',double(val));
```

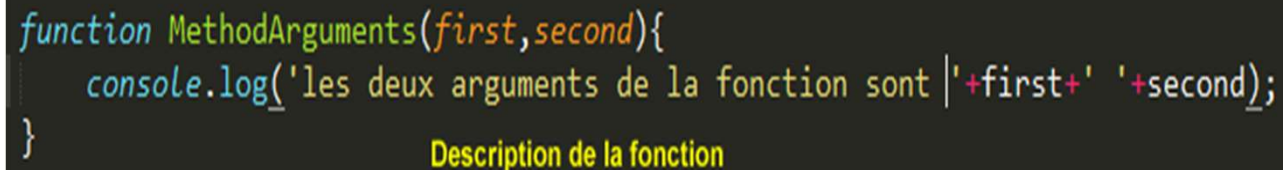
Ecrire et exécuter sa propre fonction en JavaScript

- La description de la fonction **firstMethod()**



```
1  
2  
3 function firstMethod(){  
4   console.log('My first Method');  
5 }  
6
```

- Dans l'exemple ci-dessous, on va voir comment programmer une fonction avec des arguments :



```
function MethodArguments(first, second){  
  console.log('les deux arguments de la fonction sont '+first+ ' '+second);  
}  
Description de la fonction
```

Interaction avec les utilisateurs en JavaScript

- La fonction `prompt` (chaîne de caractères) permet de recevoir une chaîne de caractère. Exemple I:

```
var userName = prompt('Entrez votre prénom :');  
alert(userName); // Affiche le prénom entré par l'utilisateur
```

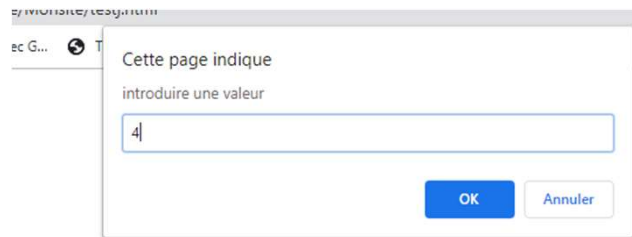


A screenshot of a JavaScript prompt dialog box. The title bar at the top right says 'Git'. The main text area contains the prompt 'Entrez votre prénom :'. Below the text is a single-line text input field with a blue border and a vertical cursor. At the bottom right of the dialog are two buttons: a blue 'OK' button and a grey 'Annuler' button. Above the input field, the text 'Cette page indique' is visible.

- La fonction **alert** (chaîne de caractères) affiche une boîte de dialogue qui informe l'utilisateur,

Interaction avec les utilisateurs en JavaScript

🌻 Exemple2:



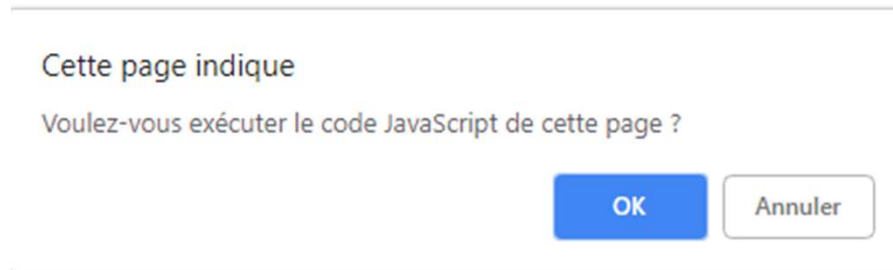
```
var x=prompt('introduire une valeur');  
  
if(isNaN(x))console.log(x+' n est pas numérique');  
else console.log('le double de '+x+' est '+ (parseInt(x)*2));
```

- 🌻 La fonction built-in **isNaN** retourne vrai si la chaîne passée en paramètre n'est pas une valeur numérique,.
- 🌻 La fonction built-in **parseInt** convertit une chaîne en un entier.

Interaction avec les utilisateurs en JavaScript

- La fonction confirm (chaîne de caractères) permet d'informer l'utilisateur et recevoir une réponse du lui qui est TRUE(pour la validation) ou false(pour annulation) Exemple:

```
var response = confirm('Voulez-vous exécuter le code JavaScript de cette page ?');
```



Gestion des événements en JavaScript

- ✿ Avec JavaScript, vous pouvez gérer des événements appliqués sur les balises et les sélecteurs HTML et leurs associer un script, l'exemple ci-dessous vous explique le principe:

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

  <p onmousemove="affiche();" >Texte</p>

                                Lorsque la souris bouge sur le paragraphe, la méthode affiche()
                                s'exécute
</body>
```

```
script.js  x  pagej.html
function affiche(){
    document.write(' la souris bouge sur le paragraphe');
} |
    Description de la fonction affiche
```

Gestion des évènements en JavaScript

🌻 Voici quelques évènements que JavaScript peut les traiter (bien évidemment, il y en a d'autres) :

click (onClick)

Cet évènement est capturé sur un objet quand on clique dessus. Idéal pour les boutons, images, hyperliens, vidéos...

dblClick (onDbClick)

Quand on clique sur un objet deux fois de suite (double clic).

mouseover (onMouseOver)

Quand on survole un objet avec le curseur de la souris.

mouseout (onMouseOut)

Quand on quitte l'objet avec la souris après l'avoir survolé.

focus (onFocus)

Quand on active un élément (quand on place le curseur dans un champ de formulaire, par click ou par tabulation, pour commencer la saisie par exemple).

blur (onBlur)

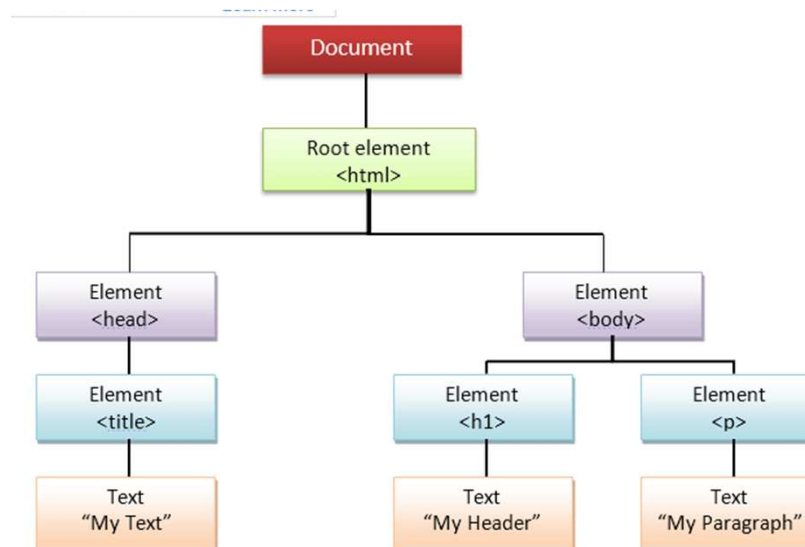
Quand un élément perd le focus (quitter un champ de formulaire après être activé par exemple).

mousemove (onMouseMove)

Quand l'internaute fait bouger le curseur de la souris dans la zone accueillant l'évènement.

Une petite présentation de DOM(Document Object Model)

- DOM est une interface de programmation pour des documents HTML qui représente le document (la page web actuelle) sous une forme d'un arbre qui permet aux langages de script comme le JavaScript d'y accéder et d'en manipuler le contenu et les styles. Exemple du modèle DOM pour une page Web



Accès aux éléments du DOM avec Java Script

- ✿ Ce que vous devez retenir sur le Dom: c'est que La variable **document** représente la balise : html.
- ✿ Pour accéder aux éléments DOM avec javascript, vous pouvez utiliser les méthodes suivantes:

document.getElementsByClassName("ClassName"): elle retourne un tableau où sont stockés tous les éléments de la page portant la classe indiqué dans cette instruction

document.getElementsByTagName(« baliseName ») elle retourne un tableau où sont stockés tous les éléments de la page portant le nom de la balise indiqué dans cette instruction

Accès aux éléments du DOM avec Java Script

`document.getElementById(« IdName »);` : elle retourne une variable qui représente l'élément HTML ayant ce ID

🌻 Exemple

```
<body>
  <div class="example">First</div>
  <div id="t1">second</div>
  <p>third </p>
  <button onclick="myFunction()">test</button>

  <script>
    function myFunction() {
      var x = document.getElementsByClassName("example");// retourne un tableau
      alert(x.length);
      var y =document.getElementsByTagName("p");// retourne un tableau
      alert(y.length);
      var z=document.getElementById("t1"); //retourne une variable
      alert(z);
    }
  </script>
```


Accès aux éléments du DOM avec Java Script

- ☀ Dans notre cours, nous allons modifier les composants de dom suivant deux propriété style (changer le style css) et innerHTML (changer le contenu de l'élément)
- ☀ Voir l'exemple:

```
<body>
  <div class="example">First</div>
  <div id="t1">second</div>
  <p>third </p>
  <button onclick="myFunction()">test</button>

  <script>
    function myFunction() {
      var x = document.getElementsByClassName("example");// retourne un tableau
      x[0].innerHTML = "Hello World!"; //changer le contenu de l'élément

      var y =document.getElementsByTagName("p");// retourne un tableau
      y[0].style.backgroundColor="green"; //changer l'arrière plan de l'élément

      var z=document.getElementById("t1"); //retourne une variable
      z.style.color="red"; //changer la couleur du texte de l'élément
    }
  </script>
</body>
```

Accès aux éléments du DOM avec Java Script

`document.querySelector(" Selector")` : renvoie le premier élément qui correspond à un ou plusieurs sélecteurs HTML spécifiés dans le document. Exemple

```
<script type="text/javascript" src="script.js"></script>
    <form action="essai@essai.rnu" method="post">
        <input type="text" onfocusout ="affiche();" id="my">
    </form>
```

Code HTML

```
function affiche(){
    //var x=document.getElementById("my").value;
    var x =document.querySelector('#my').value;
    alert(x);
    //x.style.backgroundColor = "red";
}
```

Accès aux éléments du DOM avec Java Script

Remarque: La méthode `querySelector ()` renvoie uniquement le premier élément qui correspond aux sélecteurs spécifiés. Pour renvoyer toutes les correspondances, utilisez plutôt la méthode `querySelectorAll ()`. Exemple:

```
<script type="text/javascript" src="script.js"></script>
<h2 class="example">A heading with class="example"</h2>
<p class="example">A paragraph with class="example".</p>
<button onclick="myFunction()">Try it</button>
```

Code en HTML

```
function myFunction() {
  var x, i;
  x = document.querySelectorAll(".example");
  for (i = 0; i < x.length; i++) {
    x[i].style.backgroundColor = "red";
  }
}
```

Code JavaScript

Accès aux éléments du DOM avec Java Script

Exemple:

calcul d'une prime à partir d'un salaire saisie à partir du formulaire:

le code du formulaire

```
<script type="text/javascript" src="script.js"></script>
<form action="res.html" method="post">
  <p>
    <label for="t1">Introduire votre salaire</label>
    <input type="number" id="t1">
  </p>
  <p>
    <input type="submit" value="Valider" onclick="verif();">
    <input type="reset" value="Annuler">
  </p>
</form>
```

Code HTML==> la page du formulaire

Accès aux éléments du DOM avec Java Script

Le code de la fonction verif()

```
function verif(){  
  
    var x=document.querySelector('#t1');  
    if(x.value === '') alert('SVP, introduire le salaire');  
    else{  
  
        if(parseFloat(x.value)<300)alert('SVP, introduire un salaire >300');  
        else{  
            localStorage.setItem('sal',parseFloat(x.value));  
        }  
    }  
  
}
```



Obj4: Identifier le DOM et accéder à ces différents composants en utilisant les évènements et JavaScript

Accès aux éléments du DOM avec Java Script

Le code de la page res.html

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta charset="UTF-8">
</head>
<body onload="calcul();">
<script type="text/javascript" src="script.js"></script>
</body>
</html>
```


La propriété localStorage

localStorage est une propriété de stockage Web qui permet aux sites et applications JavaScript de stocker et d'accéder aux données directement dans le navigateur sans date d'expiration. Cela signifie que les données stockées dans le navigateur persisteront même après la fermeture de la fenêtre.

Dans notre cours, nous allons utiliser les méthodes suivantes qui accompagnent **localStorage** :

- **setItem()**: Ajouter une clé et une valeur à localStorage
- **getItem()**: Récupère une valeur par la clé de localStorage
- **removeItem()**: Supprimer un élément par clé de localStorage
- **clear()**: effacer le contenu du localStorage

La propriété localStorage

Le code de la fonction calcul()

```
function calcul(){  
    if(localStorage.getItem('sal')){  
        var sal=localStorage.getItem('sal');  
        var prime= sal*0.6;  
        document.write('Calcul du salaire <br/>');  
        document.write('le salaire est '+sal+' et la prime est '+prime);  
    }  
    else{  
        document.write('Données erronées <br/>');  
        document.write('<a href="pagej.html"> Accéder au formulaire</a>');  
    }  
    localStorage.clear();  
}
```

La propriété localStorage

Exemple2:

Vérification du mot de passe d'un formulaire avec javaScript:

- Le code du formulaire

```
<script type="text/javascript" src="script2.js"></script>
<form action="pagej1.html" method="post">
  <p>
    <label for="t1">Introduire votre login</label>
    <input type="text" id="t1" onblur="test1();">
  </p>
  <p>
    <label for="t2">Introduire votre mot de passe</label>
    <input type="password" id="t2" onblur="test2();">
  </p>
  <p>
    <input type="submit" value="Valider" onclick="execute();">
    <input type="reset" value="Annuler">
  </p>
</form>
```

Code en html

La propriété localStorage

Exemple I:

Vérification du mot de passe d'un formulaire avec javaScript:

- Le code des fonctions: test1(),test2() et execute()

```
function test1(){
    if((document.querySelector("#t1").value)=== ''){
        alert('SVP, remplir le login');
        document.querySelector("#t1").focus();
    }
}

function test2(){
    if((document.querySelector("#t2").value)=== ''){
        alert('SVP, remplir le mot de passe');
        document.querySelector("#t2").focus();
    }
}

function execute(){
    if(document.querySelector("#t1").value==='essai' && document.querySelector("#t2").value==='123' ){
        localStorage.setItem('login','essai');
        localStorage.setItem('mp','123');
    }
    else alert('SVP, vérifiez votre saisie');
}
```

La propriété localStorage

- Le code de la page page1.html



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title></title>
5
6 </head>
7 <body onload="affiche();">
8   <script type="text/javascript" src="script3.js"></script>
9
10
11
12 </body>
13 </html>
```


La propriété localStorage

- Le code de la page la fonction affiche()

```
function affiche() {  
    if( localStorage.getItem('login') && localStorage.getItem('mp'))  
        document.write ('hello '+localStorage.getItem('login'));  
    else {  
        document.write('mot de passe invalide <br>');  
        document.write('<a href="formj.html"> Accéder au formulaire</a>');  
    }  
    localStorage.clear();  
}
```


La propriété localStorage

- Voir la vidéo sur youtube:

<https://www.youtube.com/watch?v=CwmweEu5LI0>



Merci pour votre attention

Merci pour votre attention

Ens: MANSOUR Sihem