

Chapitre IV : JavaScript

Plan

- Introduction
- Éléments de base de JavaScript
- Objets prédéfinis
- Les événements

Introduction

présentation

JavaScript

- Un langage de programmation **interprété**, c'est-à-dire qu'il a besoin d'un interpréteur pour pouvoir être exécuté.
- Permet de rendre **dynamique** un site internet développé en HTML.
- **Basé sur les objet, pas de classes ni d'héritage,...**
- Déclaration optionnelle des variables (**typage faible**).
- Une syntaxe proche des langages de programmation comme le C.

Introduction

Le Script JavaScript

Un **script** est une portion de code qui vient s'insérer dans une page HTML.
Le code JavaScript doit être placé toujours à l'intérieur de **script**.

<SCRIPT>

Placez ici le code de votre script.

</SCRIPT>

Introduction

Exemple

- Un petit script contenant la fonction **alert()** qui permet d'afficher une boîte de dialogue contenant le message passé en paramètre.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> Mon premier programme en JS </title>
  </head>

  <body>
    <script>
      alert("je suis une alerte en JS");
    </script>
  </body>
</html>
```

Cette page indique

je suis une alerte en JS

OK

Éléments de base de JavaScript

Variables

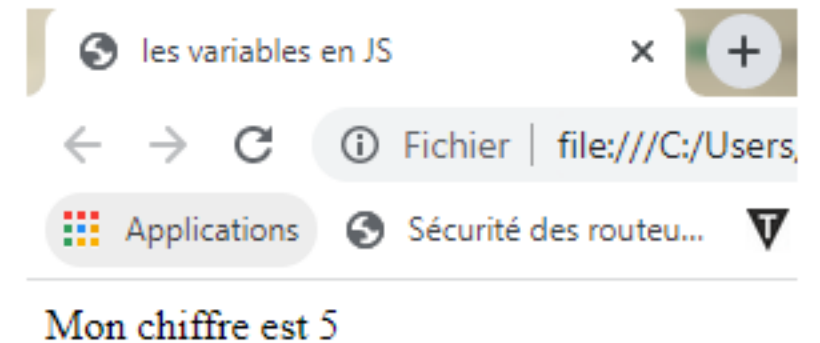
- Les variables contiennent des données qui peuvent être modifiées lors de l'exécution d'un programme.
- Chaque variable possède un nom et une valeur, et doit être déclarée
 - soit de façon explicite au moyen du mot réservé **var**.
 - soit de façon implicite : **nom_de_la_variable = valeur ;**
- Les types de valeurs JavaScript
 - **Nombres** : N'importe quel nombre sans guillemet,
 - **Chaînes de caractères**: Une série de caractères à l'intérieur des guillemets
 - **Booléens** : true ou false,
 - **Null** : exempt de toute valeur,
 - **Objets** : sont définis par leurs propriétés et méthodes,
 - **Fonctions**.

Éléments de base de JavaScript

Variables

- Exemple

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> les variables en JS </title>
  </head>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript">
      var chaine = "Mon chiffre est ";
      var variable = 5;
      document.write(chaine + variable);
    </SCRIPT>
  </BODY>
</html>
```



Éléments de base de JavaScript

les conditions

- Condition simple :

```
if (expression_booléenne) {  
    // Instructions  
}
```

- Condition avec la variante **else** :

```
if (expression_booléenne) {  
    // Instructions  
}  
else {  
    // Instructions  
}
```


Éléments de base de JavaScript

les conditions

- Exemple

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> les conditions en JS </title>
  </head>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript">
      if (confirm('Voulez-vous exécuter le code Javascript de cette page ?')) {
        alert('Le code a bien été exécuté !');
      }
    </SCRIPT>
  </BODY>
</html>
```

Cette page indique

Voulez-vous exécuter le code Javascript de cette page ?

OK

Annuler

Cette page indique

Le code a bien été exécuté !

OK

Éléments de base de JavaScript

les conditions

- Switch

```
switch (expression) {  
    case etiquette_1 : // Instructions  
        break;  
    case etiquette_2 : // Instructions  
        break;  
    .....  
    case etiquette_n : // Instructions  
        break;  
    default : // Instructions  
}
```

Éléments de base de JavaScript

les boucles

- La boucle **while** :

```
while (expression_booléenne)  
{  
    // Instructions  
}
```

- La boucle **do ... while** :

```
do  
{  
    // Instructions  
} while (expression_booléenne)
```

Éléments de base de JavaScript

les boucles

- La boucle **for**:

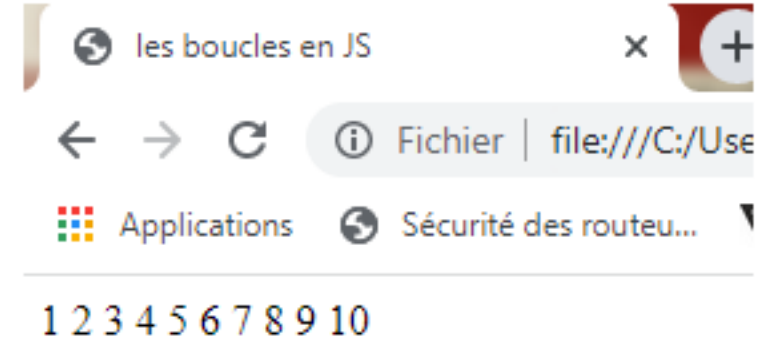
```
for (initialisation; condition; incrémentation) {  
    instruction_1;  
    instruction_2;  
    instruction_3;  
}
```

Éléments de base de JavaScript

les boucles

- Exemple (while)

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> les boucles en JS </title>
  </head>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript">
      var n = 1;
      var m = "affichage des 10 premiers nombres";
      while (n <= 10)
      {
        document.write(n+' ');
        n++;
      }
    </SCRIPT>
  </BODY>
</html>
```



Éléments de base de JavaScript

les fonctions

- En JavaScript, une fonction aura la forme suivante :

```
function nom_de_la_fonction(para_1, ..., para_n)
{
    // corps de la fonction
}
```

- Une fonction peut retourner une valeur ou non (une simple procédure).
- L'instruction **return** permet d'interrompre une fonction en envoyant la valeur de celle-ci au système.
- Si **return** est omise, la valeur retournée est par défaut « undefined ».

Éléments de base de JavaScript

les fonctions

- Exemple

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> les fonctions en JS </title>
  </head>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript">
      function Bonjour(nom,prenom,age)
      {
        document.write("votre nom est "+nom+"<br>");
        document.write("votre prénom est "+prenom+"<br>");
        document.write("votre âge est ",+age+"<br>");
      }
      nom=prompt("Entrez votre nom ");
      prenom=prompt("Entrez votre prénom");
      age=Number(prompt("Entrer votre âge "));
      document.write("bonjour <br>");
      Bonjour(nom,prenom,age);
    </SCRIPT>
  </BODY>
</html>
```

Cette page indique

Entrez votre prénom

Mohammed

OK

Annuler

Cette page indique

Entrez votre âge

23

OK

Annuler

← → ↻ ⓘ Fichier | fil

🗄 Applications 🔒 Sécurité des

bonjour
votre nom est Rahmoun
votre prénom est Mohammed
votre âge est 23

Éléments de base de JavaScript

Orienté Objet

- un style de POO qui n'utilise pas les classes.
- Ce langage ne dispose **pas d'une instruction pour déclarer une classe** (à la différence de C++ ou Java).
- Il utilise des **fonctions** comme **constructeurs** pour définir un objet. On définit les **propriétés** et **méthodes** d'un **objet** en définissant une fonction qui sera utilisée par la suite pour construire l'objet souhaité.
- Pour définir une **méthode**, on peut l'ajouter à l'intérieur du constructeur. Le nom auquel est assigné la fonction est le nom de la méthode.
- Syntaxe :

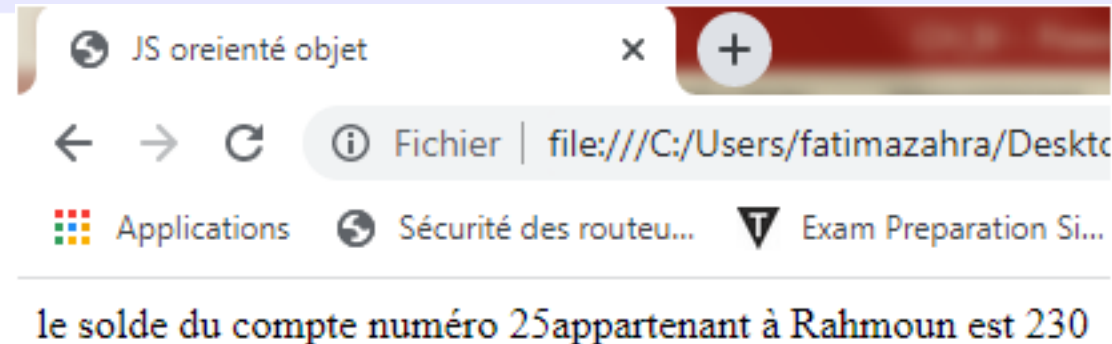
```
this.method_name = function(parametre){  
    // contenu...  
};
```


Éléments de base de JavaScript

Orienté Objet

- Exemple : constructeur CompteBancaire.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> JS oreienté objet </title>
  </head>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript">
      function CompteBancaire(num, solde, titulaire){
        this.num = num;
        this.solde = solde;
        this.titulaire = titulaire;
      }
      var cb1 = new CompteBancaire(25, 230, 'Rahmoun');
      document.write('le solde du compte numéro '+cb1.num+ ' appartenant à ' +cb1.titulaire+' est '+cb1.solde);
    </SCRIPT>
  </BODY>
</html>
```



Éléments de base de JavaScript

Orienté Objet

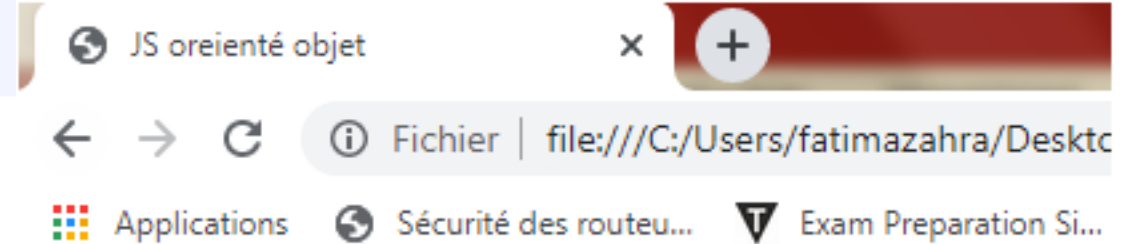
- Exemple : constructeur CompteBancaire
 - Ajouter une méthode

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> JS oreienté objet </title>
  </head>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript">
      function CompteBancaire(num, solde, titulaire){
        this.num = num;
        this.solde = solde;
        this.titulaire = titulaire;

        this.Crediter = function(montant) {
          this.solde += montant;
        }
      }

      var cb1 = new CompteBancaire(25, 230, 'Rahmoun');
      document.write('le solde du compte numéro '+cb1.num+ ' appartenant à ' +cb1.titulaire+' est '+cb1.solde)
      cb1.Crediter(20)
      document.write('<br> le solde après est '+cb1.solde);

    </SCRIPT>
  </BODY>
</html>
```



le solde du compte numéro 25 appartenant à Rahmoun est 230
le solde après est 250

Objets prédéfinis

Date

- Cet objet permet de travailler avec des **dates**.
- Quelques **méthodes** de l'objet Date :
 - getDate() : Retourne le jour du mois.
 - getDay() : Retourne le jour de la semaine.
 - getMonth() : Retourne le numéro du mois.
 - getFullYear() : Retourne l'année.
- Pour créer un objet de type Date il faut écrire :
var nom_objet = new Date()
- Une fois un objet Date construit, on peut l'utiliser par les nombreuses méthodes qu'il possède.

Objets predefinis

Date

- Exemple

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> JS oreienté objet </title>
  </head>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript">
      var maDate=new Date();
      var jour = maDate.getDate();
      var mois = maDate.getMonth()+1;
      var annee = maDate.getFullYear();
      document.write("<small><font><b>"+jour+" / "+mois+" / "+annee+" </b></font></small>");
    </SCRIPT>
  </BODY>
</html>
```

← → ↻ ⓘ Fichi

📱 Applications 🔒 Sécu

25 / 3 / 2023

Objets predefinis

String

- Cet objet permet de travailler avec des chaînes de texte et de les manipuler. Pour créer un objet de type String il faut écrire :

var nom_objet = **new String**("chaine");

- Une écriture du style suivant est tolérée :

var nom_objet = "chaine";

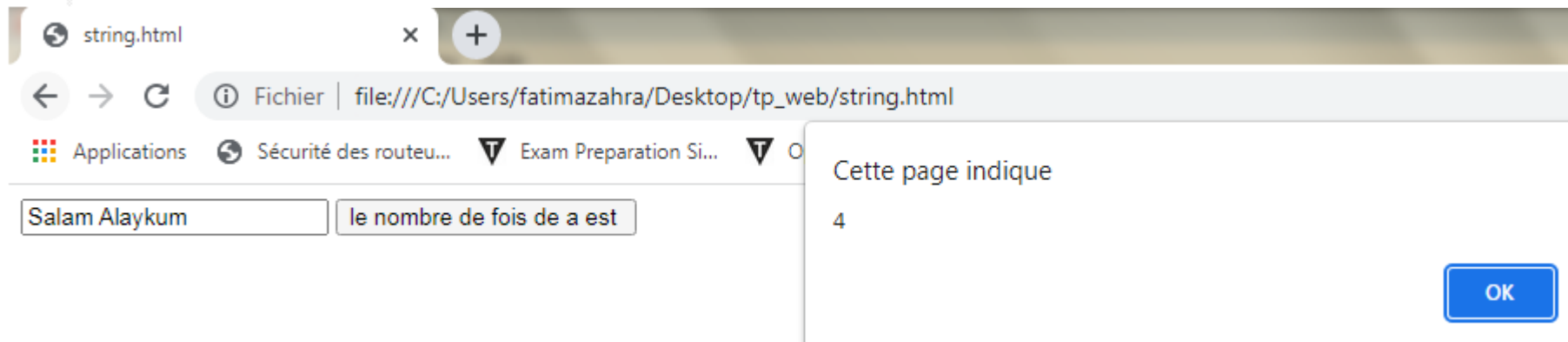
- La propriété **length** : la longueur d'une chaîne.
- La méthode **toLowerCase()** : convertit une chaîne en minuscule.
- La méthode **toUpperCase()** : convertit une chaîne en majuscule.
- La méthode **CharAt()** : extrait un caractère simple à une position connue à l'intérieur d'une chaîne.
- ...

Objets predefinis

String

- Exemple

```
<html>
  <head>
  </head>
  <body>
    <script language="javascript">
      function nombre_fois(form){
        var k=0;
        var input_chaine=form.input.value.toLowerCase();
        for(var i=0;i<input_chaine.length;i++){
          if(input_chaine.charAt(i)=="a")
            k+=1;
        }
        alert(k);
      }
    </script>
    <form name="form">
      <input type="text" name="input" value="">
      <input type="button" value="le nombre de fois de a est " onclick="nombre_fois(form)">
    </form>
  </body>
</html>
```



Objets predefinis

Math

- L'objet **Math** permet de manipuler les nombres et d'accéder aux fonctions mathématiques classiques.
- **Fonctions classiques** : `Math.abs()`, `Math.cos()`, `Math.sin()`, `Math.tan()`, `Math.exp()`, `Math.log()`, `Math.sqrt()`.
- **Math.max()** : retourne le **maximum** parmi deux nombres.
- **Math.min()** : retourne le **minimum** parmi deux nombres.
- **Math.pow(x,y)** : retourne x à la **puissance** y.
- **Math.random()** : retourne un **nombre aléatoire** entre 0 et 1.

Les événements

- Les événements sont des structures de code qui « **écoutent** » ce qui se passe dans le navigateur et qui permettent de **déclencher** des **actions** lorsque quelque chose de particulier se passe. Ils sont utilisés pour permettre l'interactivité d'une page web.
- Grâce au JavaScript, il est possible **d'associer** des **fonctions** et des **méthodes** à des **événements** tels que le passage de la **souris au-dessus d'une zone**, le **changement d'une valeur**, ...
- Généralement, les événements JavaScript peuvent être appliqués sur n'importe quel type d'éléments html. Pour cibler un élément on peut utiliser la méthode **getElementById()** de l'objet **document**. Cette méthode retourne une référence à l'objet correspondant à l'identifiant donné.

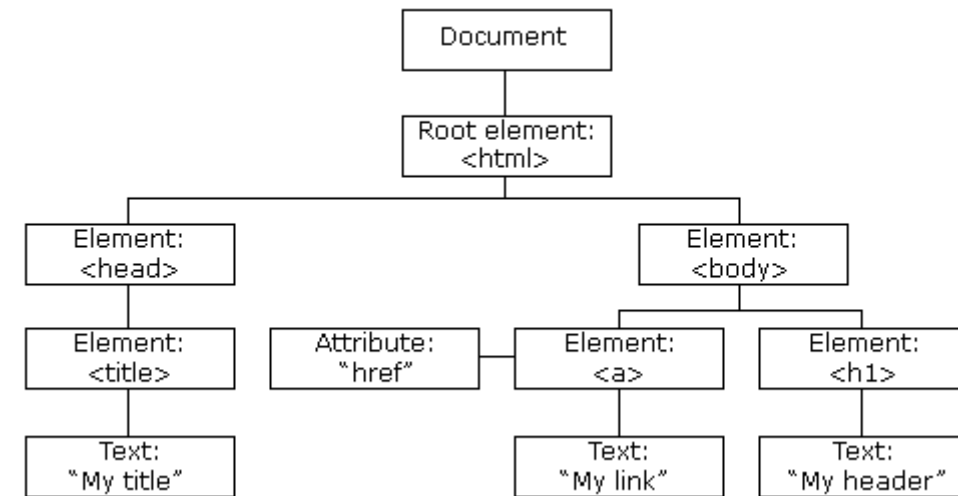
Les événements

Objet document

- L'objet **document** sert de **point d'entrée** au contenu d'une page web et apporte des fonctions générales au document.
- L'objet **document** de JavaScript utilise le Document Object Model (**DOM**), pour accéder à la page Web et ses éléments.
- Le **DOM** est une interface pour les documents HTML et XML. Il fournit une **représentation structurelle du document**, permettant de modifier son contenu et sa présentation visuelle. Fondamentalement, il **relie** les pages Web aux scripts et langages de programmation.

Le DOM HTML

- Lorsqu'une page Web est chargée, le **navigateur** crée un **DOM** (Document Object Model) pour cette page.
- Avec le DOM, JavaScript obtient toute la puissance dont il a besoin pour créer du **HTML dynamique** :
 - **Modifier** tous les éléments et attributs HTML
 - **Modifier** tous les styles CSS de la page
 - **Supprimer** des éléments HTML existants
 - **Ajouter** de nouveaux éléments et attributs HTML
 - **Réagir** à tous les événements HTML existants dans la page
 - **Créer** de nouveaux événements HTML dans la page

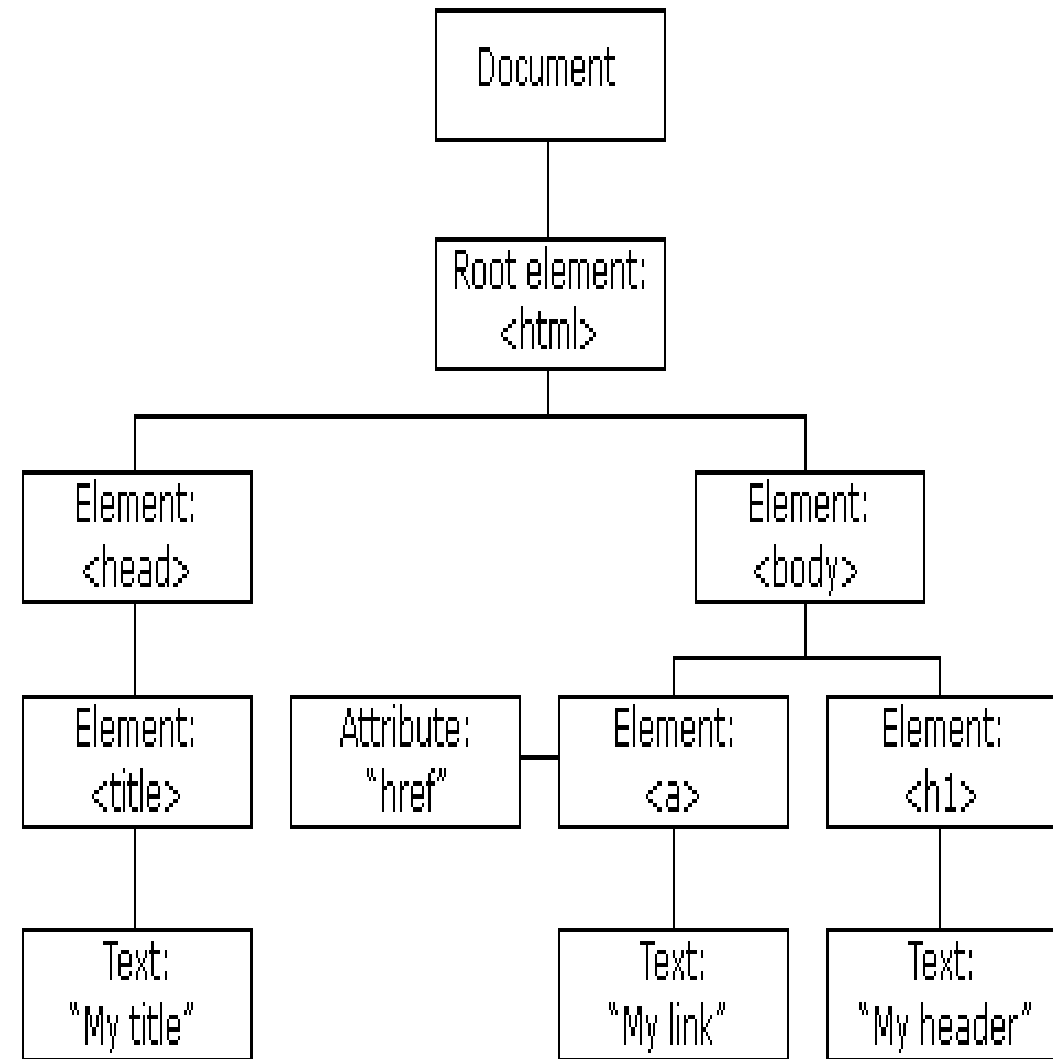


Le DOM HTML

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> My title </title>
  </head>
  <body>
    <a href="www.google.com">My link</a>
    <h1>My header</h1>
  </body>
</html>
```

Le code HTML

A.BENGAG



Le DOM HTML

Les événements

Objet document

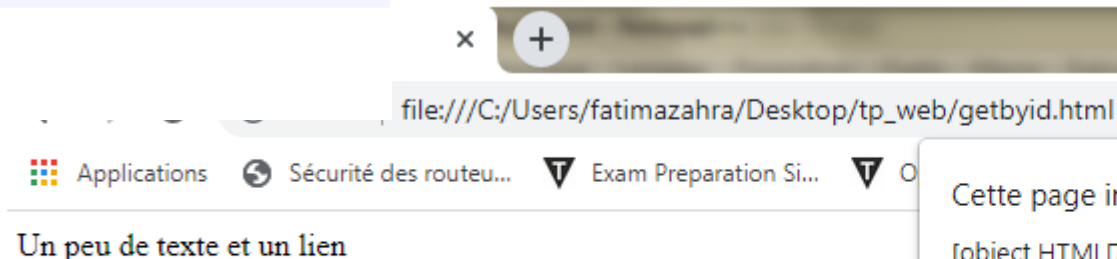
- Méthodes :
 - `document.getElementById()`: renvoie une référence à l'objet correspondant à l'identifiant donné.
 - `document.createElement`: crée un nouvel élément du type spécifié
 - `document.createTextNode`: crée un nœud de texte.
 - `document.write`: écrit du texte dans le document.
- Propriétés :
 - `document.images`: renvoie une liste des images du document.
 - `document.location`: renvoie l'URI du document courant.
 - ...

Les événements

Objet document

- Accéder aux éléments HTML par l'attribut ID
- *getElementById()* Cette méthode permet d'accéder à un élément en connaissant son ID qui est simplement l'attribut id de l'élément.
- Exemple

```
<html>
<head>
</head>
<body>
  <div id="myDiv">
    <p>Un peu de texte <a>et un lien</a></p>
  </div>
  <script>
    var div = document.getElementById('myDiv');
    alert(div);
  </script>
</body>
</html>
```



Les événements

Objet document

- La propriété **innerHTML** : permet de récupérer ou de définir le code HTML présent à l'intérieur d'un élément
- Exemple

```
<html>
  <head>
  </head>
  <body>
    <div id="myDiv">
      <p>Un peu de texte <a>et un lien</a></p>
    </div>
    <script>
      var div = document.getElementById('myDiv').innerHTML;
      alert(div);
    </script>
  </body>
</html>
```

Cette page indique

→ → → <p>Un peu de texte <a>et un lien </p>

→ →

Les événements

Objet document

- La propriété **innerText**: récupérer du texte brut, sans aucunes balises HTML
- Exemple

```
<html>
  <head>
  </head>
  <body>
    <div id="myDiv">
      <p>Un peu de texte <a>et un lien</a></p>
    </div>
    <script>
      var div = document.getElementById('myDiv').innerText;
      alert(div);
    </script>
  </body>
</html>
```

Cette page indique

Un peu de texte et un lien

OK

Les événements

appliquer un événement

- Pour appliquer un événement à un élément html:
 - **1^{ère} méthode:** associer le type **d'événement** (onclick, onload, ...) à **l'élément** en le spécifiant comme **attribut**.
 - Exemple : `<p onclick="alert('Hello !');"> Cliquez-moi ! </p>`
 - **2^{ème} méthode:** cibler **l'élément** grâce à la méthode `getElementById('id_element')`:
 - **1^{ère} solution:** appliquer la **méthode** événement directement sur **l'élément** HTML.
`element.onEvenement()="Fonction();"`
 - **2^{ème} solution:** appliquer un **évènement** sur un **élément** HTML en utilisant la méthode `'setAttribute()'`.
`element.setAttribute('onEvenement','nom_fonction('args',event));`
 - **3^{ème} solution:** utiliser la méthode `'addEventListener'`
`element.addEventListener('Event_type', function(){ });`

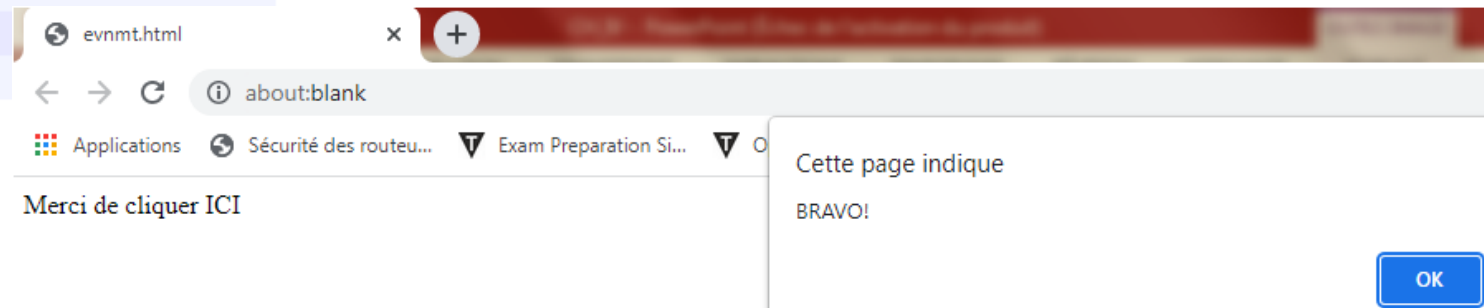
Les événements

appliquer un événement

- La méthode 'addEventListener' elle prend trois paramètres (on a spécifié que deux)
 - Le nom de l'événement;
 - La fonction à exécuter ;

- Exemple

```
<html>
  <head>
  </head>
  <body>
    <div id="myDiv">
      <p id="clic"> Merci de cliquer ICI </p>
    </div>
    <script>
      var element = document.getElementById('clic');
      element.addEventListener("click", function(){
        alert("BRAVO!");
      });
    </script>
  </body>
</html>
```



Les événements

appliquer un événement

- La méthode 'addEventListener':

- Liste d'événements

Nom de l'événement	Action pour le déclencher
click	Cliquer (appuyer puis relâcher) sur l'élément
dblclick	Double-cliquer sur l'élément
load	Cet évènement se déclenche lors de chargement du document
mouseover	Faire entrer le curseur sur l'élément
mouseout	Faire sortir le curseur de l'élément
mousedown	Appuyer (sans relâcher) sur le bouton gauche de la souris sur l'élément
mouseup	Relâcher le bouton gauche de la souris sur l'élément
mousemove	Faire déplacer le curseur sur l'élément
keydown	Appuyer (sans relâcher) sur une touche de clavier sur l'élément
keyup	Relâcher une touche de clavier sur l'élément
keypress	Frapper (appuyer puis relâcher) une touche de clavier sur l'élément
focus	« Cibler » l'élément
blur	Annuler le « ciblage » de l'élément
change	Changer la valeur d'un élément spécifique aux formulaires