ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
Membre de HONORIS UNITED UNIVERSITIES

Cours d'Initiation à la Programmation Web

Langage de Script JavaScript

Partie 1 : Introduction

Assuré par:

Mme. RAZZOUQI MAROUA

Année universitaire: 2022/2023

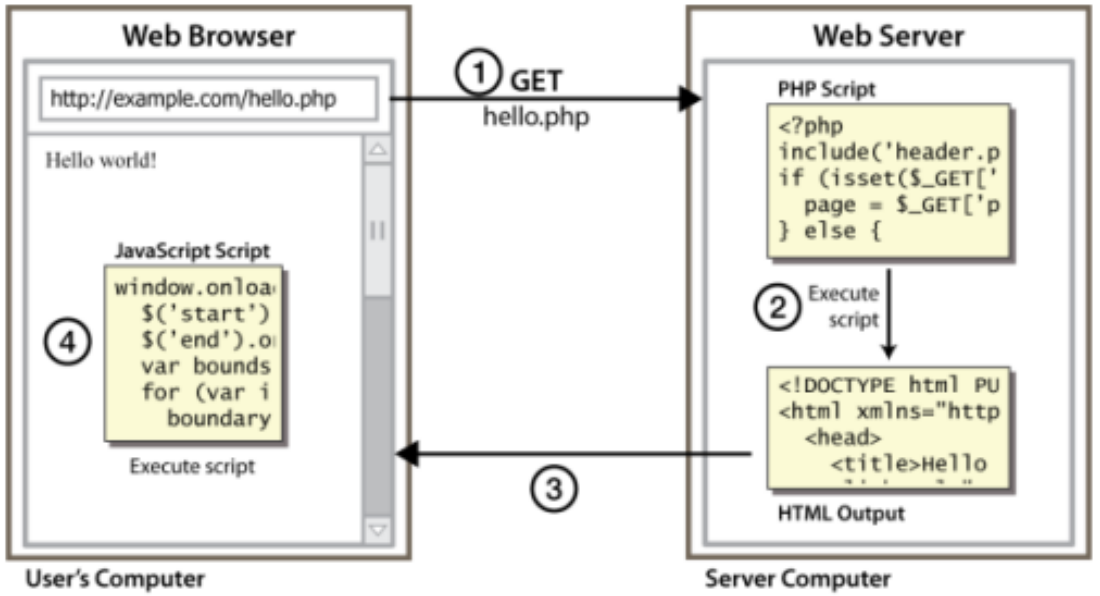
Plan du cours

1. **Introduction au langage JavaScript**
   - Programmation côté client
   - La JavaScript (JS)
   - Variables et typres
   - Les valeurs spéciales null et undefined
   - Les commentaires
   - Les boucles
   - Opérateurs logiques
   - La forme if/else
   - Les tableaux
   - Manipulation de chaînes
   - Les fonctions
   - Programmation par Events

2. **Introduction au Document Object Model**
   - La fonction document.getElementById
   - L'arbre HTML
   - Le Document Object Model (DOM)
   - Obtenir les objets DOM en JS
   - Contenu d'un objet DOM
   - Accéder aux propriétés d'un objet DOM
   - La propriété innerHTML
   - Utilisation des objets DOM
   - Modifier un objet DOM

Introduction au langage JavaScript

Terminologie : Programmation côté client



Programmation côté client : le code s'exécute dans le navigateur après avoir été envoyé (avec le HTML) par le serveur. La plupart du temps, ce code manipule l'arbre HTML en fonction d'évènements déclenchés par l'utilisateur (comme par exemple un clic).

Introduction au langage JavaScript

Le JavaScript (JS)

Un langage de script léger ("scripting language"), créé in 1995 par Brendan Elch (le nom initial était LiveScript) afin de rendre les pages HTML plus interactives :

- Insertion dynamique de code HTML
- Réaction à des évènements (chargement de page, clics, etc.)
- Lecture d'informations relatives au navigateur
- Calculs (par exemple validation de formulaire)

C'est **un standard du web** (mais pas supporté de la même manière par tous les **navigateurs**).

Il peut être utilisé dans un navigateur mais aussi dans Adobe Acrobat, Adobe Photoshop, des systèmes embarqués, le terminal,...

Introduction au langage JavaScript

La balise <script>

HTML (modèle)

```
<script src="filename" type="text/javascript" ></script>
```

HTML (exemple)

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

La balise ***script*** doit être placée dans l'élément ***head*** de la page ***HTML*** . Le code JS se trouve normalement dans un ***fichier.js*** séparé.

Il peut être placé dans le corps du document ***HTML*** (comme pour ***CSS***) :

- **Mauvais style de programmation** : on cherche à séparer le contenu (HTML), la présentation (CSS) et le comportement (JS)

Introduction au langage JavaScript

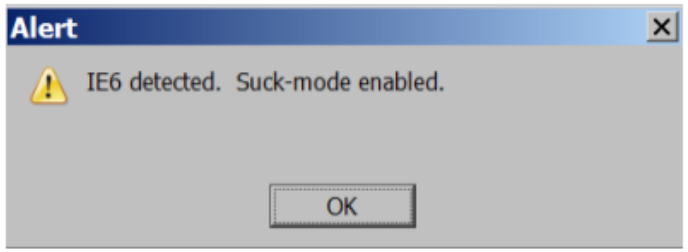
Notre première fonction JS : alert

HTML (modèle)

```
alert("message");
```

HTML (exemple)

```
alert("IE6 detected. Suck-mode enabled");
```



alert() : Une commande JS qui fait apparaitre une boîte de dialogue avec un message.

Introduction au langage JavaScript

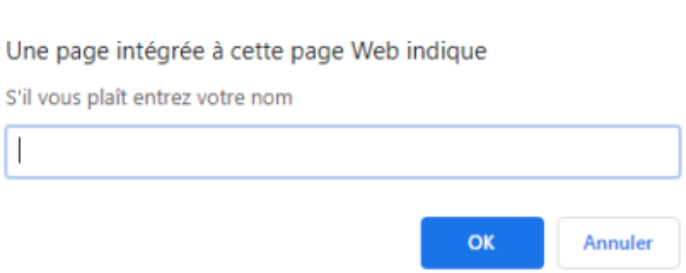
Notre deuxième fonction JS : prompt

HTML (modèle)

```
prompt("message");
```

HTML (exemple)

```
prompt("S'il vous plaît entrez votre nom");
```



prompt() : Une commande JS qui affiche une boîte de dialogue, éventuellement avec un message, qui invite l'utilisateur à saisir un texte.

Introduction au langage JavaScript

Variables et types

JS (modèle)

```
let name = expression ;
```

JS (exemple)

```
let level = 23 ;
let accuracyRate = 0.99 ;
let name = "Pikachu" ;
```

Les variables sont déclarées avec le mot-clef ***let*** ou ***var*** . JS possède des types de données (langage faiblement typé).

- ***Number*** , ***Boolean*** , ***String*** , ***Array*** , ***Object*** , ***Null*** , ***Undefined***
- La fonction **typeof()** retourne le type d'une valeur.
- Conversion de type possible et **parfois surprenante**...

Introduction au langage JavaScript

Le type Number

JS (exemple)

```
let enrollment = 99;
let mediangrade = 2.8;
let credits = 5 + 4 + (2 * 3);
```

Les nombres regroupent les entiers et les réels (pas de ***int*** et de ***double***).

Les mêmes operateurs : ***+*** , ***-*** , ******* , ***/*** , ***%*** , ***++*** , ***- -*** , ***=*** , ***+=*** , ***-=*** , ****=*** , ***/=*** , ***%=***

Même précédence que les autres langages de programmation.

Beaucoup de conversions implicites : ***"2" * 3*** donne ***6*** .

Introduction au langage JavaScript

Le type String

JS (exemple)

```
let nickName = "Sparky O'Sparkz" ;          // "Sparky O'Sparks"
let fName = nickName.substring(0, nickName.indexOf(" ")) ; //
"Sparky"
let len = nickName.length ;                 // 15
let name = 'Pikachu' ;                      // can use " " or ' '
```

Methods : **charAt**, **charCodeAt**, **fromCharCode**, **indexOf**, **lastIndexOf**, **replace**, **splice**, **substring**, **toLowerCase**, **toUpperCase**

- ***charAt*** retourne une String d'une lettre (pas de type char).
- ***length*** est une propriété (et non une méthode comme en Java).
- La concaténation des chaînes s'effectue avec l'opérateur ***+*** : ***1 + 1*** vaut ***2*** , mais ***"1" + 1*** donne *"11"* .

Introduction au langage JavaScript

Le type String

Conversions entre nombres et Strings :

JS (exemple)

```
let count = 10;                                  // 10
let stringedCount = "" + count;                  //"10"
let puppyCount = count + " puppies, yay!"; // "10 puppies, yay!"
let magicNum = parseInt("42 is the answer"); // 42
let mystery = parseFloat("Am I a number?"); // NaN
```

On peut aussi faire la conversion en utilisant : ***Number("45.3")***

On peut vérifier si une variable est numérique en utilisant : ***isNaN(....)***

Introduction au langage JavaScript

Le type String

Pour accéder aux caractères d'une String ***s***, on peut utiliser ***s[index***] ou ***s.charAt(index***) :

JS (exemple)

```
let firstLetter = puppyCount[0] ;                          // "1"
let fourthLetter = puppyCount.charAt(3) ;                  // "p"
let lastLetter = puppyCount.charAt(puppyCount.length - 1) ; //"!"
```

Introduction au langage JavaScript

Le type Booléen

JS (exemple)

```
let iLikeJS = true;
let ieIsGood = "IE6" > 0;                    // false
if ("web dev is great") {                    /* true */ }
if (0) {                                     /* false */ }
```

Toute valeur peut être utilisée comme un booléen :

- Valeurs fausses : ***false*** , ***0*** , ***0.0*** , ***NaN*** , ***""*** , ***null*** et ***undefined***
- Valeurs vraies : ***true*** et toutes les autres !

Conversion explicite d'une valeur en booléen :

- ***let boolValue = Boolean(otherValue)*** ;
- ***let boolValue = ! !(otherValue***) ;

Introduction au langage JavaScript

Les valeurs spéciales null et undefined

JS (exemple)

```
let foo = null;
let bar = 9;
let baz;

/*
* foo vaut null
* bar vaut 9
* baz vaut undefined
*/
```

- ***undefined*** : la variable n'a pas été déclarée, ou n'a pas été initialisée.
- ***null*** : la variable existe et a reçu la valeur spéciale null.

Introduction au langage JavaScript

L'objet Math

JS (exemple)

```
let rand1to10 = Math.floor(Math.random() * 10 + 1);
let three = Math.floor(Math.PI);
```

Methods : abs, ceil, floor, log, max, min, pow, sqrt, random, round, sin, cos, tan.

Propriétées : ***E*** , ***PI*** .

Introduction au langage JavaScript

Les commentaires

JS (exemple)

```
// commentaire sur une ligne

/* commentaire multi-line */
```

Identique à Java, C,...

Trois syntaxes :

- HTML : ***< !- - comment - ->***
- CSS/Java/PHP/JS ***: /* comment */***
- Java/PHP/JS : ***// comment***

Introduction au langage JavaScript

La boucle for

JS (modèle)

```
for (initialization ; condition ; update) {
   statements ;
}
```

JS (exemple)

```
let sum = 0 ;
for (let i = 0 ; i < 100 ; i++) {
   sum = sum + i ;                    // identique à sum += i ;
}
```

Introduction au langage JavaScript

La boucle for

JS (exemple)

```
let s1 = "It's a-me, Mario !" ;
let s2 = "" ;

for (let i = 0 ; i < s1.length ; i++) {
s2 += s1[i] + s1[i] ;
}

                    // s2 est égale à "IItt''ss aa- -mmee,, MMaarriioo ! !"
```

Introduction au langage JavaScript

La boucle while

JS (Modèle 1)

```
while (condition) {
   statements ;
}
```

JS (modèle 2)

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

Les instructions ***break*** et ***continue*** existent et fonctionnent comme en Java, C, etc...

Introduction au langage JavaScript

Opérateurs logiques

Relationnels : ***>*** , ***<*** , ***>=*** , ***<=***

Logiques : ***&&*** , ***||*** , ***!***

Egalité : ***==*** , ***!=*** , ***===*** , ***!==***

1. La plupart des opérateurs logiques convertissent implicitement.

   Par exemple, toutes ces expressions sont vraies :

   - ***5 < "7"***
   - ***42 == 42.0***
   - ***"5.0" == 5***

2. Les opérateurs ***===*** et ***!==*** sont les test d'égalité stricts :

   la valeur et le type doivent être égaux. Par exemple
   "5.0" === 5 vaut ***false***

Introduction au langage JavaScript

La forme if/else

JS (Modèle)

```
if (condition) {
   statements ;
}else if (condition) {
   statements ;
} else {
   statements ;
}
```

- Même structure qu'aux autres langage de programmation (Java, C,...).
- JS accepte presque n'importe quelle valeur pour la condition.

Introduction au langage JavaScript

Les tableaux

JS (Modèle)

```
let name = [];                          // tableau vide
let names = [value1, ... , valueN];     // tableau avec contenu initial
names[index] = value;                   // pour stocker une valeur
```

JS (Exemple)

```
let types = ["Electric", "Water", "Fire"];
let pokemon = [];              // []
pokemon[0] = "Pikachu";  // ["Pikachu"]
pokemon[1] = "Squirtle";  // ["Pikachu", "Sqiurtle"]
pokemon[3] = "Magikarp"; // ["Pikachu", "Sqiurtle", undefined,
"Magikarp"]
```

La propriété ***length*** donne la taille (dynamique)

Introduction au langage JavaScript

Les fonctions sur les tableaux

JS (Exemple)

```
let a = ["Mario", "Luigi"]; // [Mario, Luigi]
a.push("Koopatroopa");      // [Mario, Luigi, Koopatroopa]
a.unshift("Bowser");        // [Bowser, Mario, Luigi, Koopatroopa]
a.pop();                    // [Bowser, Mario, Luigi]
a.shift();                  // [Mario, Luigi]
a.sort();                   // [Luigi, Mario]
```

Les tableaux remplacent les structures de données : liste, pile, file, ...

Méthodes : **concat**, **join**, **pop**, **push**, **reverse**, **shift**, **slice**, **sort**, **split**, **toString**, **unshift**.

- ***push*** et ***pop*** ajoute/supprime par la fin.
- ***shift*** et ***unshift*** ajoute/supprime par le début.
- ***shift*** et ***pop*** retournent l'élément supprimé

Introduction au langage JavaScript

Manipulation de chaînes : split et join

JS (Exemple)

```
let a = ["Mario", "Luigi"]; // [Mario, Luigi]
a.push("Koopatroopa");      // [Mario, Luigi, Koopatroopa]
a.unshift("Bowser");        // [Bowser, Mario, Luigi, Koopatroopa]
a.pop();                    // [Bowser, Mario, Luigi]
a.shift();                  // [Mario, Luigi]
a.sort();                   // [Luigi, Mario]
```

- ***split*** éclate une ***String*** en un tableau par rapport à un délimiteur

  - Le délimiteur peut être une expression régulière : ***let a = s.split(/[\t]+/)*** ;

- ***join*** fusionne le contenu d'un tableau en une ***String*** et en ajoutant un délimiteur entre les éléments.

Introduction au langage JavaScript

Définition de fonctions

JS (Modèle)

```
function name() {
statement ;
...
}
```

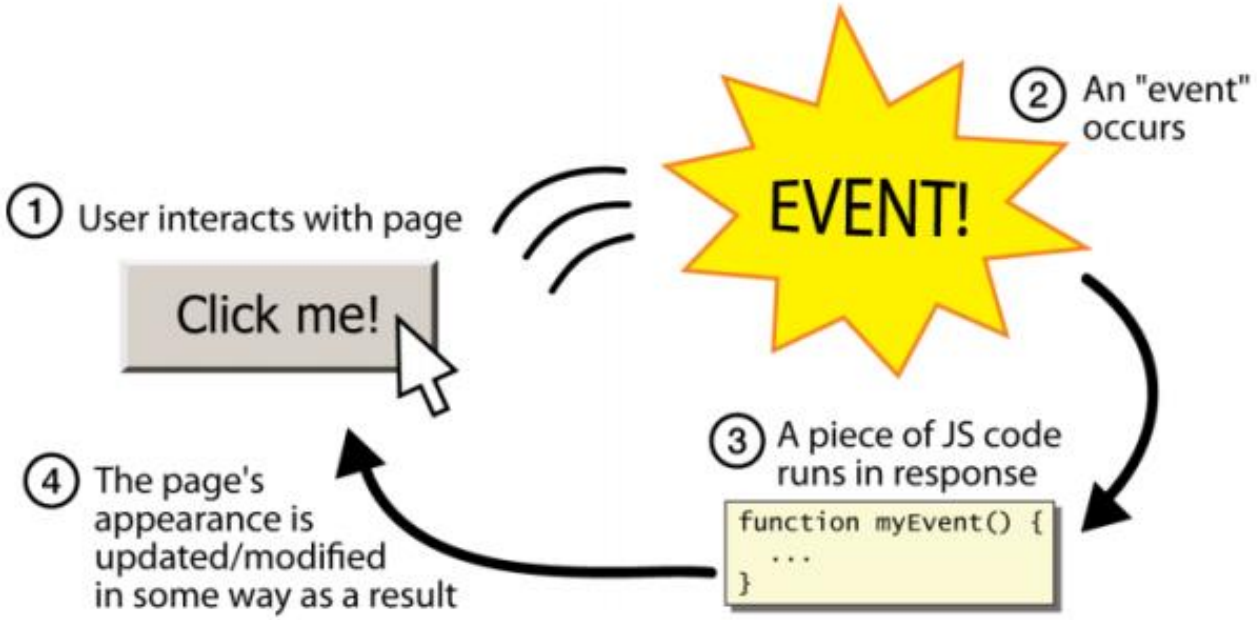
JS (Exemple)

```
function myFunction() {
alert("Hello World !") ;
}
```

Le code précédent pourrait être le contenu du fichier ***example.js*** lié à une page ***HTML*** . La fonction peut être appelée lors d'un évènement.

Introduction au langage JavaScript

Programmation par événements



Contrairement à Java ou C, un code JS n'a pas de ***main*** , mais réponds plutôt à un évènement.

Programmation par **évènements** : le code est exécuté suite à des évènements utilisateur.

Introduction au langage JavaScript

Les handlers

HTML (Modèle)

```
<element attributes onclick="function();">...
```

JS (Exemple)

```
<div onclick="myFunction();">Click me!</div>
```

Output

Click me!

Les fonctions JS sont utilisées comme des handlers d'évènements.

En cas d'évènement sur l'élément, le handler (la fonction) est appelée **onclick** un des (nombreux) évènements ***HTML*** disponibles

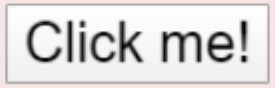
Introduction au langage JavaScript

L'élément <button>

JS (Exemple)

```
<button onclick="myFunction();">Click me!</button>
```

Output



Le contenu de l'élément button soit c'est du texte ou bien une image.

Pour fabriquer un bouton actif il suffit de :

1. Choisissez un évènement (par exemple, le clic souris).
2. Ecrivez une fonction JS qui sera appelée lorsque l'évènement se produit (le handler).
3. Attachez la fonction (le handler) à l'évènement du bouton.

Introduction au Document Object Model

La fonction document.getElementById

Donne l'accès à un élément HTML via la valeur de son attribut ***id*** (en HTML, la valeur de chaque attribut ***id*** doit être unique).

JS (Exemple)

```
let name = document.getElementById("id");
```

document.getElementById retourne l'objet DOM d'un élément HTML ayant un ***id*** donné (notez qu'on omet le caractère ***#***)

Introduction au Document Object Model

La fonction document.getElementById

Donne l'accès à un élément HTML via la valeur de son attribut id.

HTML (Exemple)

```
<img id="pokeball" src="img/pokeball.jpg" alt="a pokeball" />
<button onclick="changeImage();">Click me!</button>
```

JS (Exemple)

```
function changeImage() {
  let pokeballImg = document.getElementById("pokeball");
  pokeballImg.src = "img/mystery.gif";
}
```

Output (Avant le Click)

Introduction au Document Object Model

La fonction document.getElementById

Donne l'accès à un élément HTML via la valeur de son attribut id

HTML (Exemple)

```
<img id="pokeball" src="img/pokeball.jpg" alt="a pokeball" />
<button onclick="changeImage();">Click me!</button>
```

JS (Exemple)

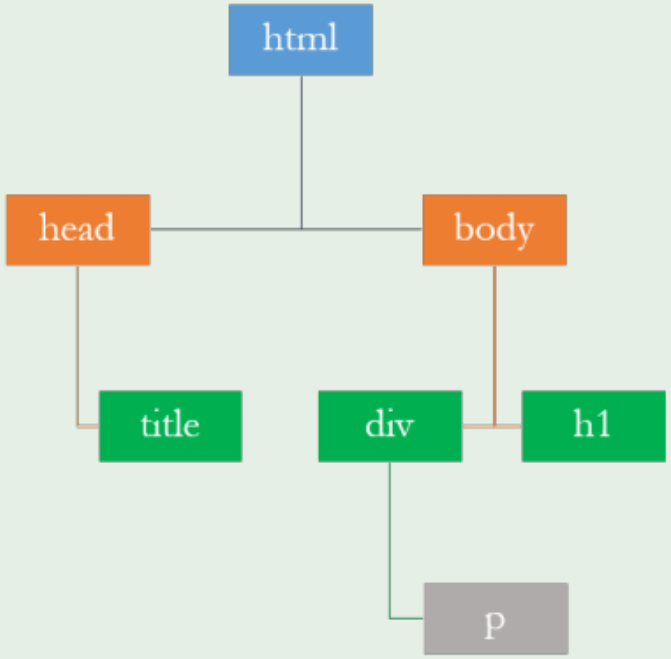
```
function changeImage() {
  let pokeballImg = document.getElementById("pokeball");
  pokeballImg.src = "img/mystery.gif";
}
```

Output (Après le Click)

Introduction au Document Object Model

L'arbre HTML

Arbre HTML (Exemple)



HTML (Exemple)

```
<html>
  <head>
    <title>...</title>
  </head>
  <body>
      <h1>...</h1>
      <div>
        <p>...</p>
      </div>
  </body>
</html>
```

Introduction au Document Object Model

Le Document Object Model (DOM)

Un ensemble d'objets JS liés à chaque éléments de la page HTML.

Chaque élément de la page HTML en cours corresponds à un objet DOM.

Les fonctions JS peuvent manipuler ces objets et lire ou modifier les attributs des éléments HTML correspondants :

- par exemple, tester si un ***INPUT*** de type ***RADIO*** a été coché.

Les fonctions JS peuvent changer ces objets et donc modifier les éléments HTML correspondants :

- par exemple, insérer un nouveau texte dans le contenu d'un élément ***DIV*** .

Les fonctions JS peuvent changer le style CSS des objets

- par exemple, changer la couleur du texte d'un élément ***P*** .

Introduction au Document Object Model

Obtenir les objets DOM en JS

Il y a toujours plusieurs façons de récupérer l'objet DOM d'un élément HTML particulier en navigant dans l'arbre HTML.

La plupart du temps, on utilise les fonctions suivantes :

- via la valeur de l'attribut ***ID*** :

  document.getElementyById(...)

- via un sélecteur CSS :
  - ***document.querySelector(...)***
  - ***document.querySelectorAll(...)***

On peut fabriquer de nouveaux éléments : ***document.createElement(...)***

Introduction au Document Object Model

Obtenir les objets DOM en JS

Dans le code HTML on a un élément P avec l'attribut id ayant la valeur october :

HTML (Exemple)

```
<p id="october"></p>
```

Dans le code JS on peut récupérer l'objet DOM correspondant à cet

élément HTML :

JS (Exemple)

```
let pTag = document.getElementById("october");
```

Introduction au Document Object Model

Contenu d'un objet DOM

Un objet DOM contient tous les attributs HTML de l'élément correspondant. Par exemple :

HTML (Exemple)

```
<img src="images/puppy.png" alt="A fantastic puppy photo"/>
```

est lié à l'objet DOM (qu'on appelle par exemple puppyImg) qui a les propriétés suivantes :

- ***puppyImg.src*** : la chaîne de caractères

  "images/puppy.png" (définie par le HTML)

- ***puppyImg.alt*** : la chaîne de caractères

  "A fantastic puppy photo" (définie par le HTML)

Introduction au Document Object Model

Accéder aux propriétés d'un objet DOM

HTML (Exemple)

```
<p>See our <a href="sale.html" id="saleslink">Sales</a>
today !</p>
<img id="icon" src="images/borat.jpg" alt="Borat" />
<caption class="photo user-upload">Beauty.</caption>
```

JS (Exemple)

```
let icon = document.getElementById("icon");
let theLink = document.getElementById("saleslink");
let caption = document.querySelector("caption");
```

Introduction au Document Object Model

Accéder aux propriétés d'un objet DOM

Propriété	Description	Exemple
`tagName`	nom de la balise HTML	`icon.tagName` est : `"IMG"`
`className`	la classe CSS classes de l'élément	`caption.className` est : `"photo user-upload"`
`src`	l'URL qui donne accès au fichier source de l'image	`icon.src` est : `"images/borat.jpg"`
`href`	l'URL vers où le lien est dirigé	`theLink.href` est : `"sale.html"`

Introduction au Document Object Model

La propriété innerHTML

Tous les éléments DOM possèdent la propriété innerHTML qui donne accès au contenu de l'élément (sous la forme d'un string) :

HTML (Exemple)

```
<ul id="thing">
  <li>Thing 1</li>
  <li>Thing 2</li>
</ul>
```

JS (Exemple)

```
let elm = document.querySelector("#thing li");
alert(elm.innerHTML);                          // "Thing 1"
```

L'expression ***document.querySelector("#thing li")*** sélectionne le premier élément ***LI*** contenu dans l'élément dont l' ***id*** est ***"thing"*** .

Introduction au Document Object Model

Utilisation des objets DOM

Si on change une ou plusieurs propriétés d'un objet DOM, le navigateur rafraîchit automatiquement l'affichage pour prendre en compte ce changement.

C'est de cette façon qu'on donne un comportement à une page HTML (qui sans quoi serait statique) :

- On utilise JS pour changer, en fonction de certains évènements, les propriétés et/ou le contenu de certains des objets DOM de la page.

Introduction au Document Object Model

Modifier un objet DOM

Avant l'exécution du JS, on a :

HTML (Exemple)

```
<a id="fb-link" href="http ://facebook.com">Facebook</a>
```

Après l'exécution du JS suivant :

JS (Exemple)

```
let link = document.getElementById("fb-link");
link.innerHTML = "FB"
```

On a :

JS (Exemple)

```
<a id="fb-link" href="http ://facebook.com">FB</a>
```