# Introduction à Javascript

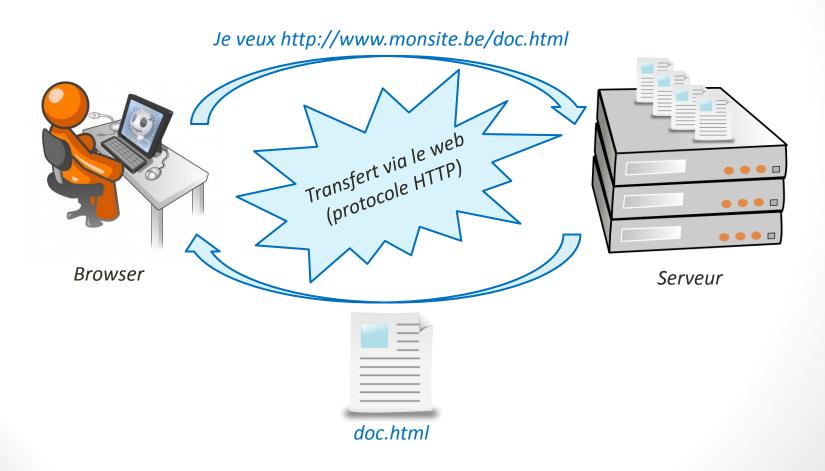
Technologies web HENALLUX — IG2 — 2015-2016

# Au programme...

- Rappels d'IG1
  - Site web, HTML, CSS
- Page statique et page HTML-dynamique
  - Différences ? Utilité de Javascript
- Introduction à Javascript
  - Présentation du langage Javascript
  - Premiers pas en Javascript
- Organisation du cours
  - Déroulement du cours, évaluation, etc.

#### Un site web

Comment fonctionne un site web?



#### Un site web

- Quelles sont les ressources utilisées ?
  - un navigateur utilisé par l'internaute
    - qui envoie les demandes,
    - réceptionne les réponses et
    - gère l'affichage (comprend les standards HTML et CSS).
  - un serveur
    - serveur = à la fois machine physique et logiciel
    - machine physique accessible via internet
    - logiciel-serveur capable de répondre aux demandes de fichiers (par exemple Apache)

#### Un site web

- Pour pouvoir communiquer entre eux, le navigateur et le serveur utilisent...
  - (aspect concret) le protocole http (<u>hypertext transfer protocol</u>)
    - pour envoyer une demande « je veux doc.html »
    - pour envoyer la réponse « voici le contenu de doc.html »
  - (aspect logique) des standards du web
    - HTML: <u>HyperText Markup Language</u>
    - CSS: <u>Cascading Style Sheets</u>
    - organisés par le W3C
    - conventions plus ou moins bien respectées par les navigateurs

#### HTML

</html>

Un document HTML est un fichier texte.

```
<!doctype html>
<html>
<head>
Partie "en-tête":
... - infos techniques (titre, langue, encodage, ...)
- d'autres éléments non directement visibles

<body>
Partie "corps":
... - éléments affichés sur la page web
</body>
```

#### HTML

- Le langage HTML (parent/descendant du XML) repose sur des balises.
  - Toute balise ouvrante doit être fermée : ...
  - Balises sans intérieur : <br/> <meta ... />
  - Balises imbriquées, pas de croisement ! <em> </em>
  - Par convention : écrite en minuscules
- Les blancs répétés sont ignorés (ce qui permet d'indenter).
- Exemple pour la partie en-tête :

```
<head>
    <meta charset="utf-8" />
    <title>Ma première page HTML</title>
    </head>
```

#### CSS

- Séparation forme / contenu
  - Pourquoi ?
  - HTML s'occupe du fond/contenu CSS s'occupe de la forme.
- Trois utilisations possibles du CSS :
  - Format inline :

```
<h2 style="color:blue">Mon sous-titre</h2>
```

Format interne (dans l'en-tête) :

```
<style>
  h2 {color:blue}
  em {font-family:courier; font-size:250%}
</style>
```

Format externe (dans un fichier séparé) :

```
<link href="style.css" rel="stylesheet" type="text/css" />
```

#### CSS

Chaque règle CSS est composée...

```
h2 { color:blue ; font-size:250% }
```

- d'un sélecteur (désigne cibles de la règle),
- d'une liste de déclarations, chacune d'elles étant composée...
  - d'une propriété (indique quel aspect modifier) et
  - d'une valeur (indique comment le modifier).
- Grand choix de sélecteurs plus ou moins complexes :
  - Balise
  - Identificateur #maphoto (en HTML: id="maphoto")
  - Classe .nom (en HTML: class="nom")
  - + combinaisons: #contenu p.intro a:first-of-type

# Pages HTML-dynamiques

- Statique et HTML-dynamique
  - Quelles sont les différences ?
- Que permet Javascript ?

Ensuite: introduction au langage Javascript

## Statique et HTML-dynamique

- Une page web statique...
  - possède un contenu et une apparence déterminée par le code HTML/CSS envoyé par le serveur;
  - et ce contenu et cette apparence ne changent pas.
- Une page web HTML-dynamique...
  - possède un contenu et une apparence initiales déterminées par le code HTML/CSS envoyé par le serveur;
  - et ce contenu et cette apparence peuvent être modifiées!
- Sur une page HTML-dynamique,...
  - le navigateur gère les modifications en local (sans le serveur).
  - le code envoyé contient des scripts indiquant comment le navigateur doit (ré)agir, généralement écrits en Javascript.

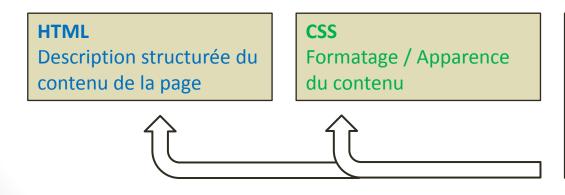
# Statique et HTML-dynamique

JS permet le **DHTML** (= Dynamic HTML).

DHTML

=

HTML + CSS + Javascript



#### **Javascript**

Actions à exécuter par le navigateur, pouvant modifier le contenu et l'apparence de la page web

#### Que permet Javascript?

- Javascript permet d'écrire des scripts clients.
  - Scripts : programmes destinés à être interprétés directement
    - plutôt que compilés puis exécutés...
    - (la différence n'est plus aussi nette que ça aujourd'hui.)
  - Clients : tout se déroule sur l'ordinateur de l'internaute.
- Par exemple, Javascript peut
  - ajouter l'heure quelque part dans le contenu de la page
  - afficher un message quand l'utilisateur clique sur un bouton
  - afficher ou cacher un menu selon la position du curseur
  - demander une information à l'utilisateur (son nom)
  - manipuler le navigateur (redirection vers une autre page)
  - permettre de basculer entre un thème clair et un thème foncé

## Introduction à Javascript

- Présentation du langage Javascript
  - Quelles sont ses caractéristiques principales ?
- Comment utiliser Javascript sur une page web?
- Premières lignes en Javascript

Ensuite : organisation du cours

- Javascript est...
  - un langage de programmation
  - impératif,
  - orienté objet,
  - événementiel et
  - faiblement / non typé,
  - où les fonctions sont des objets de premier ordre.
  - Qu'est-ce que tout cela signifie ?

- JS est un langage de programmation.
  - Contrairement à HTML et à CSS, qui sont des langages informatiques de description (ils ne permettent pas d'écrire un programme)
- ... un langage impératif (comme C, Java).
  - programme = séquence d'instructions
- ... un langage orienté objet (comme Java).
  - mais Javascript a une approche OO bien différente de Java!
     (on en reparlera!)àIntroduction au langage Javascript

- Javascript est conçu pour la programmation événementielle.
  - = programmation où certains modules s'exécutent en réponse à des éléments-déclencheurs
  - Quelques exemples d'événements :
    - clic sur un bouton
    - passage du curseur sur une image
    - passage du curseur sur un élément de menu
    - fin du chargement de la page
  - Programmation événementielle = préciser ce qui doit se passer en réponse à des événements
    - ≠ programme C ou Java avec 1 module « main »

- JS est un langage non typé.
  - Le type du contenu d'une variable peut évoluer!

```
var x = 3;
x = "One more block";
```

- Quand on déclare une variable, on ne précise pas de type (idem pour les arguments d'une fonction).
- Danger : certaines erreurs ne sont pas repérées automatiquement !

- JS traite les fonctions comme des objets de premier ordre.
  - Objets de premier ordre = objets qu'on peut
    - affecter à une variable,
    - passer comme argument à une fonction,
    - retourner comme résultat d'une fonction.

```
Exemples :
```

```
var somme = function (x,y) { return x+y; };
z = somme(2,3);
var applique = function (g,a,b) { return g(a,b); };
z = applique(somme,2,3);
```

- Exemple (événementiel + fonctions comme éléments de premier ordre)
  - Site web permettant l'affichage de fiches signalétiques des étudiants en 3 formats : simple, normal et détaillé.

```
var modeChoisi; // Simple = 0, Normal = 1, Détaillé = 2
function affSimple() { ... }
function affNormal() { ... }
function affDetaille() { ... }
var tabAffiche = [affSimple, affNormal, affDetaille];
boutonAffiche.onclick = tabAffiche[modeChoisi];
```

#### Javascript ≠ Java!

	Javascript	Java
Créateurs	Netscape	Sun / Oracle
Exécution	interprété	compilé
Objets	à base d'objets	orienté objet
Typage	non typé, dynamique	fortement typé, statique
Utilisation	propre au web	langage indépendant
Fichiers	aucun accès	lecture, écriture
Sécurité	code public	

- Un point commun: l'utilisation d'un garbage collector.
- Javascript devait s'appeler Mocha ou LiveScript.
- Le nom officiel du standard Javascript est ECMA-262, défini sur le site http://www.ecma-international.org/ecma-262/5.1/

- Solution 1 : code "interne"
  - Dans une balise script:

```
<script> codeJS </script>
```

- (Obsolète) ajouter language="javascript" ou type="text/javascript" pour distinguer de VBScript ou JScript
- Il s'agit d'exécution synchrone (immédiate).
  - On peut avoir plusieurs balises script.
  - Le navigateur exécute le contenu dès qu'il le rencontre lors de la lecture du fichier HTML.

- Solution 1 : code "interne" (suite)
  - Texte à afficher si JS est désactivé :

```
<noscript>
  Ce browser ne supporte pas JS !
</noscript>
```

• [Firefox] pour désactiver JS : modifier la propriété javascript.enabled dans about:config.

```
<html>
<head>
 <script>
  alert("Bienvenue sur ma page web !");
</script>
</head>
<body>
 Ceci est le contenu du premier paragraphe.
 Et voici un
   <script>
   document.write("<a href='http://www.google.be'>lien</a>");
   </script>
  vers Google!
 </body>
</html>
```

```
<html>
 <head>
   <script>
     alert("Script dans head !");
     function afficheHeure () {
       var now = new Date ();
       var h = now.getHours();
       var m = now.getMinutes();
       var s = now.getSeconds();
       document.write(h + ":" + m + ":" + s);
   </script>
 </head>
 <body>
   <script>alert("J'entre dans body !");</script>
   Il est exactement : <script>afficheHeure();</script>k/p>
   Ceci est mon deuxième paragraphe.
   <script>alert("Après 2e paragraphe.");</script>
   Il est maintenant <script>afficheHeure();</script>.
 </body>
</html>
```

- Solution 2 : code "inline"
  - Dans un attribut correspondant à un événement :

```
<button onclick="codeJS">Cliquez moi!</button>
```

Dans une référence href :

```
<a href="javascript:codeJS">texte</a>
```

• Il s'agit d'exécution asynchrone (reportée à plus tard, quand un événement déclencheur se produit).

```
<html>
  <head>
    <script>
      function message () {
        alert("Voici un message !");
    </script>
  </head>
  <body>
    Pour recevoir un message, cliquez
      <a href="javascript:message();">sur ce lien</a>
      ou passez la souris sur ce
      <button onmouseover='message();">bouton</button>
      .
  </body>
</html>
```

- Solution 3 : code "externe"
  - Dans un fichier séparé lié via une balise script :
     <script src="nomFichier.js"></script>
  - Ici, exécution synchrone (immédiate).

#### Résumé des 3 solutions

- inline et interne
  - peut être utile lors des tests (ou exercices)
  - à éviter dans la version finale d'un site

#### externe

- de préférence
- placer le lien <script> dans l'en-tête <head>
- principal avantage : le code Javascript est chargé séparément du fichier HTML (peut-être "caché" indépendamment)

#### Premières lignes en JS

- Sorties
  - console.log(txt): vers la console du navigateur
  - document.write(txt): vers le document HTML
  - alert(txt): dans une fenêtre popup
- Entrées
  - res = confirm(txt) : demande une confirmation
    - renvoie true si l'utilisateur a cliqué sur "Ok"
    - renvoie false si l'utilisateur a cliqué sur "Cancel"
  - res = prompt(txt [,valDefaut] ):demande une valeur
    - propose éventuellement une valeur par défaut

# Premières lignes en JS

Déclaration de variable

```
var x;var x = prompt("Entrez un nombre.", 7);
```

Déclaration de fonction

```
• function alertImportant (msg) {
   var texte = "MESSAGE IMPORTANT :\n" + msg;
   alert(texte);
}
• function carre (x) {
   return x * x;
}
```

#### Organisation du cours

#### Matière

- Le langage Javascript
  - Bases de Javascript
  - L'orienté objet en Javascript
  - Manipulation du DOM en Javascript
- Sujets connexes
  - Json, jQuery, envoi de données au serveur, ...
- Applications sur machine
  - Munissez-vous de vos notes d'IG1 (HTML/CSS) aussi !

#### Évaluation

- Examen écrit, à court ouvert, avec PC à disposition
- Comprend la matière d'IG1 et celle d'IG2!
- Attention : cours ouvert ≠ examen pour touriste!

#### Liens utiles

- Mozilla Developer Network
  - https://developer.mozilla.org/en-US/docs/Web/JavaScript
  - (également disponible en français)
- W3 Schools
  - http://www.w3schools.com/