

## Séance 7

# HTML, CSS et Javascript



Ce(tte) œuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Pas de Modification 4.0 International.

# Rappels

- Manipulation de fichiers texte et binaire
  - Ouverture et fermeture du fichier
  - Lecture et écriture d'un fichier
  - Gestion des erreurs d'entrée/sortie
- Encodage de l'information
  - Encodage des caractères, standard Unicode
  - Sérialisation avec le module pickle
  - Stockage bas niveau avec le module struct

# Objectifs

- Comprendre le **World Wide Web** (WWW)
  - Identification des ressources avec des URLs
  - Principe du client-serveur
  - Couches principales de l'Internet
- **Langages** du web
  - Création de documents avec HTML (*HyperText Markup Language*)
  - Définition de styles avec CSS (*Cascading Style Sheet*)
  - Aspect dynamique côté client avec Javascript

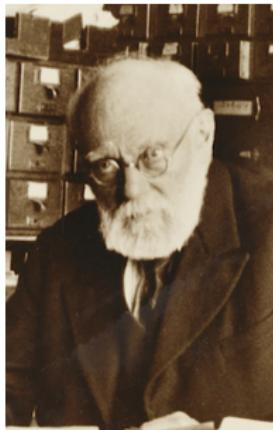


World Wide Web

# World Wide Web (WWW)

- Espace d'informations contenant des **ressources**
  - Documents texte et autres ressources
  - Ressources liées entre elles par des liens hypertextes
  - Identification des ressources par des URLs
- Bref **historique**
  - Inventé par Paul Otlet (visionnaire) en 1934
  - Inventé par Tim Berners-Lee en 1989
  - Premier navigateur web en 1990 avec l'aide de Robert Cailliau

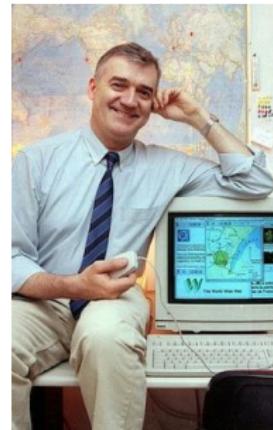
# Personnages clés



Paul Otlet  
1868–1944 (BE)



Sir Tim Berners-Lee  
1955– (UK)



Robert Cailliau  
1947– (BE)

# Internet (1)

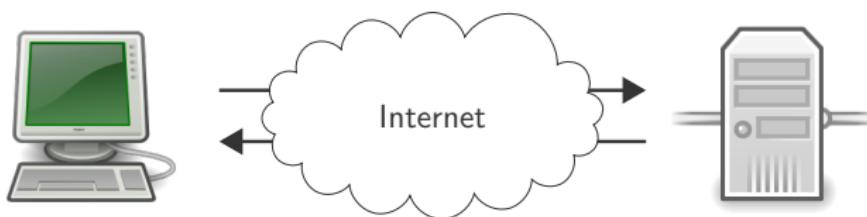
- Système composé d'un ensemble de **réseaux interconnectés**  
*Relie des milliards de machines entre elles*
- Lancé en Californie en **1969** (ARPANET)  
*Premier échange d'un message électronique entre deux machines*
- Premières **connexions intercontinentales** en 1988 (NORDUnet)  
*Réseau d'échange de données pour cinq pays nordiques d'Europe*
- Supporte de nombreuses **ressources et services**  
*WWW, app, e-mail, téléphonie, partage de fichiers, jeu en ligne...*

# Internet (2)

early milestones	Key Layers of the Internet	milestones
email@-1971 Ray Tomlinson	CONTENT	1991-.html Berners-Lee & Cailliau
Archie-1990 Emtage & Deutsch	SEARCH ENGINE*	1998-Google Brin & Page
DOS Houdini-1986 Neil Larson	BROWSERS	1993-Mosaic Marc Andreessen
( Vannevar Bush, Ted Nelson, Douglas Engelbart )	WORLD WIDE WEB	1990-http:// Tim Berners-Lee
ARPANET-1969 J.C.R. Licklider	INTERNET	1975-TCP/IP Cerf & Kahn
SAGE-1956 George Valley	NETWORKS	1973-Ethernet Robert Metcalfe
Z3-1941 Konrad Zuse	COMPUTERS	1976-Apple Jobs & Wozniak

# Client/Serveur

- Communication entre **un client et un serveur**
  - 1 Le client se connecte au serveur
  - 2 Le serveur accepte la connexion
  - 3 Le client et le serveur communiquent
    - 1 Le client envoie une requête au serveur
    - 2 Le serveur analyse la requête et répond au client
- La connexion peut être **fermée** par le client ou le serveur



# Serveur web

- Serveur utilisé pour **publier des sites web**

*Communication avec ce serveur à l'aide du protocole HTTP*

- Support d'autres protocoles pour offrir d'**autres services**

*Envoi d'e-mails, streaming, transfert de fichiers...*

- **Serveur web publics** reliés à Internet

*Certains serveurs web sont privés et limités à un réseau local*

# Uniform Resource Locator (URL)

- Une **URL** référence une ressource web sur un réseau  
*Localisation sur le réseau et mécanisme pour la récupérer*
- Plusieurs **mécanismes d'accès** possibles
  - http indique une page web
  - ftp utilisé pour le transfert de fichier
  - mailto permet l'envoi d'un e-mail
  - jdbc permet d'accéder à une base de données
  - ...

# HyperText Transfer Protocol (HTTP)

- Protocole de **communication de données** pour le WWW

*Coordonné par l'Internet Engineering Task Force (ETF)  
et le Word Wide Web Consortium (W3C)*

- Utilisation d'**hyperliens** entre des nœuds

*Permet de naviguer à travers le web*

- Protocole de type **requête-réponse**

*Construit suivant une architecture client/serveur*

fa>

HTML

# HyperText Markup Language (HTML)

- Langage de balisage pour représenter des pages web

*Fichier texte permettant d'écrire de l'hypertexte*

- Plusieurs autres rôles

- Information sémantique par rapport au contenu
- Information de mise en forme du contenu
- Inclusion de ressources multimédia (image, formulaire...)

# Historique

- **HTML 2.0** (24 novembre 1995)
- Le W3C publie **HTML 3.2** (14 janvier 1997)  
*Le W3C reprend le travail sur HTML, arrête par l'IETF*
- Trois variantes de **HTML 4.0** (18 décembre 1997)  
*HTML 4.0 strict, transitional et frameset*
- **HTML 5** publié comme recommandation W3C (28 octobre 2014)  
*Modernisation de la version 4 avec support multimédia, des images SVG, du MathML...*



# Élement et balise

- Élément du document délimité par des **balises**

*Typiquement une balise ouvrante et une balise fermante*

- Balise formée par le **nom de l'élément** entouré de chevrons

*Avec un / en plus pour distinguer la balise fermante*

- **Imbrication** de balises, en les fermant dans le bon ordre

*Dernière balise ouverte = première balise fermée*

```
1 <h1>Titre de premier niveau</h1>
2
3 <p>Voici un paragraphe avec un <b>mot en gras</b> dedans.</p>
```

# Élement et balise



```
1 <h1>Titre de premier niveau</h1>
2
3 <p>Voici un paragraphe avec un <b>mot en gras</b> dedans.</p>
```

# Attribut

- Ajout d'**attribut** pour configurer l'élément

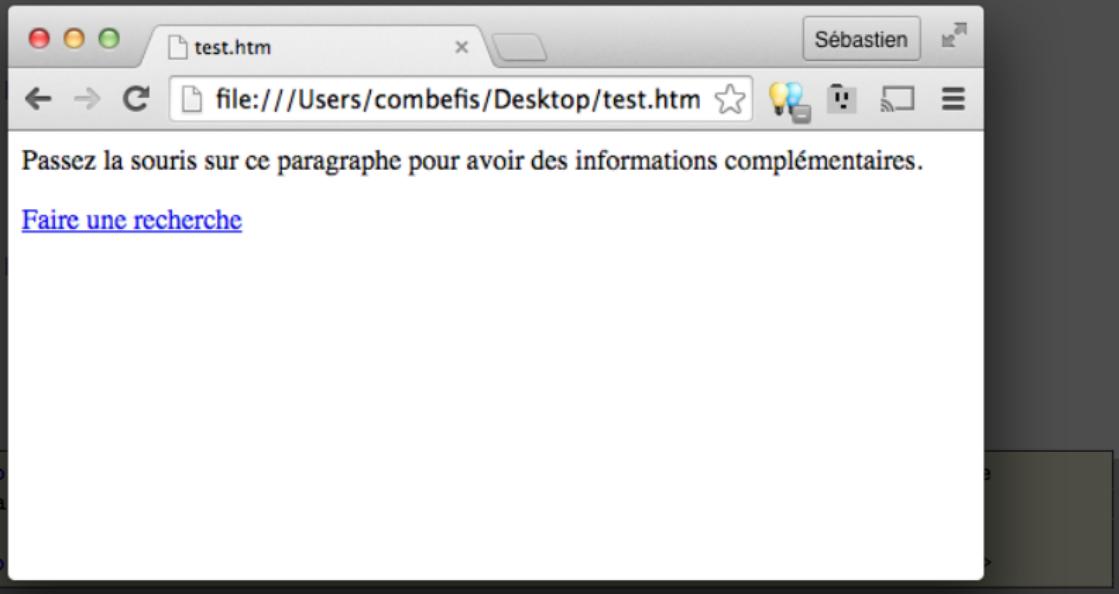
*Se placent dans la balise ouvrante*

- Un attribut est composé d'**une clé et d'une valeur**

*Valeur entre guillemets doubles, et parfois optionnelle*

```
1 <p title="Information complémentaire">Passez la souris sur ce  
paragraphe pour avoir des informations complémentaires.</p>  
2  
3 <p><a href="http://www.google.be">Faire une recherche</a></p>
```

# Attribut



# Entête et corps

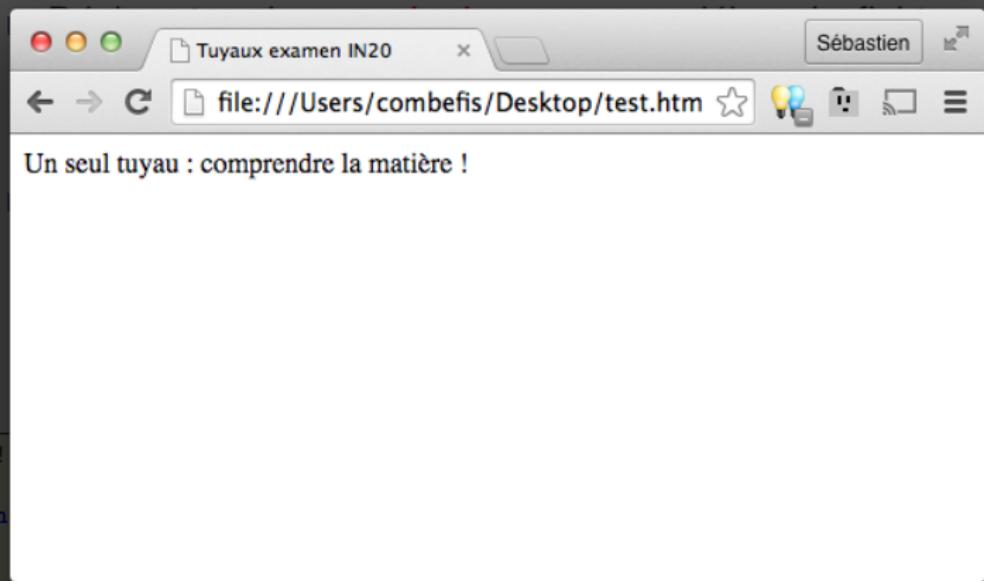
- Déclaration du **type de document** au début du fichier

*<!DOCTYPE html> pour l'HTML 5*

- Le **document** est déclaré avec l'élément `html`
  - Entête du document (méta information) avec `head`
  - Corps du document (contenu visible) avec `body`

```
1 <!DOCTYPE html>
2
3 <html>
4     <head>
5         <title>Tuyaux examen IN20</title>
6     </head>
7     <body>
8         <p>Un seul tuyau : comprendre la matière !</p>
9     </body>
10 </html>
```

# Entête et corps



# Titre

- Plusieurs **niveaux de titre** (de 1 à 6)

*Au maximum un seul titre de niveau 1 par page*

- Utilisation des éléments h1 à h6

```
1 <h1>CU20 Cuisinologie</h1>
2
3 <h2>Chapitre 1 : Dépieauter la bête</h2>
4
5 <p>Identifiez d'abord où est la peau...</p>
6
7 <h2>Chapitre 2 : Température optimale du four</h2>
8
9 <p>Le gradient de température doit être Lipschitz continu...</p>
```

Sébastien

← → ⌛ file:///Users/combefis/Desktop/test.htm ⭐ 🌟 🚧 📺 📞

# CU20 Cuisinologie

## Chapitre 1 : Dépieauter la bête

Identifiez d'abord où est la peau...

## Chapitre 2 : Température optimale du four

Le gradient de température doit être Lipschitz continu...

```
1 <h  
2 <h  
3 <p Le gradient de température doit être Lipschitz continu...  
4 <h  
5 <p>Le gradient de température doit être Lipschitz continu...</p>
```

# Section et paragraphe

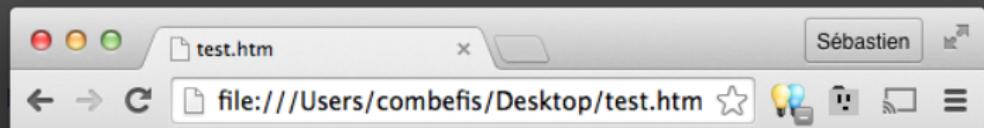
- Insertion d'une **section** avec l'élément section

*Possède un titre, plusieurs paragraphes et des sous-sections*

- **Paragraphe** inséré grâce à l'élément p

```
1 <section>
2   <h1>Série du mois</h1>
3
4   <p>La série du mois est American Horror Story. Il y a déjà 5
5     saisons à cette série, pour un total de 54 épisodes.</p>
6
7   <p>Il s'agit d'une série télévisée dramatique et d'horreur.</p>
8 </section>
```

# Section et paragraphe



## Série du mois

La série du mois est American Horror Story. Il y a déjà 5 saisons à cette série, pour un total de 54 épisodes.

Il s'agit d'une série télévisée dramatique et d'horreur.

```
1 <section>
2
3
4
5
6   <p>Il s'agit d'une série télévisée dramatique et d'horreur.</p>
7 </section>
```

# Lien hypertexte

- Lien hypertexte vers une ressource avec l'élément a

*Lien typiquement identifié par une URL*

- Attribut href pour spécifier la cible du lien

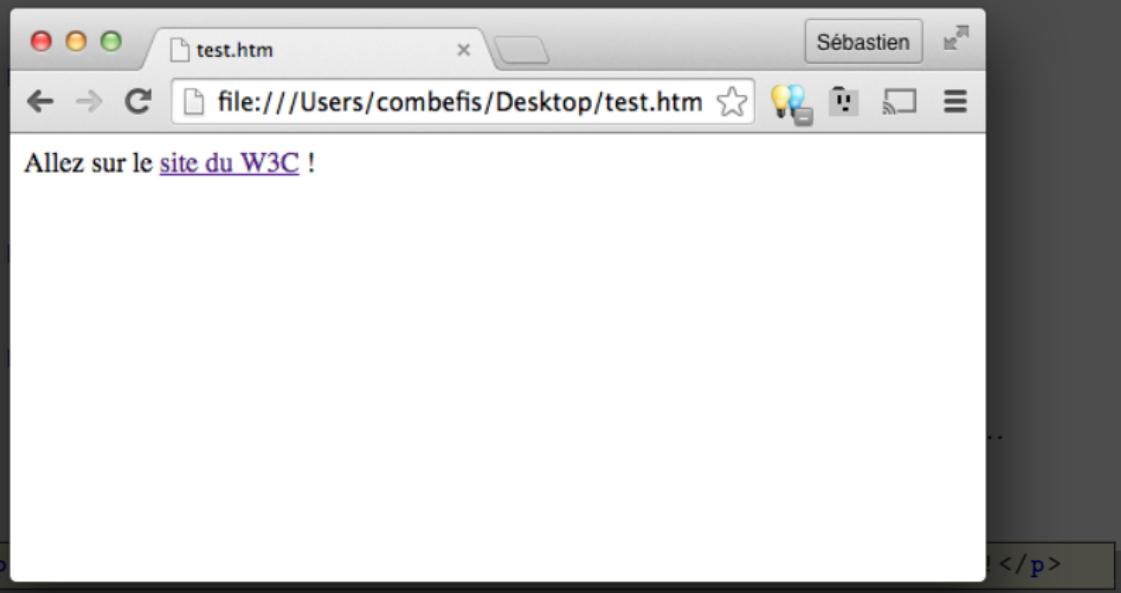
- Le texte du lien doit décrire la page liée

*Éviter des « ici », ou « cliquez ici », ou encore « ce lien »...*

1    

```
<p>Allez sur le <a href="http://www.w3.org">site du W3C</a> !</p>
```

# Lien hypertexte



# Formatage (1)

- Formatage de base de texte dans un paragraphe
  - Texte en gras avec l'élément **b**
  - Texte important avec l'élément **strong**
  - Texte en italique avec l'élément **i**
  - Texte emphasé avec l'élément **em**
- Positionnement du texte dans le texte
  - Texte en indice avec l'élément **sub**
  - Texte en exposant avec l'élément **sup**

## Formatage (2)

```
1 <p>Le <b>sel</b>, c'est un ion Na+ et un Cl-.</p>
2 <p>Mais le seul <strong>n'est pas électrique</strong>, encore
heureux !</p>
3 <p>D'abord dessus, et puis dessous.</p>
```



# Image

- Insertion d'une **image** avec l'élément `img`

*Chemin vers l'image identifié par une URL, ou en relatif*

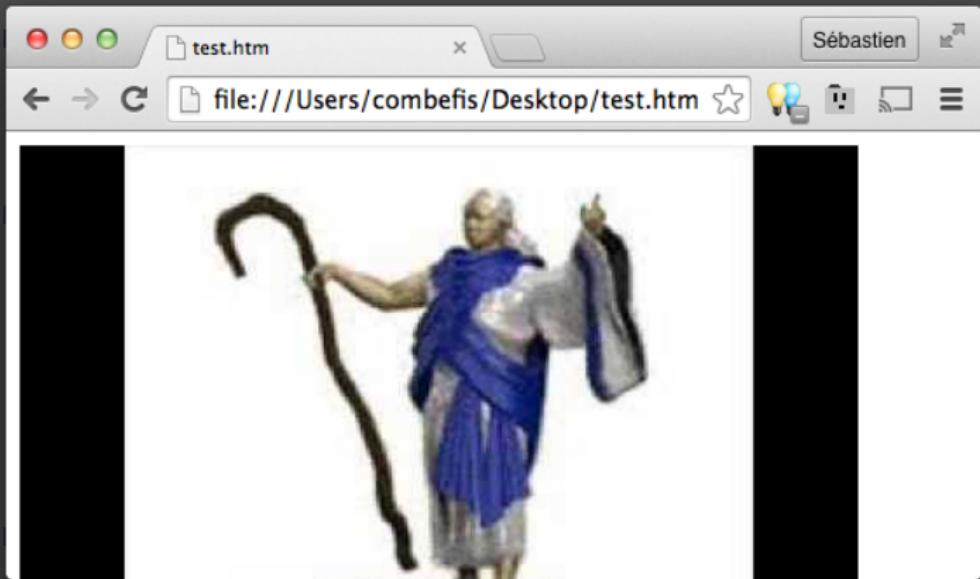
- Trois attributs pour **configurer l'image**

- Attribut `src` pour spécifier le chemin de l'image
- Attributs `width` et `height` pour les dimensions (pixels)
- Attribut `alt` pour un texte alternatif

- **Balise orpheline**, pas de balise fermante

```
1 
```

# Image



```
1 
```

# Tableau

- Insertion d'un **tableau** avec l'élément **table**
- Un tableau est décrit par des **lignes composées de cellules**  
*Lignes représentées par l'élément **tr** et cellule par **td** ou **th***

```
1 <table>
2   <tr>
3     <th>Nom</th>
4     <th>Prénom</th>
5   </tr>
6   <tr>
7     <td>Louis</td>
8     <td>Breyne</td>
9   </tr>
10 </table>
```

# Tableau

A screenshot of a web browser window titled "test.htm". The address bar shows "file:///Users/combefis/Desktop/test.htm". The page content displays the text "Nom Prénom" followed by "Louis Breyne". Below the browser window, the source code of the HTML file is visible in a code editor:

```
1 <t  
2  
3  
4  
5  
6  
7  
8  
9 </tr>  
10 </table>
```

# Liste

- Insertion d'une **liste** numérotée ou non avec l'élément ul ou ol

*Chaque élément de la liste est inséré avec l'élément li*

- Insertion d'une liste de **définitions** avec l'élément dl

*Mot inséré avec l'élément dt et définition avec dd*

```
1 <p>Quand mon réveil sonne le matin...</p>
2 <ol>
3   <li>Je le prolonge de 10 minutes</li>
4   <li>Quand il resonne, j'ouvre les yeux et j'attends un peu</li>
5   <li>Je me dis ensuite que je vais me lever</li>
6   <li>En fait non, je me rendors</li>
7 </ol>
```

# Liste

A screenshot of a web browser window titled "test.htm". The address bar shows the file path: "file:///Users/combefis/Desktop/test.htm". The page content displays a heading "Quand mon réveil sonne le matin..." followed by a numbered list:

1. Je le prolonge de 10 minutes
2. Quand il resonne, j'ouvre les yeux et j'attends un peu
3. Je me dis ensuite que je vais me lever
4. En fait non, je me rendors

The browser interface includes standard controls like back, forward, and search, along with a user profile icon for "Sébastien". A code editor view at the bottom shows the corresponding HTML and CSS code:

```
1 <p>  
2 <ol>  
3  
4  
5  
6     <li>En fait non, je me rendors</li>  
7 </ol>
```

# Formulaire

- Insertion d'un **formulaire** avec l'élément `form`

*Le formulaire se construit ensuite à l'aide de widgets*

- Deux attributs pour **configurer le formulaire**

- Attribut `method` pour spécifier la méthode d'envoi
- Attribut `action` identifie la page de traitement

```
1 <form method="post" action="login.php">
2   <p>Pseudo : <input type="text" name="pseudo" /></p>
3   <p>Mot de passe : <input type="password" name="password" /></p>
4   <p><input type="submit" value="Se connecter" /></p>
5 </form>
```

# Formulaire

The screenshot shows a web browser window titled "test.htm" with the URL "file:///Users/combefis/Desktop/test.htm". The browser interface includes standard controls like back, forward, and search, along with a user profile icon for "Sébastien".

The page content displays a login form with two text input fields labeled "Pseudo :" and "Mot de passe :" followed by a "Se connecter" button.

```
1 <f  
2  
3  
4 <p><input type="submit" value="Se connecter" /></p>  
5 </form>
```

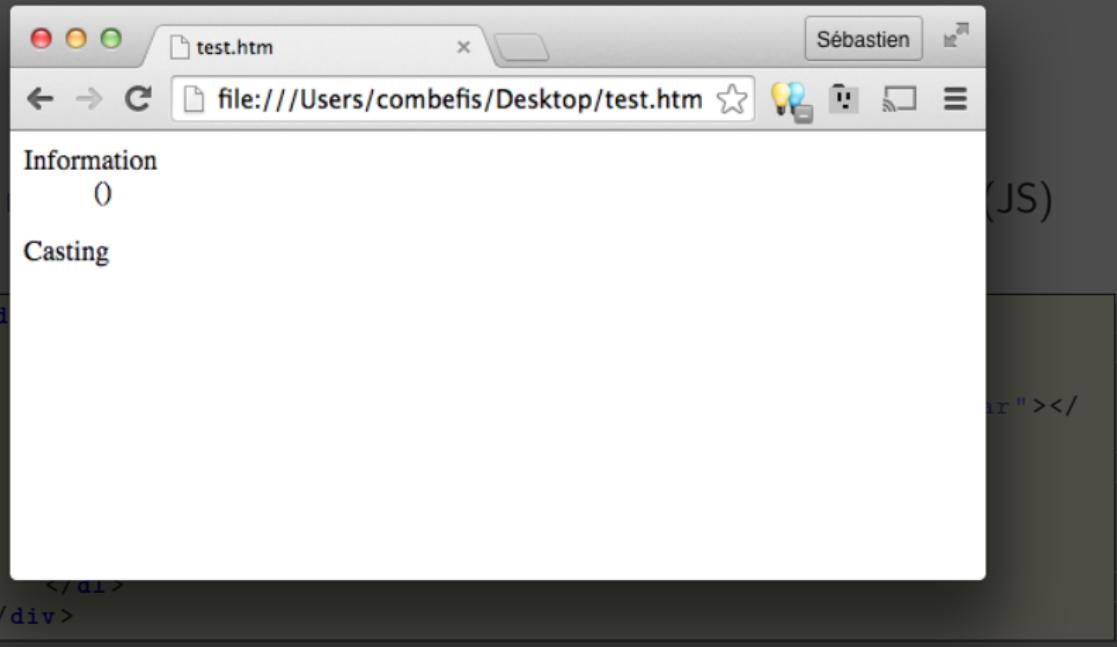
# Élement structurel

- Délimiter un bloc dans le document HTML
  - Dans le texte avec l'élément span
  - Et comme un bloc séparé du texte avec div
- Utilisé pour du style (CSS) ou du contenu dynamique (JS)

```
1 <div id="moviedetail">
2   <dl>
3     <dt>Information</dt>
4     <dd><span id="movietitle"></span> (<span id="movieyear"></span>)
5   </dl>
6   <dl>
7     <dt>Casting</dt>
8     <dd><span id="moviecasting"></span></dd>
9   </dl>
10 </div>
```

# Élement structurel

- Délimiter un bloc dans le document HTML



The screenshot shows a web browser window titled "test.htm" with the URL "file:///Users/combefis/Desktop/test.htm". The page content is displayed in two main sections: "Information" and "Casting". The "Information" section contains the number "0" and a "(JS)" label. The "Casting" section is currently empty. The browser interface includes standard navigation buttons (back, forward, search), a refresh button, and various icons for file operations and help.

```
1 <div>
2
3
4
5
6
7
8
9
10 </div>
```



CSS

# Cascading Style Sheets (CSS)

- Langage de **description de la présentation** de documents

*Fichier texte décrivant des règles de mise en page*

- **Historique** des différentes versions publiées

- CSS1 avec une cinquantaine de propriété (17 décembre 1996)
- Environ 70 nouvelles propriétés dans CSS2 (mai 1998)
- Découpe en 50 modules dans CSS 3 (premier draft en juin 1999)

# Lien avec HTML

- Trois manières d'attacher des **règles de style**
  - Directement avec l'attribut `style` sur un élément
  - Avec l'élément `style` dans la partie `head`
  - Dans un fichier séparé, avec l'extension `.css`
- Style depuis des **fichiers externes** avec l'élément `link`  
`<link rel="stylesheet" href="mystyle.css" type="text/css" />`

# Syntaxe

- Règle CSS composée de deux parties
  - Le **sélecteur** définit le(s) élément(s) affecté(s)
  - Le(s) **déclaration(s)** contiennent les règles
- Les règles s'appliquent **en cascade** aux éléments imbriqués

*Une déclaration se compose d'une propriété et d'une valeur*

```
1  h1 {  
2      color: red;  
3      border: 1px solid blue;  
4  }  
5  
6  p {  
7      text-decoration: underline;  
8  }
```

# Syntaxe



A screenshot of a web browser window titled "test.htm". The address bar shows "file:///Users/combefis/Desktop/test.htm". The main content area displays the text "CU20 Cuisinologie" in a large red font inside a blue-bordered box. Below it, the text "Chapitre 1 : Dépieauter la bête" is displayed in a large black font. A link "Identifiez d'abord où est la peau..." is shown below. The browser interface includes standard controls like back, forward, and search.

```
1 h1
2
3 }
4
5 p {
6   text-decoration: underline;
7 }
```

# Sélecteur

- Sélection directe d'un élément en mentionnant son nom

*h1 {...}*

- Sélectionner un élément unique par son identifiant

*form#login {...} ou #login {...}*

- Sélectionner tous les éléments d'une classe

*p.firstpara {...} ou .firstpara {...}*

- Sélection multiple de plusieurs éléments

*h1, div.content, #submit {...}*

# Dimension

- Largeur et hauteur d'un élément avec width et height

*Ces propriétés forcent la taille de l'élément*

- Tailles minimales et maximales avec min-width, max-width...

*Utiles lorsque l'utilisateur redimensionne la fenêtre*

- Dimensions typiquement renseignées en pixels

*D'autres unités possibles comme l'em par exemple*

# Marges et padding

- On peut régler les **marges** sur les quatre côtés

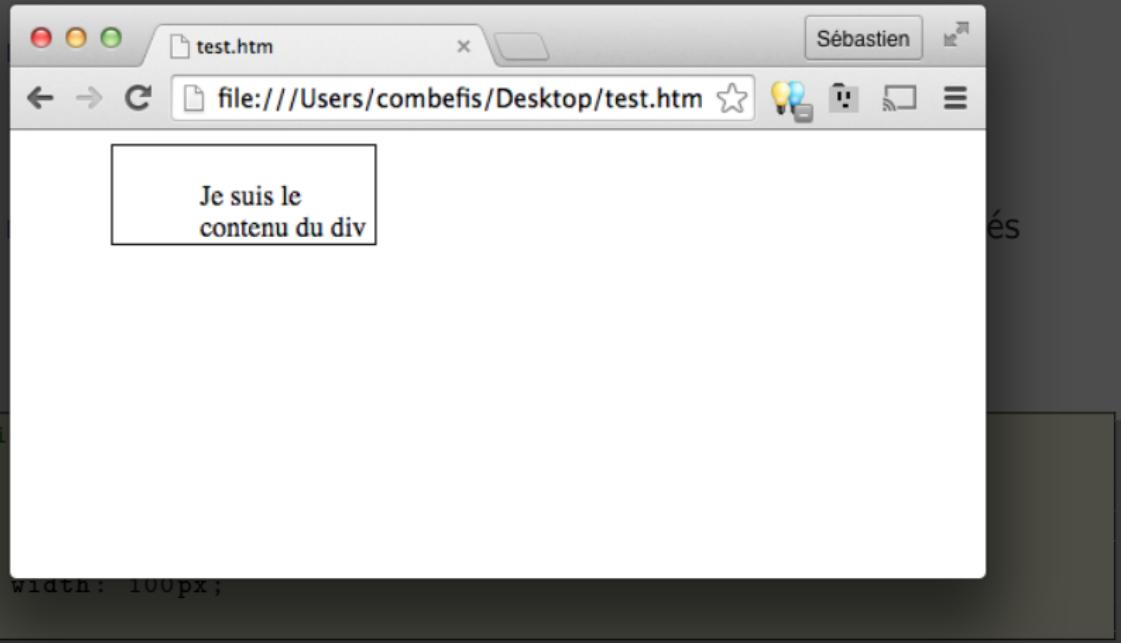
*On a accès aux marges top, bottom, right et left*

- On peut régler le **padding** également sur les quatre côtés

*Le padding correspond à la marge intérieure*

```
1 div {  
2   border: 1px solid;  
3   margin-left: 50px;  
4   padding-top: 20px;  
5   padding-left: 50px;  
6   width: 100px;  
7 }
```

# Marges et padding



The screenshot shows a web browser window titled "test.htm" with the URL "file:///Users/combefis/Desktop/test.htm". The page content is a single 

element containing the text "Je suis le contenu du div". Below the browser window, a code editor displays the following CSS code:

```
1  div {  
2      width: 100px;  
3  }  
4  
5  
6  
7 }
```

# Pseudo-classe

- Une **pseudo-classe** fait référence à un état d'un élément

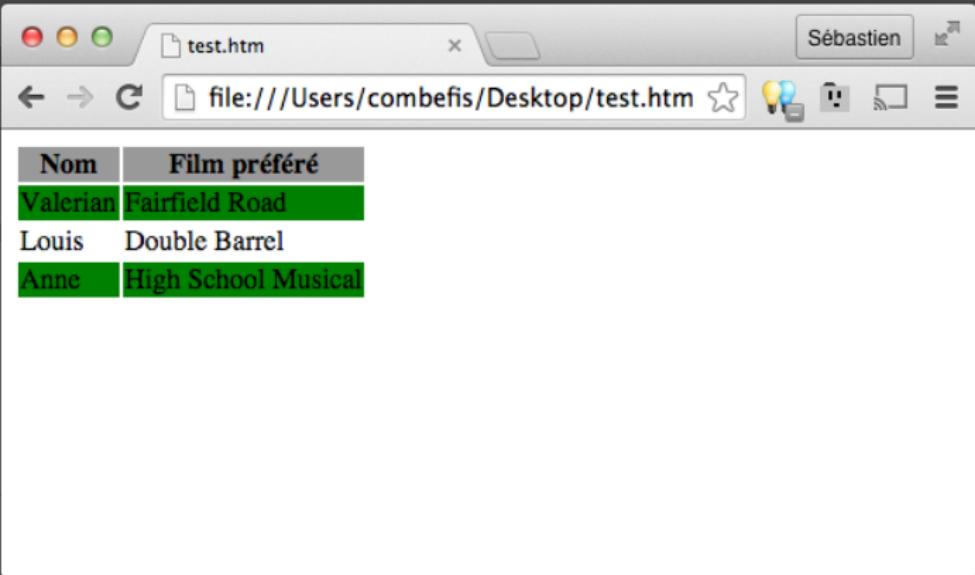
*Par exemple un lien déjà visité, le premier fils...*

- Quelques exemples

- :hover pour la souris par dessus un lien
- :first-child pour le premier élément du type uniquement

```
1 tr:first-child {  
2   background-color: #999;  
3 }  
4 tr:nth-of-type(2n) {  
5   background-color: green;  
6 }
```

# Pseudo-classe



A screenshot of a web browser window titled "test.htm". The address bar shows "file:///Users/combefis/Desktop/test.htm". The browser interface includes standard controls like back, forward, and search, along with developer tools icons.

Nom	Film préféré
Valerian	Fairfield Road
Louis	Double Barrel
Anne	High School Musical

The table rows alternate in background color, with the second and fourth rows having a green background. The browser's status bar at the bottom right shows "Sébastien" and a small icon.

```
1 tr
2
3 }
4 tr:nth-of-type(2n) {
5   background-color: green;
6 }
```

# Pseudo-élément

- Un **pseudo-élément** fait référence à une partie d'un élément

*Par exemple la première lettre, ligne...*

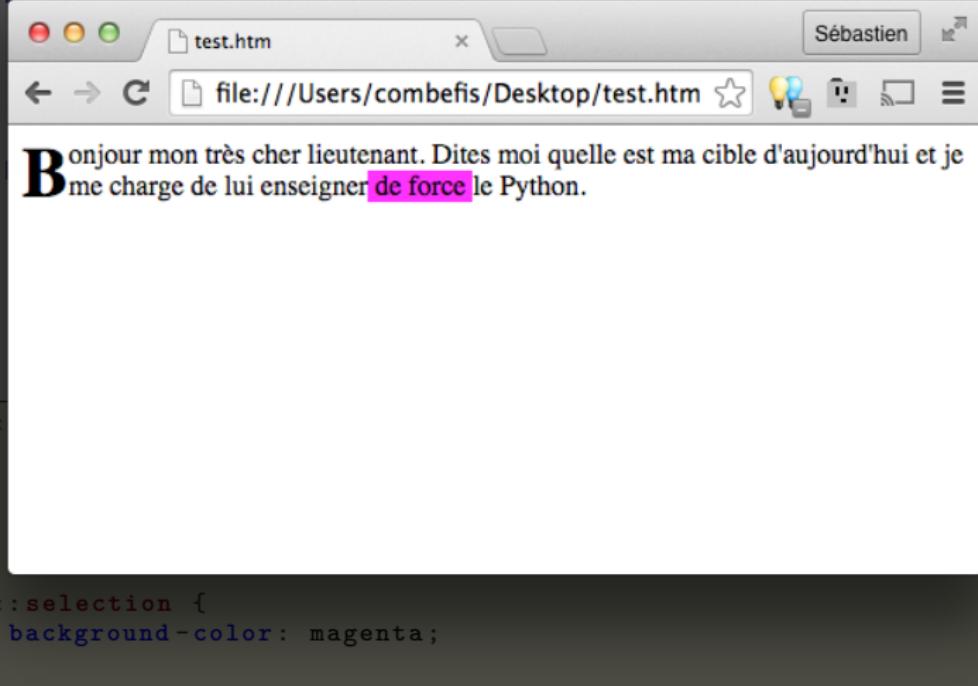
- Quelques exemples

- ::first-letter pour la première lettre
- ::selection pour la partie sélectionnée

```
1 p::first-letter {  
2   margin-top: -3px;  
3   font-size: 30pt;  
4   font-weight: bold;  
5   float: left;  
6 }  
7 p::selection {  
8   background-color: magenta;  
9 }
```

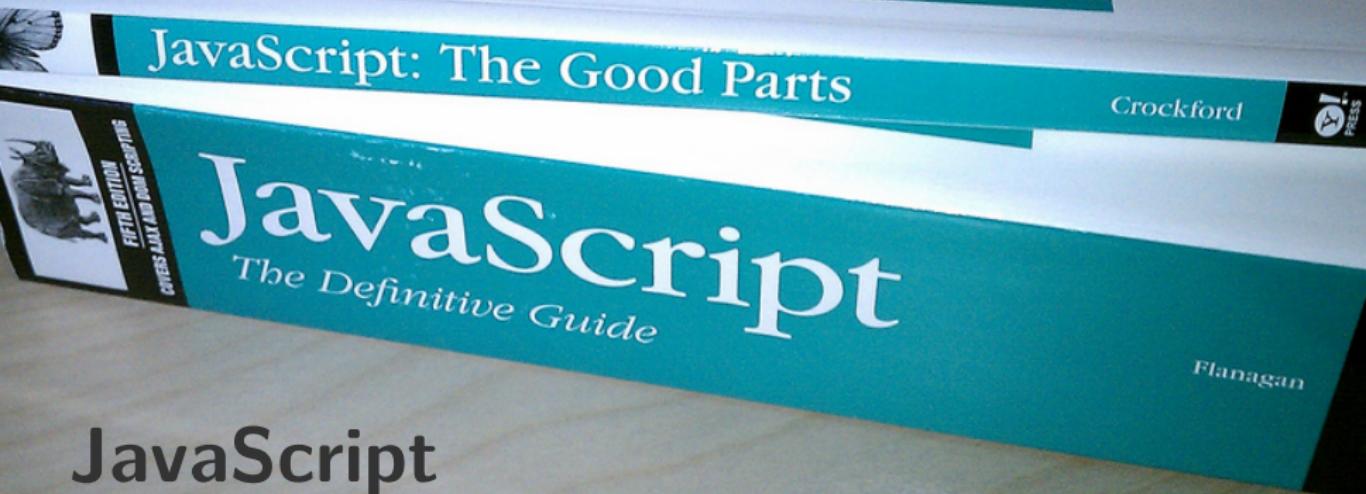
# Pseudo-élément

Un **pseudo-élément** fait référence à une partie d'un élément



A screenshot of a web browser window titled "test.htm". The address bar shows "file:///Users/combefis/Desktop/test.htm". The page content is a single paragraph: "Bonjour mon très cher lieutenant. Dites moi quelle est ma cible d'aujourd'hui et je me charge de lui enseigner de force le Python.". The word "de force" is highlighted with a magenta background, demonstrating the effect of a CSS pseudo-element selector.

```
1 p :  
2  
3  
4  
5  
6 }  
7 p::selection {  
8   background-color: magenta;  
9 }
```



JavaScript

# JavaScript (JS)

- Langage de script de haut niveau, non typé et interprété

*Le JavaScript est exécuté côté client, par le browser*

- Langage orientée objet à prototype
- Bref historique des différentes versions

- Créé en 1995 par Brendan Eich (pour le compte de Netscape)
- Adoption d'un standard **ECMAScript** en juin 1997
- Troisième version d'ECMAScript en 1999 (JavaScript 1.5)

# Lien avec HTML

- Deux manières d'attacher du **code JavaScript**
  - Avec l'élément `script`
  - Dans un fichier séparé, avec l'extension `.js`
- Code depuis des **fichiers externes** avec l'élément `script`

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

# Affichage

- Programme qui affiche un **message de bienvenue**

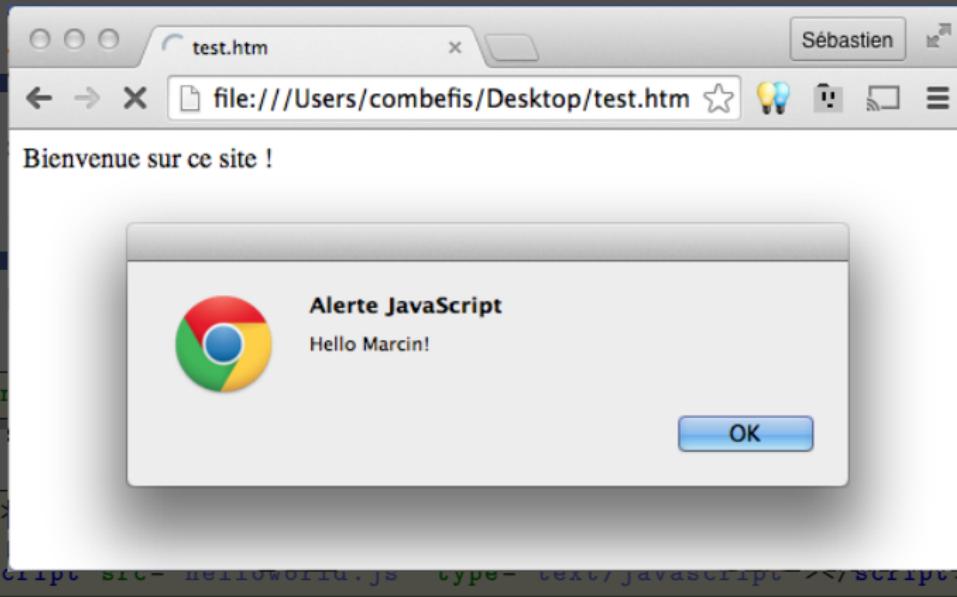
*Code exécuté au moment où celui-ci est chargé par le navigateur*

- **Inclusion du script** à l'endroit où on souhaite l'exécuter

```
1 window.alert('Hello Marcin!');
```

```
1 <p>Bienvenue sur ce site !</p>
2
3 <script src="helloworld.js" type="text/javascript"></script>
```

# Affichage



# Variable et opérations

- Déclaration d'une **variable** avec le mot réservé var

*Les variables possèdent un type dynamique*

- Données de **type** nombre entier, flottant, booléen...

*Une variable peut changer de type en cours d'exécution*

```
1 var max = 20;
2 var score = 7.5;
3 var succeeded = score >= 10;
4 var mean = score / max * 100;
5
6 window.alert(succeeded);           // Affiche false
7 window.alert('Moyenne de ' + mean); // Affiche Moyenne de 37.5
```

# Instruction if-else if-else

- Exécution conditionnelle de blocs avec if-else if-else

*Un bloc de code est délimité avec des accolades*

- Blocs else if et bloc else optionnels

```
1 var max = 20;
2 var score = 7.5;
3 var succeeded = score >= 10;
4 var mean = score / max * 100;
5
6 if (succeeded) {
7     window.alert('Réussi avec une moyenne de ' + mean + '%');
8 } else {
9     window.alert('Dommage, raté :-(');
10 }
```

# Modification de la page

- On peut accéder aux **éléments de la page** pour les modifier

Avec la fonction `document.getElementById`

- On modifie ensuite le contenu avec la propriété `innerHTML`

```
1 <p>Bienvenue sur ce site !</p>
2 <p id="result"></p>
```

```
1 var max = 20;
2 var score = 7.5;
3 var succeeded = score >= 10;
4 var mean = score / max * 100;
5
6 var p = document.getElementById('result');
7 if (succeeded) {
8     p.innerHTML = 'Réussi avec une moyenne de ' + mean + '%';
9 } else {
10     p.innerHTML = 'Dommage, raté :-(';
11 }
```

# Modification de la page

- On peut accéder aux éléments de la page pour les modifier



The screenshot shows a web browser window titled "test.htm" with the URL "file:///Users/combefis/Desktop/test.htm". The browser interface includes standard controls like back, forward, and search, along with developer tools icons. The main content area displays the HTML code and its rendered output. The rendered output consists of two paragraphs: "Bienvenue sur ce site !" and "Dommage, raté :-(". The source code in the left pane shows the following:

```
1 <p>
2 <p>
3 
4 
5 
6 val
7 if (succeeded) {
8     p.innerHTML = 'Réussi avec une moyenne de ' + mean + '%';
9 } else {
10     p.innerHTML = 'Dommage, raté :-(';
11 }
```

# Instruction while

- Répéter un bloc tant qu'une condition est vraie avec while

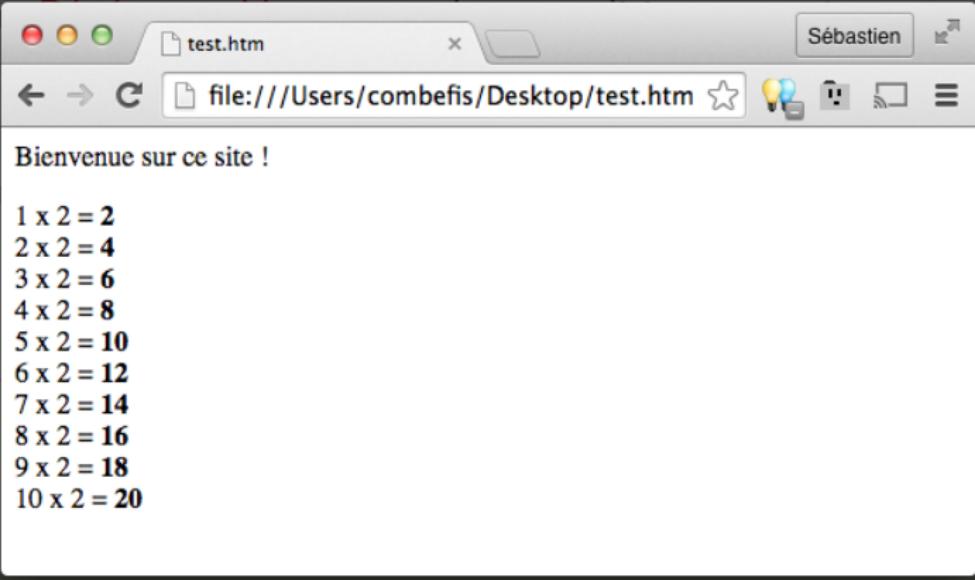
*Attention aux boucles infinies si la condition est toujours vraie*

- Construction du résultat avec une chaîne de caractères

*Génération de contenu HTML depuis le JavaScript*

```
1 var result = '';
2 var i = 1;
3
4 while (i <= 10) {
5     result += i + ' x 2 = <b>' + (i * 2) + '</b><br />';
6     i += 1;
7 }
8 document.getElementById('result').innerHTML = result;
```

# Instruction while



The screenshot shows a web browser window titled "test.htm" with the URL "file:///Users/combefis/Desktop/test.htm". The page content displays the message "Bienvenue sur ce site !" followed by a series of multiplication results:  $1 \times 2 = 2$ ,  $2 \times 2 = 4$ ,  $3 \times 2 = 6$ ,  $4 \times 2 = 8$ ,  $5 \times 2 = 10$ ,  $6 \times 2 = 12$ ,  $7 \times 2 = 14$ ,  $8 \times 2 = 16$ ,  $9 \times 2 = 18$ , and  $10 \times 2 = 20$ . Below the browser window, the corresponding JavaScript code is visible:

```
1 va
2 va
3 wh
4   10 x 2 = 20
5
6
7 }
8 document.getElementById('result').innerHTML = result;
```

# Réaction à un évènement

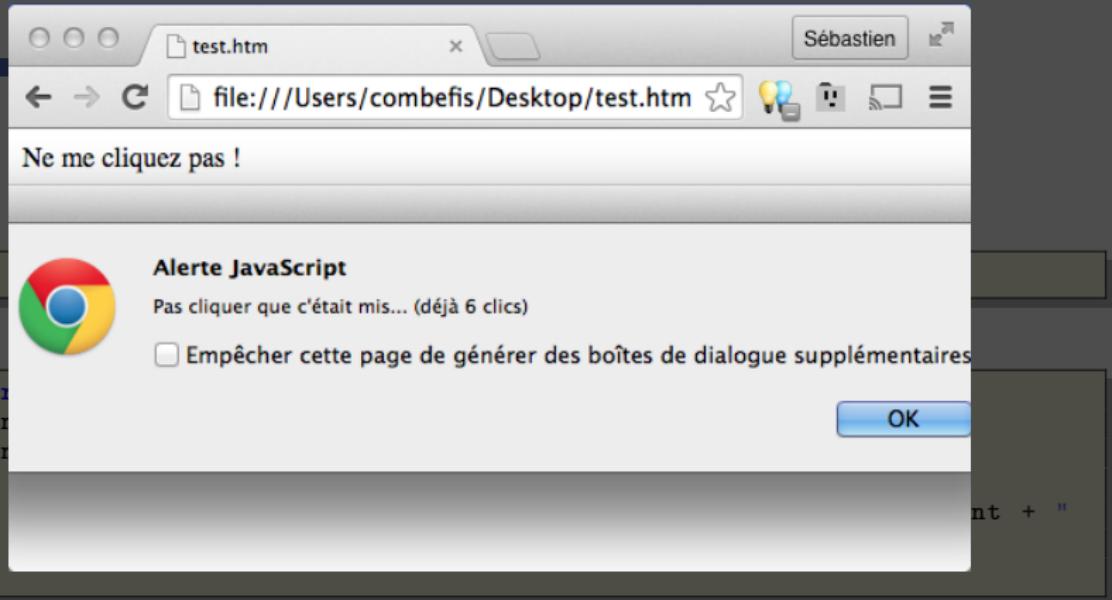
- Association d'une **action à un élément** (une fonction)

*Par exemple, l'évènement onclick*

```
1 <a id="dontclick">Ne me cliquez pas !</p>
```

```
1 var cnt = 0;
2 link = document.getElementById('dontclick');
3 link.onclick = function() {
4     cnt += 1;
5     window.alert("Pas cliquer que c'était mis... (déjà " + cnt + "
6     clics)");
```

# Réaction à un évènement



# Crédits

- <https://www.flickr.com/photos/puliarfanita/3360463235>
- [https://en.wikipedia.org/wiki/File:Paul\\_Otlet\\_%C3%A0\\_son\\_bureau\\_\(cropped\).jpg](https://en.wikipedia.org/wiki/File:Paul_Otlet_%C3%A0_son_bureau_(cropped).jpg)
- [https://en.wikipedia.org/wiki/File:Tim\\_Berners-Lee.jpg](https://en.wikipedia.org/wiki/File:Tim_Berners-Lee.jpg)
- [https://en.wikipedia.org/wiki/File:Robert\\_Cailliau\\_On\\_Desk.jpg](https://en.wikipedia.org/wiki/File:Robert_Cailliau_On_Desk.jpg)
- [https://en.wikipedia.org/wiki/File:Internet\\_Key\\_Layers.png](https://en.wikipedia.org/wiki/File:Internet_Key_Layers.png)
- <https://openclipart.org/detail/180746/tango-computer-green>
- <https://openclipart.org/detail/36565/tango-network-server>
- [https://www.flickr.com/photos/binary\\_koala/7734971772](https://www.flickr.com/photos/binary_koala/7734971772)
- [https://en.wikipedia.org/wiki/File:HTML5\\_logo\\_and\\_wordmark.svg](https://en.wikipedia.org/wiki/File:HTML5_logo_and_wordmark.svg)
- <https://www.flickr.com/photos/miwok/9160033644>
- <https://www.flickr.com/photos/nathansmith/4704268314>