

Interactions homme-machine sur le Web

Modalité de l'interaction homme-machine, évènements, boutons et formulaires

Le Web

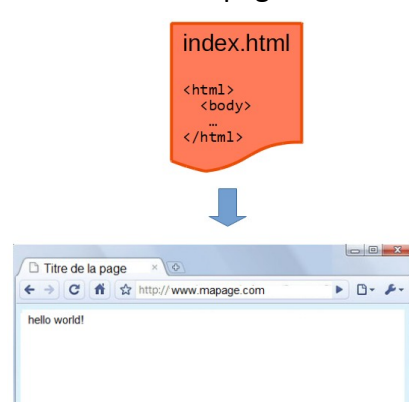
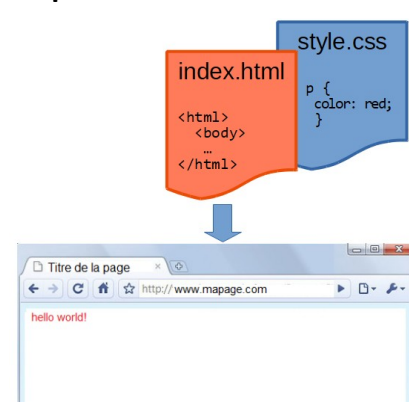
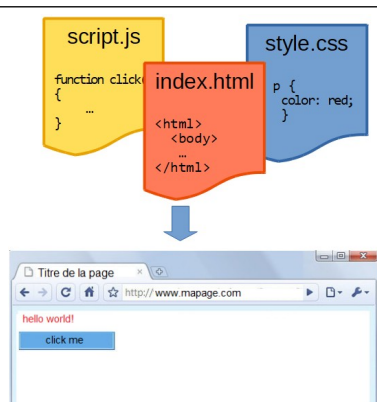
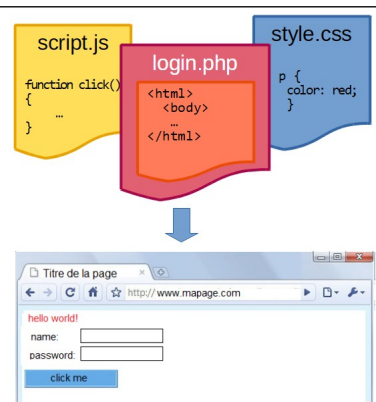
Le **World Wide Web** (littéralement la « toile (d'araignée) mondiale », abrégé **www** ou le **Web**) est inventé en **1989** par **Tim Berners-Lee** et **Robert Cailliau** au CERN. Il permet de consulter dans un navigateur des pages accessibles sur des sites. Le Web n'est qu'une application parmi d'autres fonctionnant sur Internet, comme le courrier électronique, la visioconférence et le partage de fichiers en pair à pair. Attention à ne **as confondre le Web et l'Internet** comme beaucoup de personnes le font!

Le Web repose sur trois technologies :

- Le protocole de transfert **HTTP** (*HyperText Transfer Protocol*) ou **HTTPS** en version sécurisée ;
- le langage **HTML** (*HyperText Markup Language*) ; et
- les **URL** (*Uniform Resource Locator*) couramment appelées adresses web.

Dans sa première version, le Web 1.0, les utilisateurs consultent des **pages statiques**. A partir des années 2000, avec le Web 2.0, les utilisateurs peuvent interagir à travers des **pages dynamiques et interactives** (réseaux sociaux, vidéos en ligne, blogs, etc.). Le Web 3.0 voit la décentralisation des données.

Les principaux langages du Web

<p>HTML est un langage de balisage permettant de structurer le contenu d'une page web.</p>  <pre>index.html <html> <body> ... </html></pre>	<p>CSS (<i>Cascading Styles Sheet</i>) est le langage qui s'occupe de la présentation visuelle d'une page web.</p>  <pre>index.html <html> <body> ... </html> style.css p { color: red; }</pre>
<p>JavaScript et PHP sont des langages de programmation pour créer des pages dynamiques et interactives.</p>  <pre>script.js function click() { ... } index.html <html> <body> ... </html> style.css p { color: red; }</pre> <p>Un programme JavaScript s'exécute dans le navigateur.</p>	 <pre>script.js function click() { ... } login.php <html> <body> ... </html> style.css p { color: red; }</pre> <p>Un programme PHP s'exécute sur le serveur web et génère du code HTML.</p>

HTML

HTML (*HyperText Markup Language*) est un langage de balisage (pas un langage de programmation), permettant d'afficher dans un navigateur des éléments d'une page web statique.

Exemple d'un **élément** HTML : `<p>Mon premier paragraphe en HTML</p>`

Dans cet exemple, `<p>` est la **balise ouvrante** de l'élément ; `</p>` est la **balise fermante** correspondante. Certaines balises n'ont pas de balise fermante, elles sont **orphelines** (par exemple `
` ou ``).

Les éléments peuvent avoir des **attributs** ajoutés dans leur balise ouvrante, sous la forme **attribut="valeur"**, par exemple un attribut href dans une balise `<a>` permet de créer un hyper lien : `Wiki`.

Un document HTML bien formé doit être constitué d'un élément `<html>`, contenant un élément `<head>` avec les caractéristiques générales de la page, et un élément `<body>` avec le corps du document.

Rappel des principales balises :

Balise ouvrante	Balise fermante	Rôle
<code><!DOCTYPE html></code>		Document écrit en HTML
<code><html></code>	<code></html></code>	Zone contenant du code HTML
<code><head></code>	<code></head></code>	En-tête du document
<code><title></code>	<code></title></code>	Titre du document
<code><body></code>	<code></body></code>	Contenu de la page
<code><h1></code> à <code><h6></code>	<code></h1></code> à <code></h6></code>	6 niveaux de titre
<code><p></code>	<code></p></code>	Paragraphe
<code></code> , <code></code> , <code></code>	<code></code> , <code></code> , <code></code>	Liste numérotée, à puce, élément de liste
<code><i></code> , <code></code> , <code><u></code>	<code></i></code> , <code></code> , <code></u></code>	Texte en italique, gras, souligné
<code>
</code>		Retour à la ligne
<code><!--</code>	<code>--></code>	Commentaire
<code></code>	<code></code>	Lien hypertexte
<code></code>		Afficher une image

La plupart des éléments acceptent les attributs `name` (nom de l'élément), `id` (identifiant unique) et `class` (identifiant de groupe).

- 1 Créer un nouveau dossier « mon-site » dans vos Documents et ajouter dans ce dossier un nouveau fichier texte appelé « index.html » contenant le code HTML suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <title>J'apprends HTML</title>
  </head>
  <body>
    <h1>Le trio : HTML, CSS et JavaScript</h1>
    <p>Une page web vraiment simple.</p>
  </body>
</html>
```

- 2 Ouvrir le fichier dans un navigateur. Où s'affiche le texte « J'apprends HTML » de l'élément `<title>` ?

Dans le nom de l'onglet

Le langage CSS (*Cascading Styles Sheet*) permet de mettre en forme le contenu d'une page web.

Exemple d'une **règle** CSS : `p { color: darkblue; font-family: arial; }`

Dans cet exemple, p est le **sélecteur**, il détermine quels éléments seront affectés. Il est suivi entre accolades de **propriétés** de style (par exemple « color ») auxquelles on donne une valeur (par exemple « darkblue »). Les propriétés et leur valeur sont séparées par « : ». Un sélecteur peut cibler :

- tous les éléments d'un même type, par exemple de type <p> : `p { color: darkblue ; }`
- tous les éléments contenant un même attribut class, par exemple `class="important" : .important { border : solid }`
- un élément unique identifié par un attribut, par exemple `id="special" : #special {background-color:yellow }`

On peut aussi combiner les sélecteurs, par exemple attention à ne pas confondre :

- `#special, a {...}` : tous les éléments <a> et l'élément identifié par `id="special"` ;
- `#special a {...}` : tous les éléments <a> à l'intérieur de l'élément identifié par `id="special"` ;
- `p.important a {...}` : tous les éléments <a> à l'intérieur d'un élément <p class="important">

Il existe de nombreuses propriétés CSS par exemple :

Propriété	Description	Exemple
color	Couleur du texte	color: darkblue
background-color	Couleur du fond	background-color: red
border	Épaisseur, style et couleur du bord	border: 2px solid green
width	Largeur de l'élément	width: "50%:"
font	Police de caractère	font: italic 12pt sans-serif

Une feuille de style peut être directement dans un document HTML, à l'intérieur de l'élément <style> placé dans l'élément <head>, ou dans un fichier séparé relié par l'élément <link> :

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

- 3 Modifier le fichier « index.html » avec le code suivant pour ajouter un texte sur fonds rouge :

```
<!DOCTYPE html>
<html>
  <head>
    <title>J'apprends HTML</title>
    <style>
      #avec {
        background-color:red;
      }
    </style>
  </head>
  <body>
    <h1>Le trio : HTML, CSS et JavaScript</h1>
    <p>Une page web vraiment simple <span id="avec">avec CSS</span>.</p> 1
  </body>
</html>
```

- 4 Que faut-il ajouter dans l'élément <style> pour afficher le titre 1 en vert ?

```
h1 { color: green; }
```

¹ Les éléments div et span permettent de structurer le contenu de la page, la balise <div> en créant un bloc sur plusieurs lignes, et en ligne à l'intérieur d'un élément.

JavaScript

JavaScript² est un **langage de programmation** qui permet d'écrire des pages web dynamiques et interactives. A la différence de PHP, il **s'exécute dans le navigateur de l'utilisateur**.

On retrouve les mêmes structures de base dans tous les langages informatiques, mais la syntaxe varie. Comme beaucoup d'autres langages, la syntaxe de JavaScript provient du langage C (avec des spécificités).

Les principaux types de données sont :

Nombres	entiers : 123 et flottants : 123.456
Chaînes	"texte" ou 'texte'
Booléens	true et false
Tableaux	T = [1, 2, 3] T[1] T.length
Dictionnaires	d = {a: 1, b: 2, c: 3} s.a

Et les principales instructions (une différence importante avec Python : les blocs sont délimités par « {...} » les instructions se finissent par « ; ») :

Déclaration	let nomVariable; ou var nomVariable; ³
Affectation	nomVariable = expression;
Instruction conditionnelle	if (expression) {instructions} else if {instructions} (facultatif) else {instructions} (facultatif)
Boucle bornée	for (i = 0; i < 10; i = i + 1) {instructions}
Boucle non bornée	while (expression) {instructions}
Fonction	function nomFonction(paramètres) {instructions}
Commentaires	// commentaire (ne pas confondre avec la division entière Python)
Opérateurs booléens	et : && or : not : !

On peut écrire du code JavaScript directement dans une page HTML à l'intérieur d'un élément `<script>` en fin de document juste avant `</body>`⁴, ou dans un fichier séparé relié par l'attribut `src` :

```
<script src="script.js"></script>
```

- 5 Ajouter le programme JavaScript suivant à l'intérieur d'un élément `<script>` à la fin du fichier « index.html » :

```
// ...
<p>Une page web vraiment simple <span id="avec">avec CSS</span> et JavaScript.</p>
<script>
  alert('hello')
</script>
</body>
</html>
```

- 6 Rafraîchir la page dans le navigateur et vérifier que le message « hello » apparaît dans une fenêtre.

² Il n'a rien à voir avec le langage Java.

³ Les variables qui ne sont pas déclarées avec `let` ou `var` sont des variables globales.

⁴ pour ne pas bloquer le chargement de la page.

Événements

- 7 Supprimer l'élément `<script>` et ajouter ce nouveau paragraphe dans « index.html ».

```
<p onclick="alert('hello')">Cliquez ici</p>
```

- 8 Rafraîchir la page dans le navigateur. Que se passe-t-il quand on clique sur le paragraphe ?

Lemessage hello apparait.

L'élément `<p>` a réagit à l'événement `click`.

- 9 Modifier la balise `<p>` en ajoutant un attribut `onmouseover` :

```
<p onclick="alert('hello')" onmouseover="this.style.color='blue'">Cliquez ici</p>
```

« `this` » designe l'élément concerné, ici l'élément `p`.

« `.style.color` » est une propriété de l'élément.

- 10 Rafraîchir la page dans le navigateur. Que se passe-t-il quand on survole le paragraphe avec la souris ?

Le texte est écrit en bleu.

L'élément `<p>` a réagit à l'événement `mouseover`.

Une autre propriété d'un élément est `innerHTML` qui permet de modifier le texte de cet élément.

- 11 Modifier dans la balise `<p>` l'attribut `onclick` :

```
<p onclick="this.innerHTML='Vous avez cliqué'"  
onmouseover="this.style.color='blue'">Cliquez ici</p>
```

- 12 Rafraîchir la page dans le navigateur. Que se passe-t-il quand on clique sur ce nouveau bouton ?

Le texte « Cliquez ici » est remplacé par « Vous avez cliqué ».

Un événement correspond en général à un **changement d'état** ou à **une intervention de l'utilisateur**, par exemple quand l'utilisateur survole un paragraphe avec la souris ou clique sur un bouton.

La programmation événementielle est un paradigme de programmation dans lequel **l'exécution est déclenchée lorsqu'un événement survient**.

Il existe de nombreux événements qu'on peut « capturer » dans une page HTML, par exemple :

Événements	Changement d'état ou action de l'utilisateur
<code>mouseover</code> , <code>mouseout</code>	Quand la souris passe sur l'élément, sort de l'élément.
<code>mouseup</code> , <code>mousemove</code>	Quand le bouton de la souris est relâché, quand la souris bouge.
<code>click</code> , <code>dblclick</code>	Quand l'utilisateur clique, double-clique.
<code>focus</code> , <code>blur</code>	Quand un élément gagne le focus, perd le focus.
<code>submit</code>	Quand un formulaire est envoyé.

On capture un événement dans un élément HTML en ajoutant un attribut formé de `on` suivi du nom de l'événement, par exemple l'attribut `onclick` permet de capturer l'événement `click`.

- 13 Ajouter des attributs à la balise `<p>` pour qu'il affiche 'Vous avez double-cliqué' quand l'utilisateur double-clique et à nouveau 'Cliquez ici' quand la souris quitte l'élément.

```
<p onclick="this.innerHTML='Vous avez cliqué'"  
ondblclick="this.innerHTML='Vous avez double-cliqué'"  
onmouseout="this.innerHTML='Cliquez ici'">Cliquez ici</p>
```

Boutons

Les **interactions entre l'homme et la machine** peuvent prendre de nombreuses formes sur le Web : cliquer sur un lien pour charger une page web, saisir du texte dans une zone de texte, etc. et les nombreux éléments HTML permettent de configurer toutes ces formes d'interactions, en particulier les **boutons**.

- 14 Ajouter la ligne suivante dans « index.html ».

```
<button>Cliquez ici</button>
```

- 15 Rafraîchir la page dans le navigateur. Que se passe-t-il quand on clique sur le bouton ?

Rien du tout

- 16 Ajouter des attributs à ce bouton pour que le texte devienne 'Vous avez cliqué' quand l'utilisateur clique sur le bouton, 'Vous avez double-cliqué' quand il double-clique et à nouveau 'Cliquez ici' quand sa souris sort de l'élément.

```
<button onclick="this.innerHTML='Vous avez cliqué'"
  ondblclick="this.innerHTML='Vous avez double-cliqué'"
  onmouseout="this.innerHTML='Cliquez ici'">Cliquez ici</button>
```

Avec JavaScript, on peut programmer des fonctions qui sont appelées quand un événement se produit.

- 17 Ajouter un nouveau bouton dans « index.html ».

```
<button id="monBouton" onclick="monClick()">Cliquez ici</button>
```

- 18 Rafraîchir la page dans le navigateur. Que se passe-t-il quand on clique sur le bouton ?

Rien du tout

- 19 Il faut programmer la fonction ! Ajouter le programme JavaScript suivant à l'intérieur d'un élément `<script>` à la fin du fichier « index.html » :

```
// ...
<script>
function monClick() {
    document.getElementById("monBouton").innerHTML="Vous avez cliqué";
    document.getElementById("monBouton").style.color="red";
}
</script>
</body>
</html>
```

- 20 Rafraîchir la page dans le navigateur. Que se passe-t-il maintenant quand on clique sur le bouton ?

Le texte affiché dans le bouton a été modifié en « Vous avez cliqué » affiché en rouge

- 21 Modifier le programme JavaScript pour que le texte 'Allez-y' apparaisse en bleu quand la souris passe sur le texte et 'Cliquez ici' apparaisse à nouveau en noir quand la souris sort de l'élément.

```
<button id="monBouton" onmouseover="allezy()"
  onclick="monClick()" onmouseout="cliquezici()">Cliquez ici</button>
<script>
function allezy() {
    document.getElementById("monBouton").innerHTML="Allez-y";
    document.getElementById("monBouton").style.color="blue";
}
function cliquezici() {
    document.getElementById("monBouton").innerHTML="Cliquez ici";
    document.getElementById("monBouton").style.color="black";
} ...
```

Formulaires

Les **formulaires** sont un autre type d'interactions entre l'homme et la machine sur le Web très courant. Un formulaire s'écrit avec des éléments de type `<input>` placés entre les balises `<form>` et `</form>`.

22 Ajouter les lignes suivantes dans l'élément `<body>` :

```
<form>
  Nom: <input type="text" name="nom">
  Prénom: <input type="text" name="prenom">
  <input type="button" value="Envoyer" onclick="bonjour(this.form)">
</form>
```

Éléments `<input>` de type "text"
Élément `<input>` de type "button" qui appelle la fonction `bonjour()`

et la fonction `bonjour(f)` dans l'élément `<script>` :

```
function bonjour(f) {
  alert('Bonjour ' + f.prenom.value + ' ' + f.nom.value );
}
```

23 Comment s'affiche ce formulaire dans la page ?

Nom Prénom

24 Entrer le nom et prénom d'Alan Turing. Que se passe-t-il quand on clique sur le bouton Envoyer?

Une fenêtre d'alerte s'ouvre avec « Bonjour Alan Turing »

Noter que l'élément `<input>` est une balise orpheline. L'attribut **type** permet de préciser le format : :

Type	Exemples	Apparence
text	Nom : <code><input type="text" name="nom"></code>	Nom: <input type="text" value="Turing"/>
password	Mot de Passe : <code><input type="password" name="mdp"></code>	Mot de Passe : <input type="password" value="....."/>
number	Nombre : <code><input type="number" name="n"></code>	Nombre : <input type="number" value="42"/>
checkbox (cases à cocher)	Case 1 <code><input type="checkbox" name="cc" value="0"></code> Case 2 <code><input type="checkbox" name="cc" value="1"></code> Case 3 <code><input type="checkbox" name="cc" value="2"></code>	Case 1 <input type="checkbox"/> Case 2 <input type="checkbox"/> Case 3 <input type="checkbox"/>
radio (choix exclusif)	Choix 0 <code><input type="radio" name="bb" value="0"></code> Choix 1 <code><input type="radio" name="bb" value="1"></code> Choix 2 <code><input type="radio" name="bb" value="2"></code>	Choix 0 <input type="radio"/> Choix 1 <input type="radio"/> Choix 2 <input type="radio"/>
submit ou bouton ⁵	<code><input type="submit" value="Envoyer" onclick="MaFonction()></code> <code><input type="button" value="Envoyer" onclick="MaFonction()></code>	<input type="button" value="Envoyer"/>

25 Modifier le formulaire pour demander à l'utilisateur de préciser son sexe (Homme ou Femme, un seul choix possible) et qu'il affiche « Bonjour Monsieur Alan Turing » ou « Bonjour Madame ... » en fonction.

Nom: Prénom:

Sexe : Femme Homme

```
...
Femme <input type="radio" name="sexe" value="F">
Homme <input type="radio" name="sexe" value="H">
...
if (f.sexe.value == 'H') {alert('Bonjour Monsieur ' + f.prenom.value + ' ' + f.nom.value )}
else {alert('Bonjour Madame ' + f.prenom.value + ' ' + f.nom.value )};
```

⁵ On peut aussi utiliser un élément bouton : `<button type="submit" onclick="MaFonction()">Envoyer</button>`

Séparer les codes dans des fichiers externes

Une bonne pratique pour avoir un code plus lisible et un affichage plus rapide est de **séparer les codes CSS et JavaScript dans des fichiers externes**.



On indique dans le fichier HTML l'emplacement du code CSS et programme Javascript par :

```
<link rel="stylesheet" href="style.css"> (dans l'élément <head> ) et  
<script src="script.js"></script> (juste avant la balise fermante </body> ).
```

- 26 Créer dans le dossier « mon-site » deux nouveaux fichiers textes « style.css » et « script.js », puis transférer les parties de code des éléments `<style>` et `<script>` depuis le fichier « index.html » dans ces deux fichiers et rafraîchir la page dans le navigateur et vérifier qu'elle s'affiche toujours de la même façon.

Mini-projets

Les navigateurs web sont très permissifs et font de leur mieux pour afficher un code HTML sans message d'erreur. Pour mettre au point un code qui n'affiche pas ce que l'on attend, on peut utiliser la fonction **console.log()** qui permet d'afficher des messages dans la console du navigateur (F12) .

- 27 Créer un formulaire qui demande deux nombres et affiche leur somme.

```
<form>  
  Nombre 1: <input name="n1"> <br>      Nombre 2 : <input name="n2"> <br>  
  <input type="button" value="Somme" onclick="somme(this.form)">  
</form>  
function somme(f) {  
  alert(parseInt(f.n1.value) + parseInt(f.n2.value));  
}
```

- 28 Compléter le code suivant d'un jeu de « plus ou moins » qui consiste à deviner un nombre mystère..

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
    <title>Jeu du plus ou moins</title>  
  </head>  
  <body>  
    <form>  
      Entrez un nombre entre 0 et 99 <input type="number" name="n">  
      <input type="button" value="Envoyer" onclick="plusoumoins(this.form)">  
    </form>  
    <script>  
      let mystere = Math.floor(Math.random() * 100); // nombre secret aléatoire entre 0 et 99  
      function plusoumoins(f) {  
        let essai = f.n.value  
        if (essai > mystere) {alert('plus petit');}  
        else if (essai < mystere) {alert('plus grand');}  
        else {alert('gagné'); mystere = Math.floor(Math.random() * 100);}  
      }  
    </script>
```



- 29 Écrire un jeu simple de pierre-feuille-ciseaux. On peut utiliser des éléments `<button>` pour le joueur et le bot avec le code CSS suivant :

```
button {  
  width: 100px;  
  height: 60px;  
  font-size: 20px;  
  background-color:white;  
}
```

puis `document.getElementById('xxx').style.backgroundColor = "grey";` pour modifier la propriété en Javascript.

