# Interactions entre l'homme et la machine sur le Web

Lorsqu'on navigue sur le Web nous ne faisons pas que lire passivement du texte mais parfois agissons sur le document que nous lisons. Regardons sur un site de réservation en ligne d'un billet de TGV

# 1 Les différents composants graphiques permettant d'interagir avec une application Web

Aller simple      Aller-retour		<mark>-30%</mark> .	Acheter un	e carte ou un abor	inement >
PARIS MASSY TGV	~	Aller 10/09/2019	21	A partir de <mark>06h</mark>	$\bigcirc$
Arrivée	~	Retour	21	A partir de <mark>06h</mark>	$\bigcirc$
<ul> <li>□ Trajets directs</li> <li>● 2ème classe</li> <li>○ 1ère classe</li> <li>Nombre de voyageurs</li> <li>■ 1 + de 9 voyageurs, jusqu'à -65%<sup>(2)</sup></li> </ul>				الج Voyageur	handicapé
VOYAGEUR 1 26-59 ans ✓ Sans carte de réduction ✓	Sans programme de fidéli	té 🗸	Code ava	ntage ou Bon d'a	chat
Horaires seuls	: <b>hercher</b> + de	e critères			^

La plupart du temps nous retrouvons un peu partout les même éléments :

- 1. des cases à cocher comme celle associée à Trajets directs. En <u>cliquant</u> à l'intérieur de cette case on peut voir une encoche apparaître  $\lor$  ce qui signifie que ce choix Trajets directs a été sélectionné
- 2. des boutons radio .

En <u>cliquant</u> dans ces boutons ils deviennent colorés ce qui permet de faire un choix, par exemple sur le site de réservation on peut ainsi choisir un direct ou un aller retour

- 3. des menus déroulants qui nous propose des choix que l'on peut sélectionner ainsi la date du départ est un menu déroulant ainsi que le lieu du départ.
- 4. des curseurs. Si on <u>clique</u> sur le symbole  $\oplus$  on peut augmenter d'une unité le nombre de voyageurs <u>alors</u> que si on <u>clique</u> sur le symbole  $\oplus$  on diminue d'une unité le nombre de voyageurs
- 5. **des boutons** comme le bouton **Rechercher** qui lance la recherche d'un voyage avec les critères sélectionnés à condition que l'on clique dessus

Le caractère interactif de la page est définie soit au niveau du **serveur** avec un langage comme PHP, ou au niveau du **client** avec un langage comme JavaScript

## 2 Interaction client-serveur



La plupart du temps quand on navigue sur le Web de lien en lien **en tant que client** on sollicite une information (texte, image, vidéo, son) qui se trouve sur un **serveur** distant suivant un **protocole** de communication nommé **HTTP** pour (Hyper Text Transfert Protocol) ou encore HTTPS (S pour Secure) lorsque les données sont chiffrées si on fait des achats en ligne par exemple

Si j'utilise un navigateur (Firefox par exemple) je peux entrer l'URL (pour Uniform Resource Locator) de la page d'un site que je connais sans forcément passer par un **moteur de recherche** 



Une fois la page chargée, je peux avec l'aide de la console Web avoir des informations réseau sur le chargement de la page On voit que :

État	Mé	Domaine Fichier	Source Type	Transfert	Taille 0 ms	160 ms 320 r	En-têtes Cookies Paramètres
200	GET	💋 www.ma /	document html	1,24 Ko	2,07 <b></b> 14 ms		URL de la requête : http://www.mathly.fr/ Méthode de la requête : GET Adresse distante : 85.236.158.88:80 Code d'état : 200 0K ⑦ Version : HTTP/1.1 Modifier et renvoyer
							∀ Filtrer les en-têtes
Ō	1 requ	ête 2,07 Ko / 1,24 Ko transférés Terminé e	n : 511 ms DOMCo	ontentLoaded:	49 ms   load: 463 n	ns	<ul> <li>En-têtes de la réponse (214 o)</li> </ul>

- 1. La méthode de la requête est GET
- 2. Le code est 200 (la page a bien été chargée) (un code d'erreur connu est le fameux 404)
- 3. On a l'adresse IP du serveur distant

Pour être plus précis nous utilisons le logiciel Wireshark qui permet d'analyser les paquets d'information Voici la demande (frame 284) par le client 192.168.1.21 au serveur 85.236.158.88 de la page d'accueil du site mathly.fr

Quelques millièmes de seconde après (frame 286) arrive la réponse Avec Wireshark on

► 284	1 37 <b>.</b> 31	5619 192.168.	1.21 85.236.158	B.88 HTTP	414 GET / HTTP/1.3	1	
<ul> <li>Frame</li> <li>Ether</li> <li>Inter</li> <li>Trans</li> </ul>	Frame 284: 414 bytes on wire (3312 bits), 414 bytes captured (3312 bits) on interface Ø Ethernet II, Src: Apple_18:68:79 (a0:99:9b:18:68:79), Dst: Sagemcom_3e:ba:9e (40:c7:29:3e:ba:9e) Internet Protocol Version 4, Src: 192.168.1.21, Dst: 85.236.158.88 Transmission Control Protocol, Src Port: 62750, Dst Port: 80, Seq: 1, Ack: 1, Len: 348						
Hyper	text T т / нтт	ransfer Protocol					
Ho	st: www	/.mathly.fr\r\n					
Us	er-Ager	it: Mozilla/5.0 (N	lacintosh; Intel Mac OS	X 10.14; rv:68.0) G	ecko/20100101 Firef	ox/68.0∖r∖n	
AC AC	cept: 1 cept-La	ext/html,application inguage: fr.fr–FR:	<pre>ion/xhtml+xml,applicat: a=0.8.en-US:a=0.5.en:a</pre>	10n/xml;q=0.9,*/*;q= =0.3\r\n	0.8\r\n		
Ac	cept-Er	coding: gzip, det	late\r\n				
Co	nnectio	on: keep-alive\r\r	1				
	28	4 37.315619	192.168.1.21	85.236.158.			
	28	5 37.321841	85.230.158.88 95.226.159.99	192.108.1.2			
	20	7 37 416614	192 168 1 21	85 236 158	88 TCP		
	- 20	- 20C+ 1224 h	192110011121	bite) 1224 but			
	Fram Etho	e 200: 1334 Dy rnet II Src:	Sagemeen 3e:ba:0f	(10.07:20.3e)	of) Det: Apple		
	Inte	rnet Protocol	Version 4. Src: 85	-236-158-88- Dst	: 192.168.1.21		
	Tran	smission Contr	ol Protocol. Src P	ort: 80. Dst Por	t: 62750. Seq:		
	Нуре	rtext Transfer	Protocol		, , ,		
	► HT	TP/1.1 200 OK	\r\n				
	Сс	ontent-Type: t	ext/html; charset=l	JTF−8\r\n			
	Content-Length: 1054\r\n						
	Content-Encoding: gzip\r\n						
	Vary: Accept-Encoding,User-Agent\r\n						
	Date: Sun, 18 Aug 2019 20:05:49 GMI\r\n Server: LiteSpeed\r\n						
	Server: Litespeeu(r)n						
	\r\n						
	() [H	TTP response	1/1]				
	[]	ime since rea	uest: 0.100921000 s	seconds]			
	-			-			

peut observer que les éléments d'une page arrivent séparément ainsi le fichier css arrive après (certes quelques centièmes de seconde)

Ces éléments d'information sont ensuite "lus" par le navigateur du client et l'utilisateur peut voir la page Web demandée sur son écran

Sur des nombreuses pages on nous demande de faire des **recherches** ou des **choix** Ceci se fait par l'intermédiaire de **formulaires** qui nous permettent d'affiner nos requêtes

Le site okkazeo.com est un site de vente de jeux de société neufs ou d'occasion.

Entrons directement dans l'URL d'un navigateur https://www.okkazeao.com Sur la page d'accueil vous pouvez remplir un formulaire pour préciser votre recherche

LOCALISATION					
Choisissez un département	▼				
Code postal	1 km 🔻				
CATÉGORIES					
Jeux / Extensions					
Goodies, exclus KS					
Accessoires, pièces, impressions 3D					
ANNONCES					
Occasions					
Neufs					
Choisissez une langue	▼				
PRIX					
Min € Max €					
Filtrer					

On va sélectionner par exemple le département 91, puis cocher les jeux d'occasion et entrer une fourchette de prix entre 20 euros et 50 euros puis cliquer sur le bouton Filtrer

LOCALISATION					
91 - Essonne 💌					
Code postal 1 km 💌					
CATÉGORIES					
Jeux / Extensions					
Goodies, exclus KS					
Accessoires, pièces, impressions	3D				
ANNONCES					
✓ Occasions					
Neufs					
Choisissez une langue 🔹					
PRIX					
20 € 50 €					
<b>▼</b> Filtrer					

On observe alors dans l'url du navigateur :

https://www.okkazeo.com/jeux/arrivages?departement=91&FiltreCodePostal= &FiltreDistance=1&FiltreRechOccasion=on&FiltreLangue=0&FiltreRechPrixMin= 10&FiltreRechPrixMax=50

La méthode GET ajoute des paramètres dans l'URL à la suite du nom

#### de domaine et du chemin vers le fichier après le symbole?

S'il y a plusieurs paramètres ils sont rassemblés avec le symbole & (pour ET) En conclusion on retiendra la structure de l'URL d'une page



## 3 La méthode POST

Sur de nombreux sites on peut se connecter par exemple pour laisser un commentaire sur un site d'informations ou pour pouvoir y programmer comme sur le site de **France I.O.I** à l'aide **d'un formulaire** 

On voit ci-dessous le formulaire à l'écran et le code de la page. Un formulaire est codé par les balises <form> et </form>.

Or pour des raisons de confidentialité on n'a pas envie que l'identifiant et le mot de passe soit visible dans l'URL. (De même lorsqu'on fait un achat en ligne)

Il existe alors une autre méthode pour la transmission des informations, la méthode POST, qui n'affiche pas dans l'URL les paramètres d'une requête.



Vous pouvez tester la méthode POST, tout simplement en vous connectant à l'E.N.T avec votre identifiant et votre mot de passe et en observant l'URL de votre navigateur.

## 4 Les bases du HTML et du CSS

Ce qui suit est un exposé "simple" de quelques éléments des langages HTML et CSS Pour avoir des informations supplémentaires on peut consulter le site de la fondation Mozilla https://developer.mozilla.org/fr/docs/Apprendre

Il faut distinguer le fond ou la structure de la page de la forme (ou la mise en page) de la page

On peut regarder une page Web comme la Une d'un journal papier : certains éléments forment la structure de la page :

- 1. Titre ou en-tête
- 2. Menu
- 3. Article
- 4. Image
- 5. paragraphe(s)

Une mise en page possible peut ressembler à

Pa	ge				
Tit	tre				
M	enu				
۸ <i>.</i>	ticlo				
Ar	ticle				
Image	paragraphe				
paragraphe					

Mais d'autres mises en page sont possibles, comme :

Page	
Titre	
Menu	
Article	
Image	
paragraphe	

- 1. D'où un langage, le langage HTML (pour HyperText Markup Language), pour décrire la structure de la page
- 2. Et un deuxième langage, le langage CSS (pour Cascading Style Sheets), pour décrire la mise en page de la page

Voici la copie d'écran d'une page Web très simple

# **Comment écrire une page Web ?**

HTML

<u>CSS</u>

## Codage d'une page Web



Deux langages sont nécessaires pour écrire une page Web

- 1. Le langage HTML (pour HyperText Markup Language), pour décrire la structure de la page
- 2. Le langage CSS (pour Cascading Style Sheet), pour décrire la mise en page de la page

Voici le contenu du fichier exemple.html qui a donné la page ci-dessus

```
1
      <!DOCTYPE html >
 2 🔻
      <html lang="fr">
3 🔻
         <head>
             <meta charset="utf-8" />
 4
            <!--<link href="feuilleSty.css" rel="stylesheet"/>-->
 5
 6
            <title>HTML et CSS</title>
 7 🖿
         </head>
 8
9 🔻
         <body>
10 🔻
          <header>
11
               <h1>Comment écrire une page Web ?</h1>
12 🛏
          </header>
13
              <nav>
14 🔻
15
16
                   <a href="html.html">HTML</a>
                   <a href="css.html">CSS</a>
17
18
              </nav>
19 -
20
             <article>
21 🔻
22
                <h2>Codage d'une page Web</h2>
                <img width = "450" align = "middle" src="images/html.png"/>
23
24
                 >Deux langages sont nécessaires pour écrire une page Web
25 💌
                 <01>
26 🔻
                     Le langage HTML (pour HyperText Markup Language),
27 ⊾
                     pour décrire la structure de la page
28 🔻
                     Le langage CSS (pour Cascading Style Sheet),
29 ⊾
                     pour décrire la mise en page de la page
30 🛏
                 31 🛏
              </article>
32 🛏
        </body>
33 ⊾ </html>
```

- 1. Le langage HTML est un langage à **balises**. La plupart des balises sont ouvrantes et fermantes comme des parenthèses, par exemple :
  - (a) Tout le document est encadré par <html>...</html> (ligne2 et ligne 33)
  - (b) Les balises <header>.....</header> (ligne 10 et 12) encadrent tout ce qui concerne l' en-tête de la page
  - (c) Les balises <article>.....</articles> encadrent tout ce qui concerne le coeur de la page
- 2. Une balise a des paramètres ou descripteurs ainsi la balise <html> a pour paramètre lang="fr"
- 3. Il existe une relation de **parentalité** entre les balises. On dit par exemple que la balise <html> est le parent de la balise <body> ou encore que la balise <body> est un enfant de la balise <html>, parce que la balise <html> "encadre" la balise <body>

ce qui permet de dessiner une sorte d'arbre généalogique ou DOM (pour Document Object Model) du document HTML, dont voici une version incomplète

4. La balise <head>...</head> donne des informations sur le document , ainsi l'encodage des caractères du document est fait en utf-8. On remarque au passage la présence d'une des rares balises n'ayant qu'un seul terme, la balise meta



La balise <title>...</title> contrairement à ce que l'on pourrait croire ne concerne pas le titre visible de la page mais le titre de l'onglet associé à la page

- 5. La balise <body>...</body> délimite tout ce qui est visible dans la page
- 6. Une balise très importante <a>....</a> crée des **liens** entre les pages et le paramètre href ici donne la référence **relative** (on reviendra plus tard sur cette notion) de la page.

Lorsqu'on déplace le curseur de la souris sur une page Web, s'il y a un lien la "flèche" se transforme en une "main avec un index pointé" ce qui signifie que l'on peut cliquer à cet endroit pour visualiser une autre page Web

- 7. Dans la structure article on observe un titre de niveau 2 avec la balise <h2>
- 8. La balise <img /> est une des rares balises à être seule, elle a plusieurs paramètres comme src (pour source) pour indiquer où se trouve l'image
- 9. La balise (pour ordered list) permet de faire des listes numérotées. Chaque élément de la liste est encadrée par les balises <item> et </item>
- 10. La balise <link href="feuilleSty.css"/> permet de relier au fichier exemple.html un fichier CSS feuilleSty.css qui "mettra en forme" le fichier précédent Il faut "décommenter" la ligne où se trouve la balise <link/> pour la rendre active, c'est à dire enlever les caractères <!- et ->

Ce que fait le fichier feuilleSty.css pour l'instant c'est de mettre en évidence d'abord les trois structures descendantes de la balise body, en mettant une bordure bleue autour de l'en-tête, une verte autour du menu et une rouge autour de l'article

# **Comment écrire une page Web ?**

<u>HTML</u>

<u>CSS</u>

## Codage d'une page Web

```
<!DOCTYPE html >
1
2 💌
      <html lang="fr">
3 🔻
         <head>
            <meta charset="utf-8" />
4
           <!--<link href="feuilleSty.css" rel="stylesheet"/>-->
5
6
           <title>HTML et CSS</title>
7 ⊾
         </head>
8
9 🔻
         <body>
10 🔻
         <header>
              <h1>Comment écrire une page Web ?</h1>
11
12 🛏
         </header>
13
14 🔻
             <nav>
15
                  <a href="html.html">HTML</a>
16
17
                  <a href="css.html">CSS</a>
18
19 🛏
             </nav>
20
21 🔻
            <article>
22
               <h2>Codage d'une page Web</h2>
                <img width = "450" align = "middle" src="images/html.png"/>
23
24
                >Deux langages sont nécessaires pour écrire une page Web
25 💌
                <01>
                    Le langage HTML (pour HyperText Markup Language),
26 🔻
27 🛏
                     pour décrire la structure de la page
28 🔻
                    Le langage CSS (pour Cascading Style Sheet),
29 🛏
                    pour décrire la mise en page de la page
30 🛏
                </01>
31 🛏
             </article>
32 🛏
        </body>
33 ► </html>
```

Deux langages sont nécessaires pour écrire une page Web

- 1. Le langage HTML (pour HyperText Markup Language), pour décrire la structure de la page
- 2. Le langage CSS (pour Cascading Style Sheet), pour décrire la mise en page de la page

Voici le code CSS qui a permis cela :



On constate qu'un fichier CSS consiste à donner des propriétés de mise en forme aux balises par exemple

header  $\{$ 

border :2px solid blue;

}

met une bordure bleue de 2 pixels autour de l'en-tête de la page

Les "structures" sont alignées sur le bord droit de la fenêtre on va "centrer" les structures header, nav et article en les décalant du bord

On va utiliser la propriété **padding** voir https://developer.mozilla.org/fr/docs/ Web/CSS/padding-left

On va aussi mettre de la couleur de fond aux structures avec la propriété background-color pour finalement obtenir une page :

On peut avec le même fichier HTML avoir deux fichiers CSS différents et avoir deux mises en forme différentes

## Comment écrire une page Web ?

HTML

<u>CSS</u>

#### Codage d'une page Web



Le langage HTML (pour HyperText Markup Language), pour décrire la structure de la page
 Le langage CSS (pour Cascading Style Sheet), pour décrire la mise en page de la page



#### 4.1 Exercices : HTML et CSS

Ex 1

Compléter le DOM du fichier exemple.html

Ex 2

La structure de la page html.html est quasiment identique à celle de exemple.html avec quand même quelques modifications

- 1. Pour pouvoir revenir vers la page exemple.html insérer un lien dans le menu autour du mot Accueil
- 2. On suppose que l'on a une icône accueil.png représentant une maison. Faire un lien autour de l'icône permettant de revenir vers la page exemple.html

#### Ex 3

Essayer d'écrire la page css.html sans regarder le cours ni la documentation Ex 4

On veut mettre en évidence les mots "structures" et "mise en page" dans l'article. Allez chercher dans le Web des informations sur une balise qui met en gras le texte encadré, et sur une autre balise qui souligne le texte encadré

Réécrire les paragraphes en question en utilisant ces balises

#### Ex 5

On considère maintenant que la mise en évidence d'une partie d'un texte n'est pas le fait du langage HTML mais du langage CSS

On va donc créer une balise <div class="important">...</div> avec lesquelles on va encadrer du texte que l'on veut mettre en valeur

- 1. Encadrer les mots "structures" et "mise en page" dans l'article avec cette balise
- 2. Dans le fichier CSS associé au fichier exemple.html ajouter .important{ color :#FF0000; }

Allez chercher dans le Web comment mettre en gras et/ou surligner avec CSS

#### **Ex 6**

Dans le menu les éléments sur lesquels il y a un lien sont soulignés et sont en bleu. On aimerait changer leur couleur et qu'ils ne soient pas soulignés. Allez voir sur le Web comment faire cela

## 5 Sélecteurs dans le DOM

Une page web est **structurée** par le langage HTML qui est un langage de balises La plupart des balises sont ouvrantes et fermantes ce qui induit une relation hiérarchique entre les balises, que l'on peut matérialiser par un arbre appelé le DOM (document object model)

Voici la partie visible d'une page "simple", entre les balises <body> et </body>



Les **sélecteurs** sont les éléments de cet ordre , il y a trois types de sélecteurs mais on ne travaillera qu'avec deux types

- 1. Le premier type est formé par toutes les balises d'origine du HTML comme <a>, , etc...
- 2. Le deuxième type est formé par les balises que l'on a **particularisé** en leur donnant un nom avec l'attribut class, par exemple la balise a été particularisé par l'attribut class = 'legende' par conséquent le sélecteur n'est pas p mais .legende

## 6 Association sélecteur - variable Javascript

Si on veut rendre la page interactive et sensible à un évènement comme le clic de souris on associe au sélecteur désignant l'endroit de la page que l'on veut rendre interactive, à une variable Javascript dans un premier temps

#### Exemples :

```
var btn = document.querySelector('button');
var img = document.querySelector('.image');
```

## 7 Ecouteur d'évènement

Ensuite on associe à ces variables Javascript des écouteurs d'évènements qui sont des fonctions qui vont s'exécuter lorsque l'évènement se produira à l'endroit du DOM représenté par le sélecteur

```
btn.addEventListener('click', changer);
img.addEventListener('dblclick', change_image);
```

Ce qui signifie que si on clique une fois sur le bouton où il y a écrit "Changer" la fonction changer sera exécutée et si on double -clique sur l'image de la page alors la fonction change\_image sera exécutée

```
Voici les fonctions en question dans le script javascript
```

```
function changer(){
    let de = Math.floor(Math.random() * images.length) ;
    legende.textContent = legendes[de];
    img.setAttribute('src',images[de]);
}
function change_image(){
let de = Math.floor(Math.random() * images.length) ;
img.setAttribute('src',images[de]);
legende.textContent = legendes[de];
}
```

Voici le "main" du script Javascript

```
/*-----*/
var images = ['chat.png','chatBar.png','chatContour.png'];
var legendes = ['Image d\'un chat','Image inversée','Contour de l\'image du chat'];
var legende = document.querySelector('.legende');
var btn = document.querySelector('button');
btn.addEventListener('click', changer);
var img = document.querySelector('.image');
img.addEventListener('dblclick', change_image);
```

## 7.1 Exercices : HTML /CSS et Javascript

 $\mathbf{Ex} \ \mathbf{1}$ 

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<link href="style.css" rel="stylesheet"/>
<title>Premiere N.S.I</title>
</head>
```

```
< body >
```

```
<h1>Quelques langages de programmation</h1>
Python
C
C
<script type="text/javascript"
src="liste.js">
</script>
<button>Ajouter des langages</button>
</body>
</html>
```

- 1. Quel est le DOM de cette page html?
- 2. Quels sont les noeuds et les noeuds littéraux?
- 3. Faire un script Javascript qui ajoute les langages suivants OCaml ,C++ et php