

De nombreuses personnes confondent (à tort) Internet et le Web. Il faut savoir que le Web *fait partie* d'Internet. Internet est un grand ensemble qui comprend, entre autres : le Web, les e-mails, la messagerie instantanée, etc.

\*le web (toile ou réseau) désigne le système donnant accès à un ensemble de données (page, image, son, vidéo) reliées par des liens hypertextes et accessibles sur le réseau internet.

\*Sur le web, les contenus sont exprimés et assemblés dans divers formats normalisés par le consortium mondial W3C (world wide web consortium), qui permet une circulation standardisée de ces informations.

Le web n'est qu'une application d'Internet comme le courrier électronique, la messagerie instantanée, le partage de fichier, etc.

\*quand on parle du web, on pense généralement à la partie visible, indexée par les moteurs de recherche. Pourtant, il existe une partie invisible, que l'on appelle web profond (deep web).

Il comprend toutes les pages qui ne sont pas accessibles par les moteurs de recherche classiques.

Elles peuvent être de tous les types, mais ce sont le plus souvent des bases de données, des bibliothèques de données ou encore des pages protégées par un mot de passe.



## I. Repères historiques

### 1) L URL dans la barre de navigateur

#### Activité 1 : décrypter la navigation documents p 36 Hatier

a. Que fait cette URL ?

<mailto://nomducorrespondant@messagerie.fr?subject=Commande&body=Bonjour>

b. Qui peut administrer le sous domaine de « Corine Berthier » :

[www.corinneberthier.hebergeur.fr](http://www.corinneberthier.hebergeur.fr)

c. Expliquer le rôle du serveur lors d'une requête http.

d. Sur la requête http doc 4, dans quel en-tête apparaît le navigateur utilisé par le client ?

#### BILAN :



Apprendre à décrypter l'adresse d'un site Web est assez simple. Mais pourquoi le faire ?

Parce qu'une simple adresse peut renseigner sur la nature d'un site et des documents qu'il abrite, leur origine, voire leur crédibilité.

Savoir lire une adresse permet de mieux se repérer dans la masse de documents présents sur Internet en anticipant les contenus de l'information.

Une adresse Internet (ou URL) définit un itinéraire conduisant aux données recherchées dans le web. Elle rassemble un certain nombre de données que le navigateur interprète pour trouver l'information.

Dans toutes les parties du monde, la structure d'une adresse web est la même et est exploitable par tous les logiciels web.

Elle se décompose en parties selon les délimiteurs (/ ou barre oblique) et de gauche à droite.

1. L'indication de la nature du protocole : `http://` pour HyperTexte Transfert Protocole (un « s » est rajoutée à la fin lorsque le site est sécurisé.). il est généralement accompagné de `www`, mais ce n'est pas obligatoire.
2. Le nom Internet du site.
3. Un dossier et éventuellement des sous-dossiers
4. Un fichier `.html`.

Prenons comme exemple l'adresse Internet d'une banque.

`https://particuliers.societegenerale.fr/nous_contacter/service_urgence.html`

Découpage de l'adresse web :

**Découpage de l'adresse web**

<code>https://particuliers.societegenerale.fr/nous_contacter/service_urgence.html</code>			
<code>https://</code>	<code>particuliers.societegenerale.fr</code>	<code>/nous_contacter</code>	<code>/service_urgence.html</code>
Protocole utilisé : HyperText Transfert Protocole Ici : sécurisé (https)	C'est le nom Internet	Constitue le dossier Il peut y avoir un dossier et des sous-dossiers Bien entendu, leur nombre varie en fonction de l'emplacement de la page, et de la structure du site.	Est le nom du fichier HTML

**Exemple avec dossier et sous-dossier.**  
`https://professionnels.societegenerale.fr/votre_site/nous_contacter/services_clientele.html`

En fait, vous remarquez que cela ressemble énormément à l'adresse d'un fichier sur un disque dur à l'exception du protocole et du nom de domaine.

Cette similitude est tout à fait normale, puisque le serveur sur lequel se trouve le fichier n'est ni plus ni moins qu'un ordinateur et contient donc des disques durs qui sont structurés comme celui (ceux) de votre ordinateur.

**Les numéros IP et le nom Internet**

Chaque ordinateur relié à Internet peut être identifié par son numéro ou adresse IP, numéro unique, composé d'une suite de quatre chiffres (156.84.65.20) qui nous permet de le localiser.

Mais comme cette suite de numéros est trop complexe pour l'homme, celle-ci a été traduite par le "nom Internet", qui a la même fonction, mais permet d'être plus lisible.

**Le nom Internet est composé de trois parties séparées par un point**

Nous lisons de gauche à droite

- le nom du serveur ou de la machine
- le sous-domaine
- le domaine.

## Activité 2 : les URLs

Lors d'une navigation sur le web, l'internaute rencontre plusieurs formes d'adresse souvent cachées sous des liens. L'adresse apparaît explicitement soit au survol du lien, soit dans la barre du navigateur après avoir cliqué sur le lien.

1. Aller sur la page <http://lyc-sabatier-carcaassonne.fr/category/lycee-21>  
Noter l'adresse du lien et en particulier le premier mot de cette adresse avant le caractère `:`  
Que se passe-t-il si on clique sur ce lien ?
2. Enlever toute la partie d'adresse suivant le premier `/` de cette adresse.  
Que se passe-t-il si on clique sur ce lien raccourci ?
3. Aller sur <http://lyc-sabatier-carcaassonne.fr/category/vie-culturelle/les-dispositifs/section-sportive-judo>  
Puis clic droit sur l'article, puis afficher le code source de la page.  
En haut de la page, vous allez trouver <http://www.w3.org/1999/xhtml>, copier cette adresse et coller dans la barre d'adresse, puis entrée.  
Observer l'adresse qui apparaît dans la barre de saisie et en particulier avant les `:`  
Noter vos observations .....

#### 4. Bilan :

Quelle peut être la différence entre http et https ?

Si l'on remplace http par mailto, que se passe-t-il ?

Interpréter le sens de cette première partie de l'adresse.

#### Conclusion :

**Le nom Internet**

Reprenons l'exemple précédent :

[https://particuliers.societegenerale.fr/nous\\_contacter/service\\_urgence.html](https://particuliers.societegenerale.fr/nous_contacter/service_urgence.html)

Dans cette adresse, Le nom Internet est : particuliers.societegenerale.fr

particuliers.societegenerale.fr		
particuliers	societegenerale	.fr
Nom du serveur de la clientèle de particuliers	sous domaine (nom de l'entreprise)	domaine (France)

**Découpage de l'adresse web**

Une U.R.L. (une adresse internet) se lira de la façon suivante :

protocole://www.nom\_internet/dossier/sous\_dossier/nom\_du\_fichier

 Sans le www

protocole://nom\_internet/dossier/sous\_dossier/nom\_du\_fichier

**Quelques exemples de domaines**

Des domaines de fonction :      Des domaines de Pays :

.com	site commercial
.edu	site éducatif américain
.net	fournisseurs réseau x
.gov	gouvernement méricain
.org	organisation non-lucrative
.mil	militaire US
.ac	académique (Royaume Uni)
.info	actualité, information

fr	France
ca	Canada
ch	Suisse
uk	Royaume Uni
it	Italie
es	Espagne
eu	Union Européenne

### Activité 3 : Le WEB

1- Qui est le créateur d' internet ?

<https://twitter.com/franceculture/status/1105353736078139392>

A partir du portrait de **Tim Berners Lee** répondre au questionnaire :

<https://learningapps.org/watch?v=puhzmtka319>

2- Frise chronologique **Historique du WEB**

Replacer les informations au bon endroit, rechercher sur internet :

<https://learningapps.org/6698506>

( frise réalisée à partir des dates fournies dans le programme)

3- **Faire une recherche dans un moteur de recherche sur Internet et sur e-sidoc**

Comment fonctionne un moteur de recherche : Vidéo de Claire Cassaigne (1min30)

**Comparer Un moteur de recherche sur Internet qu'ils utilisent en masse Google et Esidoc :**

Un **univers infini** d' un côté (des milliards de résultats) et un univers **fini** de l' autre (30000 références au grand max).

Des sites non vérifiés à valider soi-même (validation des sources : auteur, fautes, site officiel, **sources** ...) d' un côté et des ressources variées (livres, articles, sites...) vérifiées validées par des auteurs reconnus, des éditeurs et des professeurs documentalistes de l' autre.

**Point commun** : bandeau de recherche, mots-clés (descripteurs) à croiser avec des booléens (et/ou sauf avec un “et implicite dans les deux cas / AND OR “-” ), citation exacte avec les guillemets, troncature, possibilité d’affiner (date, support ou type de fichier, auteur, éditeur, type de site...).

Pour présenter les booléens :

- faire se lever les élèves en fonction des consignes (ex : cheveux longs ET yeux bleus / filles OU garçons)
- noter les résultats au tableau et les analyser avec eux (quand y a-t-il le plus de résultats avec ET ou avec OU ?)

Présentation recherche (vidéo) **en attendant la nouvelle version d’Esidoc** :

<http://documentation.solutionsdoc.net/la-recherche-usager/>

<https://www.youtube.com/watch?v=RQ5YLWNtUI0&t=21s>

[https://fenetresur.files.wordpress.com/2012/10/fiche\\_outil\\_cdi\\_recherche\\_esidoc.pdf](https://fenetresur.files.wordpress.com/2012/10/fiche_outil_cdi_recherche_esidoc.pdf)

<https://fr.slideshare.net/thetys258/faire-une-recherche-simple>

#### **Activité 4 : Un site à explorer**

**Objectif** : découvrir un site emblématique du web

L encyclopédie en ligne wikipedia fait partie des sites web les plus consultés. Son fonctionnement collaboratif est pourtant moins connu.

C est pourtant un des sites les plus caractéristiques de ce que l avènement du web a rendu possible.

1. Consulter la page principale d accueil de la version française du site <https://fr.wikipedia.org>
2. Combien de pages comporte le site ?
3. Rechercher les principes fondateurs de wikipedia.
4. Rechercher des informations sur le world wide web.
5. Ouvrir l onglet **voir l historique** pour consulter les modifications de la page. Jusqu’à quelle date peut-on remonter ? Comment était considéré le web à cette époque ?

Conclusion :

Une page web est accessible via un navigateur web ( mozilla firefox, google chrome, safari, etc...)

Elle dispose d une adresse web unique, nommée URL (uniform resource locator) , à saisir directement dans la barre d adresse de ces navigateurs.

Elle peut aussi être retrouvée via un moteur de recherche, ou encore en suivant un lien hypertexte présent sur le site ou un autre site web.

## ***II. Constitution d une Page web***

### **a. Comprendre la structure d une page web**

La conception d un site internet et de son interface correspond au travail d ergonomie. Elle a pour but l optimisation de la navigation et de l expérience utilisateur.

#### **Activité 5 :**

**Objectif** :

L’objectif est d’analyser le contenu d’une page web existante et écrite en langage HTML.

- Afficher le code source d’une page web depuis un navigateur.
- Repérer sur une page web ce qui relève du contenu et ce qui relève du style.

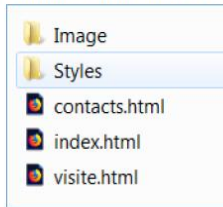


►Changer la mise forme d'une page en modifiant son CSS.

## TD html : Découverte du HTML et des CSS

Ce TD a pour but de découvrir comment sont faites les pages web.  
Chaque page web est en fait un programme codé en langage HTML.

Contenu du dossier TD1CRIEE :



Ce dossier est à télécharger dans votre répertoire personnel sur le réseau du lycée :

\*Image : les images du site

\* Styles : Feuilles de styles utilisées

\* Index.html, visite.html et contacts.html : les trois pages du site

### Écran 1 : Le code HTML de la page

1. Ouvrir la page index.html dans votre navigateur .

**Bienvenue sur le site de la criée de POULGOAZEC**

**Port de Plouhinec / Poulgoazec**



- [Accueil](#)
- [Visite de la criée](#)
- [Contacts](#)

La criée de POULGOAZEC est :

- située sur la commune de Plouhinec (29)
- constituée de 48 bateaux

Préserver une activité, c'est aussi la faire connaître. La criée de Poulgoazec ouvre ses portes et ses bateaux.

!!! Nouvelles activités de la criée !!!

1. Sorties en mer
  1. Individuelles
  2. Groupes
2. Visites de la criée

Directrice de publication : Madame Tarot

Webmestre : Monsieur Rivoire

### Écran 2 : La page web étudiée

## 2. Ouvrir la page index.html dans l'éditeur HTML

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="utf_8">
5   <title> Criée de Poulgoazec</title>
6
7   <!--<link rel="stylesheet" href="Styles/style1.css"> -->
8
9 </head>
10
11 <body>
12   <header>
13     <h1>Bienvenue sur le site de la criée de POULGOAZEC </h1>
14     <h2>Port de Plouhinec / Poulgoazec</h2>
15     
16     <hr />
17   </header>
18   <nav >
19     <ul>
20       <li>
21         <a href="index.html" >Accueil</a>
22       </li>
23       <li>
24         <a href="visite.html" target="_blank" >Visite de la criée</a>
25       </li>
26       <li>
27         Contacts
28       </li>
29     </ul>
30   </nav>
31   <section >
32     <p> La criée de POULGOAZEC est : </p>
33     <ul>
34       <li>située sur la commune de Plouhinec (29)</li>
35       <li>constituée de 48 bateaux </li>
36     </ul>
```

## MODIFICATION DES PAGES HTML DU SITE

3. Modifier la page index.html afin d'afficher l'image port.jpg en remplacement de l'image crieepoulgoazec.jpg

4. Modifier la liste à puces suivante

La criée de POULGOAZEC est :

- située sur la commune de Plouhinec (29)
- constituée de 48 bateaux

afin de rajouter la ligne « financée à 85% par la région » et d'obtenir le résultat suivant :

La criée de POULGOAZEC est :

- située sur la commune de Plouhinec (29)
- constituée de 48 bateaux
- financée à 85% par la région

5. Ouvrir la page visite.html dans l'éditeur HTML et ajouter un horaire de visite le vendredi à 8H afin d'obtenir le résultat suivant :

### Les Horaires

Jours	Heures de rendez-vous
Lundi	7 heure
Jeudi	8 heure
Vendredi	8 heure

6. Sur les pages `index.html` et `visite.html`, ajouter le lien hypertexte qui permet d'ouvrir la page `contacts.html` quand l'utilisateur clique sur l'option « contacts » du menu

## **MODIFICATION DE LA FEUILLE DE STYLE DU SITE**

La ligne 7 est un commentaire. Un commentaire se paramètre ainsi : `<!--commentaire-->`.

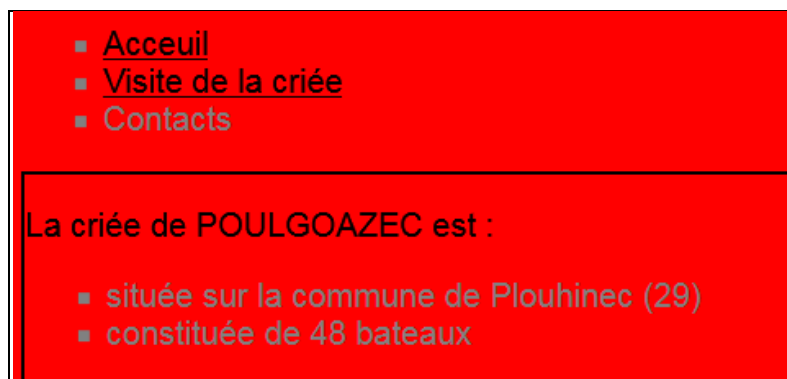
7. Modifiez la ligne dans les pages `index.html` et `visite.html` en supprimant les marques de commentaire afin de faire le lien entre les pages HTML et la feuille de style.
8. Enregistrez votre travail puis visualisez dans le navigateur les pages du site.

Visitez la page en passant votre souris sur les liens....

9. Que remarquez-vous ?

Ouvrir la page `style1.css` dans votre éditeur HTML

10. Modifier la couleur d'arrière-plan des pages du site
11. Modifier la puce des listes afin d'obtenir le résultat suivant :



12. Modifier le style de la bordure du bloc « section »

## **POUR ALLER PLUS LOIN.....**

### **Document HTML**

La page d'accueil présente un lien vers la page `visite.html` qui présente toutes les activités de la criée. M. Rivoire souhaite ajouter les contacts à la fin de la page afin de permettre aux internautes de téléphoner pour se renseigner sur les visites.

1. Modifier la page `visite.html`, et insérer le texte suivant au-dessus du tableau des horaires :

Le programme de la visite:

1. Rendez-vous fixé au local de la Mairie annexe sur le port.
2. Projection d'une vidéo sur la pêche au bar
3. Visite de la criée
4. Débat avec les marins pêcheurs

2. Modifier la page `visite.html`, et insérer le tableau suivant après le tableau des horaires.

Nom	Téléphone	E-mail
M. Maréchal	06.65.34.87.56	marechal@poulgoaec.com
M. Grumier	06.23.54.16.78	grumier@poulgoazec.com

3. Tout comme cela a été fait pour les mots "description" et "horaires", faites-en sorte qu'un clic sur le mot `contacts` conduise directement au tableau des contacts en bas de page.

## Document CSS

4. Inversez les listes ordonnées pour obtenir la présentation suivante :

- A. Sorties en mer
  - 1. Individuelles
  - 2. Groupes
- B. Visites de la criée

5. Amusez-vous à modifier la présentation et la charte graphique selon vos goûts !!

## Activité 6 : Manipuler le HTML et le CSS

### **I. Découvrez le fonctionnement des sites web**

#### Partie 1 - Découvre les bases de HTML5

- Aller sur la page d'accueil de wikipedia.

- clic droit : afficher le code source de la page. Apparaît alors le code html complet qui fait fonctionner wikipedia.

Le navigateur a pour rôle de transformer une page html en une page web telle que nous les connaissons.

Les 2 langages html et css sont complémentaires pour créer une page web.

**Les langages HTML et CSS sont à la base du fonctionnement de tous les sites web. Quand vous consultez un site avec votre navigateur, il faut savoir que, en coulisses, des rouages s'activent pour permettre au site web de s'afficher. L'ordinateur se base sur ce qu'on lui a expliqué en HTML et CSS pour savoir ce qu'il doit afficher, comme le montre la figure suivante.**

Html décrit ce qu'il y a dans la page.

CSS décrit la présentation de la page : comment cela va s'afficher.

Pour écrire une page html, on a besoin d'un logiciel : sublime text ou notepad++, outil pour rédiger du code.

Attention le navigateur utilisé peut avoir un impact sur la page web qu'on est entrain de créer : si vous voulez savoir si votre navigateur peut être utilisé pour lire un code html ou css, vous pouvez utiliser le logiciel CAN I USE.

**HTML et CSS sont deux « langues » qu'il faut savoir parler pour créer des sites web. C'est le navigateur web qui fera la traduction entre ces langages informatiques et ce que vous verrez s'afficher à l'écran.**

#### Partie 2 : Les rôles de HTML et CSS

Pour expliquer à l'ordinateur ce que vous voulez faire, il va falloir utiliser un langage qu'il comprend. Et c'est là que les choses se corsent, parce qu'il va falloir apprendre deux langages !

Pourquoi avoir créé deux langages ? Un seul aurait suffi, non ?

Vous devez vous dire que manipuler deux langages va être deux fois plus complexe et deux fois plus long à apprendre... mais ce n'est pas le cas ! Je vous rassure, s'il y a deux langages c'est, au contraire, pour faciliter les choses. Nous allons avoir affaire à deux langages qui *se complètent* car ils ont des rôles différents :

- **HTML** (*HyperText Markup Language*) : il a fait son apparition dès 1991 lors du lancement du Web. Son rôle est de gérer et organiser le contenu. C'est donc en HTML que vous écrirez ce qui doit être affiché sur la page : du texte, des liens, des images... Vous direz par exemple : « Ceci est mon titre, ceci est mon menu, voici le texte principal de la page, voici une image à afficher, etc. ».
- **CSS** (*Cascading Style Sheets*, aussi appelées *Feuilles de style*) : le rôle du CSS est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte...). Ce langage est venu compléter le HTML en 1996.

Vous avez peut-être aussi entendu parler du langage XHTML. Il s'agit d'une variante du HTML qui se veut plus rigoureuse et qui est donc un peu plus délicate à manipuler. Elle n'est plus vraiment utilisée aujourd'hui.

Dans ce cours, nous allons travailler sur la dernière version de HTML (HTML5) qui est aujourd'hui le langage d'avenir que tout le monde est incité à utiliser.



Vous pouvez très bien créer un site web uniquement en HTML, mais celui-ci ne sera pas très beau : l'information apparaîtra « brute ». C'est pour cela que le langage CSS vient toujours le compléter.

Pour vous donner une idée, la figure suivante montre ce que donne la même page sans CSS puis avec le CSS.

HTML  
(pas de CSS)



HTML + CSS



Le HTML définit le contenu (comme vous pouvez le voir, c'est brut de décoffrage !). Le CSS permet, lui, d'arranger le contenu et de définir la présentation : couleurs, image de fond, marges, taille du texte...

Comme vous vous en doutez, le CSS a besoin d'une page HTML pour fonctionner. C'est pour cela que nous allons d'abord apprendre les bases du HTML avant de nous occuper de la décoration en CSS.

Vos premières pages ne seront donc pas les plus esthétiques, mais qu'importe ! Cela ne durera pas longtemps.

### Partie 3 : Les différentes versions de HTML et CSS

Au fil du temps, les langages HTML et CSS ont beaucoup évolué. Dans la toute première version de HTML (HTML 1.0) il n'était même pas possible d'afficher des images !

Voici un très bref historique de ces langages pour votre culture générale.

#### Les versions de HTML

- **HTML 1** : c'est la toute première version créée par Tim Berners-Lee en 1991.
- **HTML 2** : la deuxième version du HTML apparaît en 1994 et prend fin en 1996 avec l'apparition du HTML 3.0. C'est cette version qui posera en fait les bases des versions suivantes du HTML. Les règles et le fonctionnement de cette version sont donnés par le W3C (tandis que la première version avait été créée par un seul homme).
- **HTML 3** : apparue en 1996, cette nouvelle version du HTML rajoute de nombreuses possibilités au langage comme les tableaux, les applets, les scripts, le positionnement du texte autour des images, etc.
- **HTML 4** : cette version aura été utilisée un long moment durant les années 2000. Elle apparaît pour la première fois en 1998 et propose l'utilisation de frames (qui découpent une page web en plusieurs parties), des tableaux plus complexes, des améliorations sur les formulaires, etc. Mais surtout, cette version permet pour la première fois d'exploiter des feuilles de style, notre fameux CSS !
- **HTML 5** : c'est la dernière version. De plus en plus répandue, elle fait beaucoup parler d'elle car elle apporte de nombreuses améliorations comme la possibilité d'inclure facilement des vidéos, un meilleur agencement du contenu, de nouvelles fonctionnalités pour les formulaires, etc. C'est cette version que nous allons découvrir ensemble.

#### Les versions de CSS

- **CSS 1** : dès 1996, on dispose de la première version du CSS. Elle pose les bases de ce langage qui permet de présenter sa page web, comme les couleurs, les marges, les polices de caractères, etc.
- **CSS 2** : apparue en 1999 puis complétée par CSS 2.1, cette nouvelle version de CSS rajoute de nombreuses options. On peut désormais utiliser des techniques de positionnement très précises, qui nous permettent d'afficher des éléments où on le souhaite sur la page.
- **CSS 3** : c'est la dernière version, qui apporte des fonctionnalités particulièrement attendues comme les bordures arrondies, les dégradés, les ombres, etc.

Notez que HTML5 et CSS3 ne sont pas encore des versions "officiellement" finalisées par le W3C. Cependant, même s'il peut y avoir des changements mineurs dans ces langages, je vous recommande chaudement de commencer dès aujourd'hui avec ces nouvelles versions. Leurs apports sont nombreux et valent vraiment le coup. La plupart des sites web professionnels se construisent aujourd'hui sur ces dernières versions.

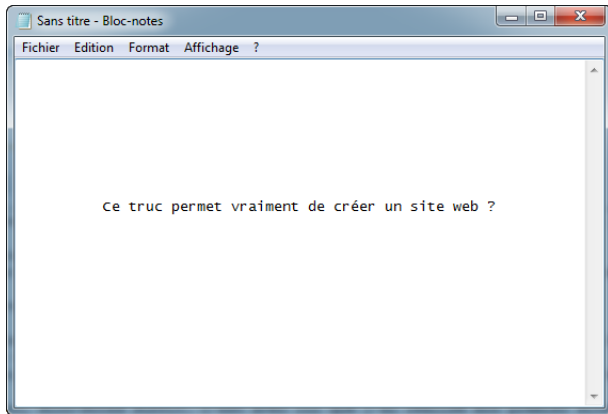
#### Partie 4 : L'éditeur de texte

De quel logiciel vais-je avoir besoin pour créer mon site web ?

Vais-je devoir casser ma tirelire pour acheter un logiciel très complexe que je vais mettre des mois à comprendre ?

*Il existe effectivement de nombreux logiciels dédiés à la création de sites web. Mais, je vous rassure, vous n'aurez pas à déboursier un seul centime. Pourquoi aller chercher un logiciel payant et compliqué, alors que vous avez déjà tout ce qu'il faut chez vous ?*

Eh oui, accrochez-vous bien parce qu'il suffit de... Bloc-Notes !



*Le logiciel Bloc-notes de Windows*

Incroyable mais vrai : on peut tout à fait créer un site web uniquement avec Bloc-Notes, le logiciel d'édition de texte intégré par défaut à Windows.

Il y a cependant des logiciels plus puissants aujourd'hui et personne n'utilise vraiment Bloc-Notes. On peut classer ces logiciels de **création de site web** en deux catégories :

- Les **WYSIWYG** (*What You See Is What You Get* - Ce Que Vous Voyez Est Ce Que Vous Obtenez) : ce sont des programmes qui se veulent très faciles d'emploi, ils permettent de créer des sites web sans apprendre de langage particulier. Parmi les plus connus d'entre eux : Mozilla Composer, Microsoft Expression Web, Dreamweaver... et même Word ! Leur principal défaut est la qualité souvent assez mauvaise du code HTML et CSS qui est automatiquement généré par ces outils. Un bon créateur de site web doit tôt ou tard connaître HTML et CSS, c'est pourquoi je ne recommande pas l'usage de ces outils.
- Les **éditeurs de texte** : ce sont des programmes dédiés à l'écriture de code. On peut en général les utiliser pour de multiples langages, pas seulement HTML et CSS. Ils se révèlent être de puissants alliés pour les créateurs de sites web !

Vous l'aurez compris, nous allons utiliser un éditeur de texte dans ce cours.

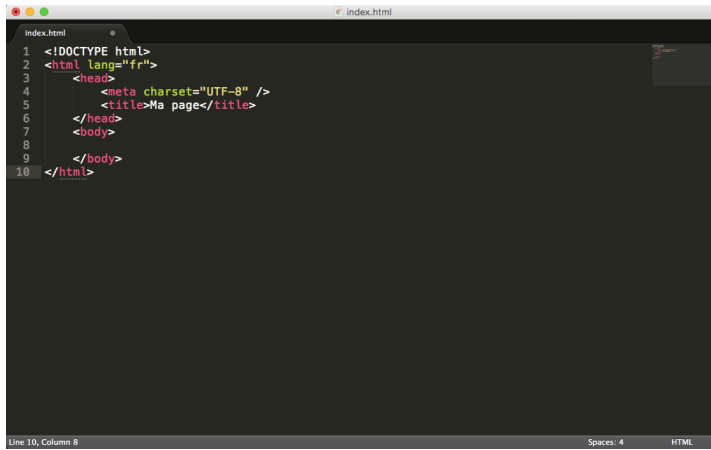
De nombreux éditeurs de texte fonctionnent que vous soyez sous Windows, Mac OS X ou Linux (ils sont disponibles partout). Nous allons commencer par présenter... Sublime Text !

#### Partie 5 : Sublime Text : mon éditeur

Sublime Text est un éditeur de texte devenu très populaire parmi les développeurs. On l'utilise aussi bien pour développer en HTML et CSS que dans d'autres langages (Python, Ruby, etc.). Il fonctionne sur Windows, Mac OS X et Linux.

##### Site web de Sublime Text

Il a l'avantage d'être simple, épuré et facile à lire dès le départ. Pas de centaines de boutons dont on ne comprend pas à quoi ils servent.



```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="UTF-8" />
5     <title>Ma page</title>
6   </head>
7   <body>
8
9 </body>
10 </html>
```

*L'éditeur Sublime Text : c'est beau, c'est propre, c'est pur*

Malgré les apparences, il ne faut pas croire qu'il est limité. Au contraire : il est possible de l'étendre avec tout un système de plugins.

ça devient un peu plus compliqué et on ne rentrera pas là-dedans, mais il faut savoir que certains personnalisent énormément leur Sublime Text pour gagner du temps !

En somme, **Sublime Text est à la fois simple et puissant**. Même pour l'usage basique que nous allons avoir, il s'avèrera très pratique.

Sublime Text peut tout à fait être utilisé gratuitement, mais son auteur demande à payer au bout d'un certain temps d'usage. Vous pourrez toujours continuer à l'utiliser gratuitement mais de temps en temps un écran vous rappellera que ce serait bien de payer pour le logiciel. 😊

### Sous Windows

Voici quelques logiciels que vous pouvez essayer sous Windows si vous voulez en tester plusieurs :

- [Sublime Text](#) ;
- [Notepad++](#) ;
- [Brackets](#) ;
- [jEdit](#) ;
- [PSPad](#) ;
- [ConTEXT](#) ;
- ... et bien d'autres si vous recherchez « Éditeur de texte » sur le Web.

### Sous Mac OS X

Je recommande la plupart des mêmes logiciels car ils sont multi-plateformes. Voici une petite sélection :

- [Sublime Text](#) ;
- [Brackets](#) ;
- [jEdit](#) ;
- [Smultron](#) ;
- [TextWrangler](#).

## Partie 6 : Les navigateurs

### *Pourquoi le navigateur est important ?*





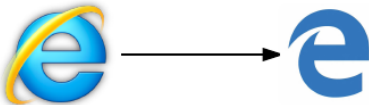


Le navigateur est le programme qui nous permet de voir les sites web. Comme je vous l'ai expliqué plus tôt, le travail du navigateur est de lire le code HTML et CSS pour afficher un résultat visuel à l'écran. Si votre code CSS dit « Les titres sont en rouge », alors le navigateur affichera les titres en rouge. Le rôle du navigateur est donc essentiel !

On ne dirait pas, mais un navigateur est un programme extrêmement complexe. En effet, comprendre le code HTML et CSS n'est pas une mince affaire. Le principal problème, vous vous en rendrez vite compte, c'est que *les différents navigateurs n'affichent pas le même site exactement de la même façon* ! Il faudra vous y faire et prendre l'habitude de vérifier régulièrement que votre site fonctionne correctement sur la plupart des navigateurs.

## Les navigateurs sur ordinateur

### Télécharger les navigateurs

Il existe de nombreux navigateurs différents. Voici les principaux à connaître :

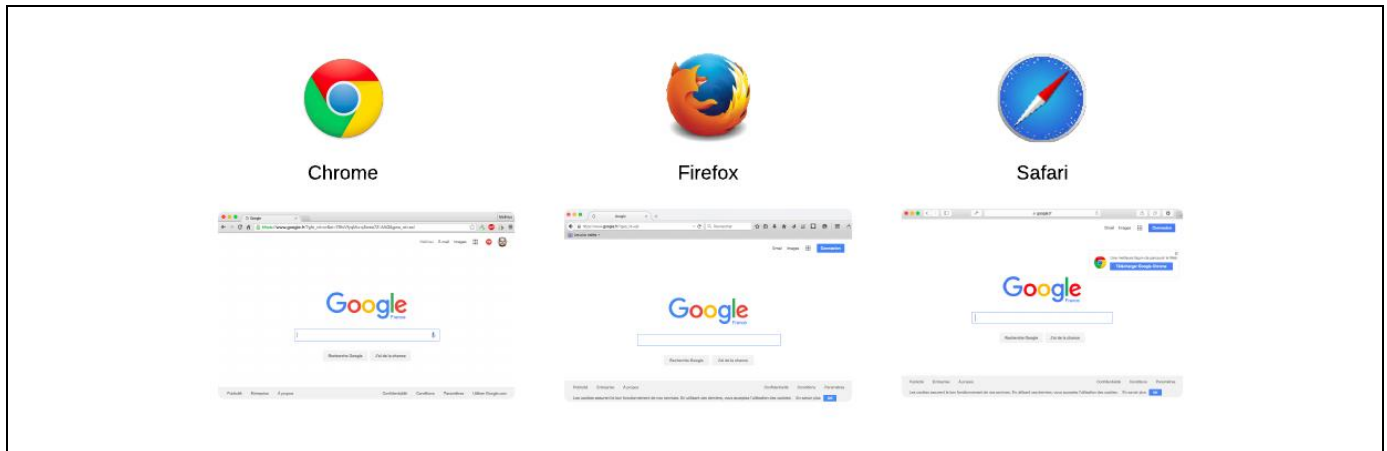
Navigateur	OS	Téléchargement	Commentaires
<b>Google Chrome</b>  Google Chrome	Windows Mac Linux	<a href="#">Téléchargement</a>	Le navigateur de Google, simple d'emploi et très rapide. <b>C'est le navigateur que j'utilise au quotidien.</b>
<b>Mozilla Firefox</b>  Firefox	Windows Mac Linux	<a href="#">Téléchargement</a>	Le navigateur de la fondation Mozilla, célèbre et réputé. Je l'utilise fréquemment pour tester mes sites web.
<b>Internet Explorer</b>  IE9	Windows	(Déjà installé sur Windows)	Le navigateur de Microsoft, qui équipe tous les PC Windows jusqu'à Windows 10.
<b>Edge</b>  Edge	Windows	(Déjà installé sur Windows 10)	Le nouveau navigateur de Microsoft, qui équipe tous les PC à partir de Windows 10. Il ressemble à Internet Explorer (les logos sont proches !) mais c'est une toute nouvelle version bien plus à jour. Edge est le remplaçant d'Internet Explorer.
 Internet Explorer → Edge			
Internet Explorer est remplacé par Edge			
<b>Safari</b>  Safari	Windows Mac	<a href="#">Téléchargement</a> (Déjà installé sur Mac OS X)	Le navigateur d'Apple, qui équipe tous les Mac.
<b>Opera</b>  Opera	Windows Mac Linux	<a href="#">Téléchargement</a>	L'éternel <i>outsider</i> . Il est moins utilisé mais propose de nombreuses fonctionnalités.

Il est conseillé d'installer plusieurs navigateurs sur son ordinateur pour s'assurer que son site fonctionne correctement sur chacun d'eux.

De manière générale, je conseille de tester son site web régulièrement au moins sur Google Chrome, Mozilla Firefox et Internet Explorer/Edge.

Notez que Safari et Google Chrome affichent les sites web quasiment de la même façon. Il n'est pas forcément nécessaire de tester son site sur Safari et Google Chrome, même si c'est toujours plus sûr.

La figure suivante vous montre un aperçu du résultat produit par quelques-uns de ces principaux navigateurs sur la page d'accueil de Google.



Aperçu de quelques navigateurs

## Comprendre les différences entre navigateurs

A vue de nez, ces navigateurs se ressemblent beaucoup.

Mais Pourquoi les navigateurs n'affichent pas toujours un même site web *exactement* de la même façon. ?

Cela est dû au fait que les navigateurs ne connaissent pas toujours les dernières fonctionnalités de HTML et CSS. Par exemple, Internet Explorer a longtemps été en retard sur certaines fonctionnalités CSS (et paradoxalement, il a aussi été en avance sur quelques autres).

Pour compliquer les choses, **plusieurs versions des navigateurs co-existent**. Aujourd'hui, un navigateur comme Chrome sort une nouvelle version presque tous les mois. Les mises à jour sont (heureusement) de plus en plus fréquentes.

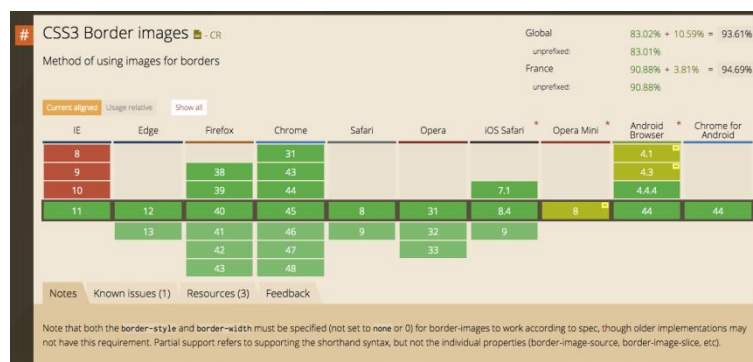
Chaque version prend en charge de nouvelles fonctionnalités mais, si les utilisateurs ne mettent pas à jour leur(s) navigateur(s), cela devient un problème pour les **webmasters** comme vous qui créent des sites web.

Chrome a résolu en grande partie le problème en mettant en place des mises à jour automatiques, sans intervention de l'utilisateur. Firefox a semble-t-il décidé de suivre le rythme lui aussi. Internet Explorer est de plus en plus à jour, et son remplaçant Edge n'a pas à rougir des autres navigateurs.

Bref, les navigateurs supportent un grand nombre de fonctionnalités.

La compatibilité reste toujours un problème malgré tout, mais ce n'est pas aussi grave qu'à une époque.

Le célèbre site [caniuse.com](http://caniuse.com) tient à jour une liste des fonctionnalités prises en charge par les différentes versions de chaque navigateur (figure suivante).



*caniuse.com vous permet de connaître la compatibilité d'une fonctionnalité*

## Les navigateurs sur mobile

En plus de ces navigateurs, il faut savoir qu'il existe des variantes de ces navigateurs conçues pour les téléphones portables, en particulier pour les **smartphones**.



De plus en plus de personnes consultent aujourd'hui des sites web sur leur portable, il faut donc connaître un minimum le fonctionnement des navigateurs des téléphones.

En fait, vous n'allez pas être dépayés : la plupart des navigateurs sur smartphones sont les mêmes que sur ordinateur, dans une version plus légère adaptée aux mobiles. Tout dépend du type de téléphone.

- **iPhone** : sur l'iPhone d'Apple, le navigateur utilisé est Safari Mobile. Il s'agit d'une version *light* et néanmoins très complète de Safari pour ordinateur.
- **Android** : les portables sous Android bénéficient du navigateur Chrome Mobile. Là encore, il s'agit d'une version adaptée aux mobiles.
- **Windows Phone** : sous Windows Phone, on retrouve... Internet Explorer/Edge Mobile ! Le principe est le même que pour les précédents navigateurs : il s'agit d'une version dédiée aux mobiles.
- **Blackberry** : les Blackberry font exception car ils ont leur propre navigateur (il n'existe pas d'équivalent sur ordinateur). Néanmoins, les versions les plus récentes de ce navigateur se basent sur un noyau commun à Safari et Chrome (il s'agit du moteur de rendu Webkit). Par conséquent, l'affichage est en général proche de celui proposé par Safari et Chrome. Enfin, il faut reconnaître que les Blackberry sont de moins en moins utilisés.

Les navigateurs pour mobiles prennent en charge la plupart des dernières fonctionnalités de HTML et CSS. De plus, le système de mise à jour automatisé des mobiles nous garantit que les utilisateurs auront le plus souvent les dernières versions.

Sachez néanmoins que des différences existent entre ces différents navigateurs mobiles et qu'il est conseillé de tester son site sur ces appareils aussi ! En particulier, l'écran étant beaucoup moins large, il faudra vérifier que votre site s'affiche correctement.

Les tablettes tactiles sont équipées des mêmes navigateurs, l'écran est simplement plus large. Ainsi, l'iPad est fourni avec Safari Mobile.

### En résumé

- Le Web a été inventé par Tim Berners-Lee au début des années 1990.
- Pour créer des sites web, on utilise deux langages informatiques :
  - HTML : permet d'écrire et organiser le contenu de la page (paragraphe, titres...);
  - CSS : permet de mettre en forme la page (couleur, taille...).
- Il y a eu plusieurs versions des langages HTML et CSS. Les dernières versions sont HTML5 et CSS3.
- Le navigateur web est un programme qui permet d'afficher des sites web. Il lit les langages HTML et CSS pour savoir ce qu'il doit afficher.
- Il existe de nombreux navigateurs web différents : Google Chrome, Mozilla Firefox, Internet Explorer, Safari, Opera... Chacun affiche un site web de manière légèrement différente des autres navigateurs.
- Dans ce cours, nous allons apprendre à utiliser les langages HTML et CSS. Nous travaillerons dans un programme appelé « éditeur de texte » (Sublime Text, Notepad++, jEdit, vim...).

## **II. Créez votre première page web en HTML**

Vous avez installé tous les logiciels ?

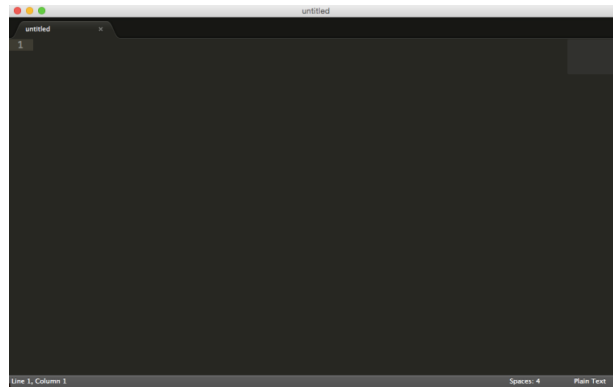
Vous devriez maintenant avoir un éditeur de texte pour *créer votre site* (comme Sublime Text) et plusieurs navigateurs pour le *tester* (Mozilla Firefox, Google Chrome...).

Dans ce chapitre, nous allons commencer à pratiquer ! Nous allons découvrir les bases du langage HTML et enregistrer notre toute première page web !

Alors oui, bien sûr, ne vous attendez pas encore à réaliser une page web exceptionnelle dès ce second chapitre, mais patience... cela viendra !

### Partie 1 : Créer une page web avec l'éditeur

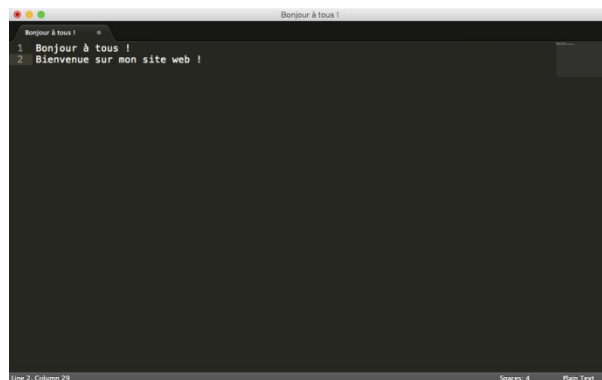
Allez, mettons-nous en situation ! Nous allons créer notre site dans un éditeur de texte : Sublime Text.



*Ouverture de Sublime Text*

Bon, qu'est-ce qu'on fait maintenant ? Qu'est-ce qu'on écrit ?

On va faire un petit essai. Je vous invite à écrire ce qui vous passe par la tête, comme moi à la figure suivante.

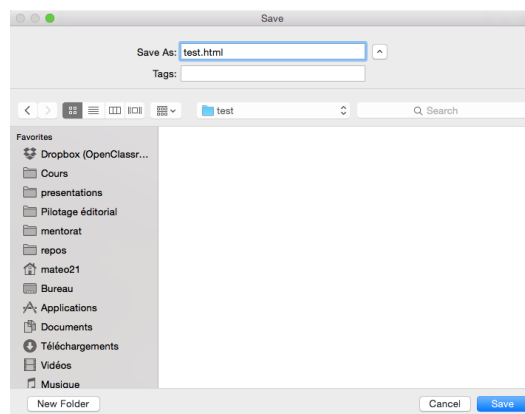


*Du texte dans Sublime Text*

Maintenant, enregistrons ce fichier.

Pour cela, c'est très simple : comme dans tous les programmes, vous avez un menu Fichier>Enregistrer (ou File >Save en anglais)

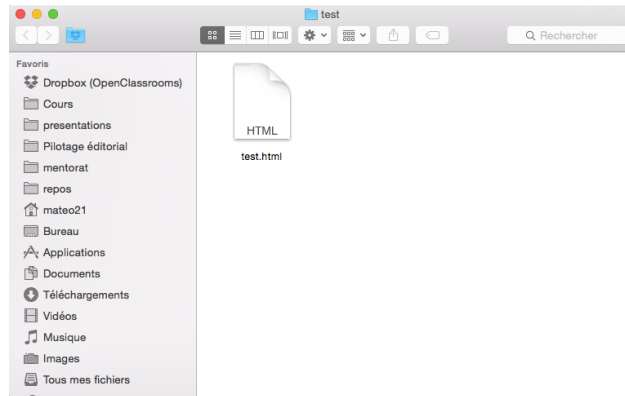
Une boîte de dialogue vous demande où enregistrer le fichier et sous quel nom. Enregistrez-le où vous voulez. Donnez au fichier le nom que vous voulez, en terminant par .html, par exemple test.html, comme indiqué à la figure suivante.



*Enregistrement d'un fichier sous Sublime Text*

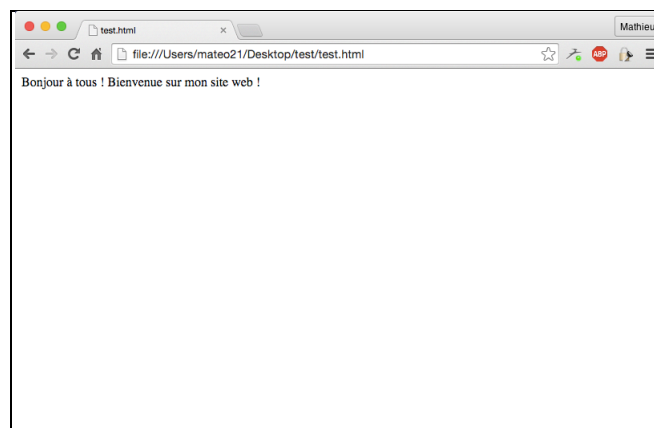
Je vous recommande de créer un nouveau dossier test dans vos documents qui contiendra les fichiers de votre site. Puis y mettre votre fichier test.html.

Ouvrez maintenant l'explorateur de fichiers dans le dossier où vous avez enregistré votre page (l'Explorateur sous Windows ou le Finder sous Mac). Vous y verrez le fichier que vous venez de créer:



*Le fichier dans l'explorateur*

L'icône qui représente le fichier dépend de votre navigateur web par défaut en général. Vous pouvez y voir une icône de Firefox, de Chrome... ou un aperçu comme ici. N'y prêtez pas attention. Faites simplement un double-clic sur ce fichier et... votre navigateur s'ouvre et affiche le texte que vous avez écrit.



*La page web affichée*

Cela ne marche pas bien, on dirait ! Tout le texte s'affiche sur la même ligne alors qu'on avait écrit deux lignes de texte différentes !?

En effet, bien vu !

Le texte s'affiche sur la même ligne alors qu'on avait demandé à l'écrire sur deux lignes différentes. Que se passe-t-il ?

En fait, pour créer une page web il ne suffit pas de taper simplement du texte comme on vient de le faire. En plus de ce texte, il faut aussi écrire ce qu'on appelle des **balises**, qui vont donner des instructions à l'ordinateur comme « aller à la ligne », « afficher une image », etc.

## Partie 2 : Les balises et leurs attributs

Il ne suffit pas d'écrire « simplement » du texte dans l'éditeur, il faut aussi donner des instructions à l'ordinateur. En HTML, on utilise pour cela des **balises**.

### Les balises

Les pages HTML sont remplies de ce qu'on appelle des balises. Celles-ci sont invisibles à l'écran pour vos visiteurs, mais elles permettent à l'ordinateur de comprendre ce qu'il doit afficher.

Les balises se repèrent facilement. Elles sont entourées de « chevrons », c'est-à-dire des symboles `<et>`, comme ceci :`<balise>`

À quoi est-ce qu'elles servent ? Elles indiquent la nature du texte qu'elles encadrent. Elles veulent dire par exemple : « Ceci est le titre de la page », « Ceci est une image », « Ceci est un paragraphe de texte », etc.

On distingue deux types de balises : les balises en paires et les balises orphelines.

### Les balises en paires

Elles s'ouvrent, contiennent du texte, et se ferment plus loin. Voici à quoi elles ressemblent :

```
<titre>Ceci est un titre</titre>
```

On distingue une balise ouvrante (`<titre>`) et une balise fermante (`</titre>`) qui indique que le titre se termine. Cela signifie pour l'ordinateur que tout ce qui n'est *pas* entre ces deux balises... n'est pas un titre.

Ceci n'est pas un titre <titre>Ceci est un titre</titre> Ceci n'est pas un titre

## Les balises orphelines

Ce sont des balises qui servent le plus souvent à insérer un élément à un endroit précis (par exemple une image). Il n'est pas nécessaire de délimiter le début et la fin de l'image, on veut juste dire à l'ordinateur « Insère une image ici ».

Une balise orpheline s'écrit comme ceci :

```
<image />
```

Notez que le / de fin n'est pas obligatoire. On pourrait écrire seulement<image>. Néanmoins, pour ne pas les confondre avec le premier type de balise, les webmasters recommandent de rajouter ce /(*slash*) à la fin des balises orphelines. Vous me verrez donc mettre un /aux balises orphelines et je vous recommande de faire de même, c'est une bonne pratique.

## Les attributs

Les attributs sont un peu les options des balises. Ils viennent les compléter pour donner des informations supplémentaires. L'attribut se place après le nom de la balise ouvrante et a le plus souvent une valeur, comme ceci :

```
<balise attribut="valeur">
```

À quoi cela sert-il ? Prenons la balise<image />que nous venons de voir. Seule, elle ne sert pas à grand chose. On pourrait rajouter un attribut qui indique le nom de l'image à afficher :

```
<image nom="photo.jpg" />
```

L'ordinateur comprend alors qu'il doit afficher l'image contenue dans le fichier photo.jpg.

Dans le cas d'une balise fonctionnant « par paire », on ne met les attributs que dans la balise ouvrante et pas dans la balise fermante. Par exemple, ce code indique que la citation est de Neil Armstrong et qu'elle date du 21 Juillet 1969 :

```
<citation auteur="Neil Armstrong" date="21/07/1969">
```

C'est un petit pas pour l'homme, mais un bond de géant pour l'humanité.

```
</citation>
```

Toutes les balises que nous venons de voir sont fictives. Les vraies balises ont des noms en anglais (eh oui !), nous allons les découvrir dans la suite de ce cours.

## Structure de base d'une page HTML5

Reprenons notre éditeur de texte (dans mon cas Sublime Text). Je vous invite à écrire ou à copier-coller le code source ci-dessous dans votre éditeur de texte. Ce code correspond à la base d'une page web en HTML5 :

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <title>Titre</title>

  </head>

  <body>

  </body>

</html>
```

J'ai mis des espaces au début de certaines lignes pour « décaler » les balises. Ce n'est pas obligatoire et cela n'a aucun impact sur l'affichage de la page, mais cela rend le code source plus lisible. On appelle cela l'**indentation**. Dans votre éditeur, il suffit d'appuyer sur la touche `Tab` pour avoir le même résultat.

Copié dans Sublime Text, vous devriez voir:



Code HTML5 minimal dans Sublime Text

Vous noterez que les balises s'ouvrent et se ferment dans un ordre précis. Par exemple, la balise `<html>` est la première que l'on ouvre et c'est aussi la dernière que l'on ferme (tout à la fin du code, avec `</html>`).

*Les balises doivent être fermées dans le sens inverse de leur ouverture.*

Un exemple :

- `<html><body></body></html>`: **correct**. Une balise qui est ouverte à l'intérieur d'une autre doit aussi être fermée à l'intérieur.
- `<html><body></html></body>`: **incorrect**, les balises s'entremêlent.

Explications sur toutes les balises que l'on vient de copier dans l'éditeur ?

### Le doctype

`<!DOCTYPE html>`

La toute première ligne s'appelle le **doctype**. Elle est indispensable car c'est elle qui indique qu'il s'agit bien d'une page web HTML.

Ce n'est pas vraiment une balise comme les autres (elle commence par un point d'exclamation). Vous pouvez considérer que c'est un peu l'exception qui confirme la règle.

Cette ligne du doctype était autrefois incroyablement complexe. Il était impossible de la retenir de tête. Pour XHTML 1.0, il fallait écrire :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Dans le cadre de HTML5, il a été décidé de la simplifier, pour le plus grand bonheur des webmasters. Quand vous voyez une balise doctype courte (`<!DOCTYPE html>`), cela signifie que la page est écrite en HTML5.

### La balise `</html>`

`<html>`

`</html>`

C'est la balise principale du code. Elle englobe tout le contenu de votre page. Comme vous pouvez le voir, la balise fermante `</html>` se trouve tout à la fin du code !

### L'en-tête `<head>` et le corps `<body>`

Une page web est constituée de deux parties :

- L'en-tête `<head>`: cette section donne quelques informations générales sur la page comme son titre, l'encodage (pour la gestion des caractères spéciaux), etc. Cette section est généralement assez courte. Les informations que contient l'en-tête ne sont pas affichées sur la page, ce sont simplement des informations générales à destination de l'ordinateur. Elles sont cependant très importantes !
- Le corps `<body>`: c'est là que se trouve la partie principale de la page. Tout ce que nous écrirons ici sera affiché à l'écran. C'est à l'intérieur du corps que nous écrirons la majeure partie de notre code.

Pour le moment, le corps est vide (nous y reviendrons plus loin). Intéressons-nous par contre aux deux balises contenues dans l'en-tête...



## L'encodage (charset)

```
<meta charset="utf-8" />
```

Cette balise indique l'encodage utilisé dans votre fichier .html.

Sans rentrer dans les détails, car cela pourrait vite devenir compliqué, l'encodage indique la façon dont le fichier est enregistré. C'est lui qui détermine comment les caractères spéciaux vont s'afficher (accents, idéogrammes chinois et japonais, caractères arabes, etc.).

Il y a plusieurs techniques d'encodage portant des noms bizarres et utilisées en fonction des langues : ISO-8859-1, OEM 775, Windows-1253... Une seule cependant devrait être utilisée aujourd'hui autant que possible : UTF-8. Cette méthode d'encodage permet d'afficher sans aucun problème pratiquement tous les symboles de toutes les langues de notre planète ! C'est pour cela que j'ai indiqué `utf-8` dans cette balise.

Il ne suffit pas de *dire* que votre fichier est en UTF-8. Il faut aussi que votre fichier soit bien *enregistré* en UTF-8. C'est heureusement le cas désormais par défaut dans la plupart des éditeurs de texte.

Si les accents s'affichent mal par la suite, c'est qu'il y a un problème avec l'encodage. Vérifiez que la balise meta indique bien UTF-8 et que votre fichier est enregistré en UTF-8 (sous Sublime Text, allez dans le menu File > Save with Encoding > UTF-8 pour vous assurer que votre fichier est enregistré en UTF-8.).

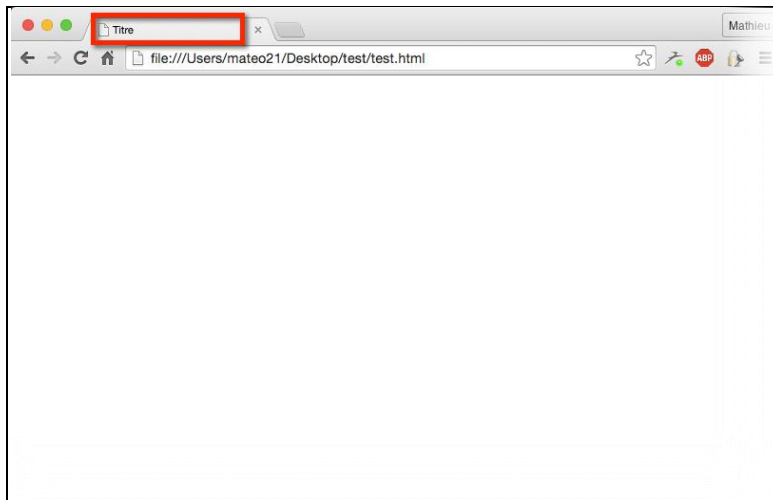
## Le titre principal de la page

```
<title>
```

C'est le titre de votre page, probablement l'élément le plus important ! Toute page doit avoir un titre qui décrit ce qu'elle contient.

Il est conseillé de garder le titre assez court (moins de 100 caractères en général).

Le titre ne s'affiche pas dans votre page mais en haut de celle-ci (souvent dans l'onglet du navigateur). Enregistrez votre page web et ouvrez-la dans votre navigateur. Vous verrez que le titre s'affiche dans l'onglet, comme sur la figure suivante.



Le titre de la page apparaît en haut du navigateur

Il faut savoir que le titre apparaît aussi dans les résultats de recherche, comme sur Google (figure suivante).



Le titre de la page apparaît dans les recherches Google

Autant vous dire que bien choisir son titre est important !

## Les commentaires

Nous avons appris à créer notre première *vraie* page HTML dans ce chapitre. Avant de terminer, j'aimerais vous présenter le principe des commentaires.

Un **commentaire** en HTML est un texte qui sert simplement de mémo. Il n'est pas affiché, il n'est pas lu par l'ordinateur, cela ne change rien à l'affichage de la page.

Cela sert à *vous* et aux personnes qui liront le code source de votre page. Vous pouvez utiliser les commentaires pour laisser des indications sur le fonctionnement de votre page.

Quel intérêt ? Cela vous permettra de vous rappeler comment fonctionne votre page si vous revenez sur votre code source après un long moment d'absence.

### Insérer un commentaire

Un commentaire est une balise HTML avec une forme bien spéciale :

```
<!-- Ceci est un commentaire -->
```

Vous pouvez le mettre où vous voulez au sein de votre code source : il n'a aucun impact sur votre page, mais vous pouvez vous en servir pour vous aider à vous repérer dans votre code source (surtout s'il est long).

```
<!DOCTYPE html>

<html>

  <head>

    <!-- En-tête de la page -->

    <meta charset="utf-8" />

    <title>Titre</title>

  </head>

  <body>

    <!-- Corps de la page -->

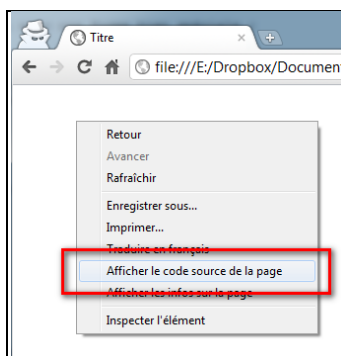
  </body>

</html>
```

***Tout le monde peut voir vos commentaires... et tout votre code HTML !***

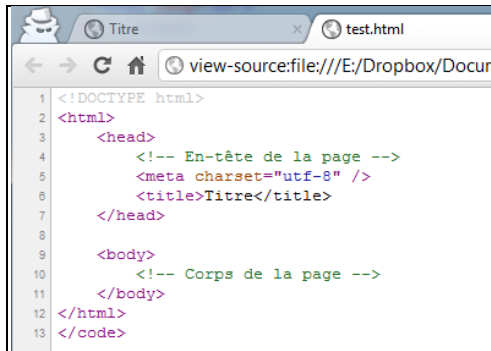
Terminons par une remarque importante : *tout le monde peut voir le code HTML de votre page* une fois celle-ci mise en ligne sur le Web.

Il suffit de faire un clic droit sur la page et de sélectionner « Afficher le code source de la page » (l'intitulé peut changer selon votre navigateur), comme le montre la figure suivante.



*Menu afficher le code source*

Le code source s'affiche alors (figure suivante).



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <!-- En-tête de la page -->
5     <meta charset="utf-8" />
6     <title>Titre</title>
7   </head>
8
9   <body>
10    <!-- Corps de la page -->
11  </body>
12 </html>
13 </code>
```

## Affichage du code source

Vous pouvez tester cette manipulation sur n'importe quel site web, cela marche ! Garanti à 100%. Cela s'explique assez facilement : le navigateur *doit* obtenir le code HTML pour savoir ce qu'il faut afficher. Le code HTML de tous les sites est donc public.

La morale de l'histoire ? Tout le monde pourra voir votre code HTML et vous ne pouvez pas l'empêcher. Par conséquent, ne mettez pas d'informations sensibles comme des mots de passe dans les commentaires... et soignez votre code source !

Lorsque vous regarderez le code de certains sites web, ne prenez pas peur s'il vous paraît long ou ne pas respecter les mêmes règles que celles que je vous présente dans ce livre. Tous les sites ne sont pas écrits en HTML5 (loin de là) et, parfois, certains webmasters rédigent très mal leur code, ce ne sont pas toujours des exemples à suivre !

## En résumé

- On utilise l'éditeur de texte (Sublime Text, Notepad++, jEdit, vim...) pour créer un fichier ayant l'extension `.html` (par exemple : `test.html`). Ce sera notre page web.
- Ce fichier peut être ouvert dans le navigateur web simplement en faisant un double-clic dessus.
- À l'intérieur du fichier, nous écrivons le contenu de notre page, accompagné de balises HTML.
- Les balises peuvent avoir plusieurs formes :
  - `<balise> </balise>`: elles s'ouvrent et se ferment pour délimiter le contenu (début et fin d'un titre, par exemple).
  - `<balise />`: balises orphelines (on ne les insère qu'en un seul exemplaire), elles permettent d'insérer un élément à un endroit précis (par exemple une image).
- Les balises sont parfois accompagnées d'attributs pour donner des indications supplémentaires (exemple : `<image nom="photo.jpg" />`).
- Une page web est constituée de deux sections principales : un en-tête (`<head>`) et un corps (`<body>`).
- On peut afficher le code source de n'importe quelle page web en faisant un clic droit puis en sélectionnant Afficher le code source de la page.

---

## Organisez votre texte

Bon, la page blanche c'est bien joli, mais votre site web risque d'avoir un succès mitigé si vous le laissez comme cela.

Nous allons découvrir de nombreuses balises HTML dans ce chapitre. Certaines existent depuis la toute première version de HTML, d'autres ont été introduites plus récemment dans HTML5.

Nous allons voir successivement dans ce chapitre :

- comment rédiger des paragraphes ;
- comment structurer sa page avec les titres ;
- comment donner de l'importance à certains mots de son texte ;
- comment organiser les informations sous forme de listes.

Motivés ? Allez, vous allez voir, ce n'est pas compliqué.

### Partie 1 : Les paragraphes

La plupart du temps, lorsqu'on écrit du texte dans une page web, on le fait à l'intérieur de paragraphes. Le langage HTML propose justement la balise `<p>` pour délimiter les paragraphes.

```
<p>Bonjour et bienvenue sur mon site !</p>
```

- `<p>` signifie « Début du paragraphe » ;
- `</p>` signifie « Fin du paragraphe ».

Comme je vous l'ai dit au chapitre précédent, on écrit le contenu du site web entre les balises `<body></body>`. Il nous suffit donc de mettre notre paragraphe entre ces deux balises et nous aurons enfin notre première vraie page web avec du texte !

Je reprends donc exactement le même code qu'au chapitre précédent et j'y ajoute mon paragraphe :

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <title>Paragraphe</title>

  </head>

  <body>

    <p>Bonjour et bienvenue sur mon site !</p>

  </body>

</html>
```

Essayez, vous allez voir le résultat !

Bon, c'est un bon début !

Mais ne nous arrêtons pas en si bon chemin. Nous allons voir maintenant quelque chose d'un peu particulier en HTML : le saut de ligne. Cela paraît simple et pourtant, cela ne fonctionne pas vraiment comme dans un traitement de texte habituel...

## Partie 2 : Sauter une ligne

En HTML, si vous appuyez sur la touche `Entrée`, cela ne crée pas une nouvelle ligne comme vous en avez l'habitude. Essayez donc ce code :

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <title>Essais de sauts de ligne</title>

  </head>

  <body>

    <p>Bonjour et bienvenue sur mon site !

    Ceci est mon premier test alors soyez indulgents s'il vous plaît, j'apprends petit à petit comment cela marche.

    Pour l'instant c'est un peu vide, mais revenez dans 2-3 jours quand j'aurai appris un peu plus de choses, je vous assure que vous allez être surpris !</p>

  </body>

</html>
```

Tout le texte s'affiche sur la même ligne alors qu'on est bien allé à la ligne dans le code !

Taper frénétiquement sur la touche `Entrée` dans l'éditeur de texte ne sert donc strictement à rien.

Comme vous devez vous en douter, il y a pourtant bien un moyen de faire des sauts de ligne en HTML.

En fait, si vous voulez écrire un deuxième paragraphe, il vous suffit d'utiliser une deuxième balise `<p>`. Votre code HTML devrait donc être au final rempli de balises de paragraphe !

Un exemple :

```
<!DOCTYPE html>

<html>

  <head>
```

```

<meta charset="utf-8" />

<title>Paragraphes</title>

</head>

<body>

  <p>Bonjour et bienvenue sur mon site !

  Ceci est mon premier test alors soyez indulgents s'il vous plaît, j'apprends petit à petit comment cela marche.</p>

  <p>Pour l'instant c'est un peu vide, mais revenez dans 2-3 jours quand j'aurai appris un peu plus de choses, je vous assure que vous allez être surpris !</p>

</body>

</html>

```

Le résultat se trouve à la figure suivante.



*Deux paragraphes avec 2 balises <p>*

Oui, mais si je veux juste aller à la ligne dans un paragraphe et non pas sauter une ligne ?

Eh bien , il existe une balise « Aller à la ligne » !

C'est une balise **orpheline** qui sert juste à indiquer qu'on doit aller à la ligne :<br />. Vous devez obligatoirement la mettre à l'intérieur d'un paragraphe.

Voici comment l'utiliser dans un code :

```

<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <title>Sauts de ligne</title>

  </head>

  <body>

    <p>Bonjour et bienvenue sur mon site !<br />

    Ceci est mon premier test alors soyez indulgents s'il vous plaît, j'apprends petit à petit comment cela marche.</p>

    <p>Pour l'instant c'est un peu vide, mais revenez dans 2-3 jours quand j'aurai appris un peu plus de choses, je vous assure que vous allez être surpris !</p>

  </body>

</html>

```



Vous pouvez théoriquement mettre plusieurs balises `<br />` d'affilée pour faire plusieurs sauts de lignes, mais on considère que c'est une mauvaise pratique qui rend le code délicat à maintenir.

Pour décaler un texte avec plus de précision, on utilisera le CSS, ce langage qui vient compléter le HTML et dont nous parlerons un peu plus loin.

Donc c'est compris ?

- `<p> </p>`: pour organiser son texte en paragraphes ;
- `<br />`: pour aller à la ligne.

Maintenant qu'on sait écrire des paragraphes, voyons voir comment on crée des titres.

### Partie 3 : Les titres

Lorsque le contenu de votre page va s'étoffer avec de nombreux paragraphes, il va devenir difficile pour vos visiteurs de se repérer. C'est là que les titres deviennent utiles.

En HTML, on est verni, on a le droit d'utiliser six niveaux de titres différents. Je veux dire par là qu'on peut dire « Ceci est un titre très important », « Ceci est un titre un peu moins important », « Ceci est un titre encore moins important », etc. On a donc six balises de titres différentes :

- `<h1> </h1>`: signifie « titre très important ». En général, on s'en sert pour afficher le titre de la page au début de celle-ci.
- `<h2> </h2>`: signifie « titre important ».
- `<h3> </h3>`: pareil, c'est un titre un peu moins important (on peut dire un « sous-titre » si vous voulez).
- `<h4> </h4>`: titre encore moins important.
- `<h5> </h5>`: titre pas important.
- `<h6> </h6>`: titre vraiment, mais alors là vraiment pas important du tout.

Attention : ne confondez pas avec la balise `<title>` ! La balise `<title>` affiche le titre de la page dans la barre de titre du navigateur comme nous l'avons vu.

Les titres `<h1>` et compagnie, eux, servent à créer des titres qui seront affichés *dans* la page web.

Ne vous laissez pas impressionner par toutes ces balises. En fait, six niveaux de titres, c'est beaucoup. Dans la pratique, personnellement, on n'utilise que les balises `<h1>`, `<h2>` et `<h3>`, et très rarement les autres.

Votre navigateur affiche le titre très important en très gros, le titre un peu moins important en un peu moins gros, etc.

Ne choisissez pas votre balise de titre en fonction de la taille qu'elle applique au texte ! Il faut impérativement bien structurer sa page en commençant par un titre de niveau 1 (`<h1>`), puis un titre de niveau 2 (`<h2>`), etc. Il ne devrait pas y avoir de sous-titre sans titre principal !

Si vous voulez modifier la taille du texte, sachez que nous apprendrons à faire cela en CSS un peu plus tard.

Essayez de créer une page web avec des titres pour voir ce que cela donne :

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <title>Niveaux de titres</title>

  </head>

  <body>

    <h1>Titre super important</h1>

    <h2>Titre important</h2>

    <h3>Titre un peu moins important (sous-titre)</h3>

    <h4>Titre pas trop important</h4>

    <h5>Titre pas important</h5>
```

```

    <h6>Titre vraiment pas important du tout</h6>

</body>

</html>

```

Allez, exemple d'utilisation des titres dans une page web :

```

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8" />

    <title>Présentation d'OpenClassrooms</title>

</head>

<body>

    <h1>Bienvenue sur OpenClassrooms !</h1>

    <p>Bonjour et bienvenue sur mon site : OpenClassrooms.<br />

    OpenClassrooms, qu'est-ce que c'est ?</p>

    <h2>Des cours pour débutants</h2>

    <p>OpenClassrooms vous propose des cours (tutoriels) destinés aux débutants : aucune connaissance n'est requise pour lire ces cours !</p>

    <p>Vous pourrez ainsi apprendre, sans rien y connaître auparavant, à créer un site web, à programmer, à construire des mondes en 3D !</p>

    <h2>Une communauté active</h2>

    <p>Vous avez un problème, un élément du cours que vous ne comprenez pas ? Vous avez besoin d'aide pour créer votre site ?<br />

    Rendez-vous sur les forums ! Vous y découvrirez que vous n'êtes pas le seul dans ce cas et vous trouverez très certainement quelqu'un qui vous aidera aimablement à résoudre votre problème.</p>

</body>

</html>

```

Voilà une page web qui prend forme !

Oui, mais moi je veux centrer mon titre, l'écrire en rouge et le souligner !

Nous ferons tout cela lorsque nous apprendrons le CSS (dès la deuxième partie du cours). Il faut savoir que `<h1>` ne signifie pas « Times New Roman, taille 16 pt », mais « *Titre important* ».

Grâce au langage CSS, vous pourrez dire « Je veux que mes titres importants soient centrés, rouges et soulignés ». Pour le moment, en HTML, nous ne faisons que structurer notre page. *Nous rédigeons le contenu avant de nous amuser à le mettre en forme.*

#### Partie 4 : La mise en valeur

Au sein de vos paragraphes, certains mots sont parfois plus importants que d'autres et vous aimeriez les faire ressortir. HTML vous propose différents moyens de mettre en valeur le texte de votre page.

##### Mettre un peu en valeur

Pour mettre *un peu* en valeur votre texte, vous devez utiliser la balise `<em>` `</em>`.

Son utilisation est très simple : encadrez les mots à mettre en valeur avec ces balises et c'est bon ! Reprendre peu l'exemple de tout à l'heure et y mettre quelques mots en évidence :

```

<!DOCTYPE html>

<html>

<head>

```

```

<meta charset="utf-8" />

<title>Emphase</title>

</head>

<body>

  <p>Bonjour et bienvenue sur mon site !<br />

  Ceci est mon premier test alors <em>soyez indulgents</em> s'il vous plaît, j'apprends petit à petit comment cela marche.</p>

</body>

</html>

```

Comme vous pouvez le voir, utiliser la balise `<em>` a pour conséquence de mettre le texte en *italique*.

En fait, c'est le navigateur qui choisit comment afficher les mots.

On lui dit que les mots sont assez importants et, pour faire ressortir cette information, il change l'apparence du texte en utilisant l'italique.

### Mettre bien en valeur

Pour mettre un texte bien en valeur, on utilise la balise `<strong>` qui signifie « fort », ou « important » si vous préférez.

Elle s'utilise exactement de la même manière que `<em>`:

```

<!DOCTYPE html>

<html>

<head>

  <meta charset="utf-8" />

  <title>Forte emphase</title>

</head>

<body>

  <p>Bonjour et bienvenue sur mon site !<br />

  Ceci est mon premier test alors <strong>soyez indulgents</strong> s'il vous plaît, j'apprends petit à petit comment cela marche.</p>

</body>

</html>

```

Vous voyez sûrement le texte s'afficher en gras. Là encore, le gras n'est qu'une *conséquence*. Le navigateur a choisi d'afficher en gras les mots importants pour les faire ressortir davantage.

La balise `<strong>` ne signifie pas « mettre en gras » mais « important ». On pourra décider plus tard, en CSS, d'afficher les mots « importants » d'une autre façon que le gras si on le souhaite.

### Marquer le texte

La balise `<mark>` permet de faire ressortir visuellement une portion de texte. L'extrait n'est pas forcément considéré comme important mais on veut qu'il se distingue bien du reste du texte. Cela peut être utile pour faire ressortir un texte pertinent après une recherche sur votre site par exemple.

```

<!DOCTYPE html>

<html>

<head>

  <meta charset="utf-8" />

  <title>Marquage du texte</title>

</head>

```

```
<body>

<p>Bonjour et bienvenue sur mon site !<br />

Ceci est mon premier test alors <mark>soyez indulgents</mark> s'il vous plaît, j'apprends petit à petit comment cela marche.</p>

</body>

</html>
```

Par défaut, `<mark>` a pour effet de surligner le texte. On pourra changer l'affichage en CSS (décider de surligner dans une autre couleur, d'encadrer le texte, etc.). C'est le même principe que ce que je vous disais pour les balises précédentes : elles indiquent le *sens* des mots et non pas comment ceux-ci doivent s'afficher.

### N'oubliez pas : HTML pour le fond, CSS pour la forme

Les balises `<em>`, `<strong>`, `<mark>`... ne servent pas à mettre en italique, en gras et à surligner du texte en HTML ! ».

Le rôle des balises est d'indiquer le *sens* du texte. Ainsi, `<strong>` indique à l'ordinateur « Ce texte est important ». C'est tout.

Et pour *montrer* que le texte est important, l'ordinateur décide de le mettre en gras (mais il pourrait aussi bien l'écrire en rouge !). La plupart des navigateurs affichent les textes importants en gras, mais rien ne les y oblige.

Je ne comprends pas. À quoi cela sert-il que l'ordinateur sache qu'un texte est important ?  
Il n'est pas assez intelligent pour comprendre !

Détrompez-vous ! De nombreux programmes analysent le code source des pages web, à commencer par les robots de moteurs de recherche. Ces robots parcourent le Web en lisant le code HTML de tous les sites. C'est le cas des robots de Google et de Bing, par exemple. Les mots-clés « importants » ont tendance à avoir plus de valeur à leurs yeux, donc si quelqu'un fait une recherche sur ces mots, il a plus de chances de tomber sur votre site.

Bien entendu, c'est une explication grossière et il ne faut pas croire qu'utiliser la balise `<strong>` à tout-va améliorera votre référencement. Il faut simplement faire confiance aux ordinateurs : ils comprennent ce qu'un texte « important » veut dire et peuvent se servir de cette information.

Mais alors, comment fait-on pour mettre spécifiquement en gras, pour écrire en rouge, et tout et tout ?  
Tout cela se fait en CSS. Souvenez-vous :

- le HTML définit le fond (contenu, logique des éléments) ;
- le CSS définit la forme (apparence).

Nous verrons le CSS plus loin, pour l'instant nous nous concentrons sur le HTML et ses balises, qui ont chacune un sens particulier.

### Les listes

Les listes nous permettent souvent de mieux structurer notre texte et d'ordonner nos informations.  
Nous allons découvrir ici deux types de listes :

- les listes non ordonnées ou listes à puces ;
- les listes ordonnées ou listes numérotées ou encore énumérations.

#### *Liste non ordonnée*

Une liste non ordonnée ressemble à ceci :

- Fraises
- Framboises
- Cerises

C'est un système qui nous permet de créer une liste d'éléments sans notion d'ordre (il n'y a pas de « premier » ni de « dernier »). Créer une liste non ordonnée est très simple. Il suffit d'utiliser la balise `<ul>` que l'on referme un peu plus loin avec `</ul>`. Commencez donc à taper ceci :

```
<ul></ul>
```

Et maintenant, on va écrire chacun des éléments de la liste entre deux balises `<li></li>`. Chacune de ces balises doit se trouver entre `<ul>` et `</ul>`. Vous allez comprendre de suite avec cet exemple :

```
<ul>

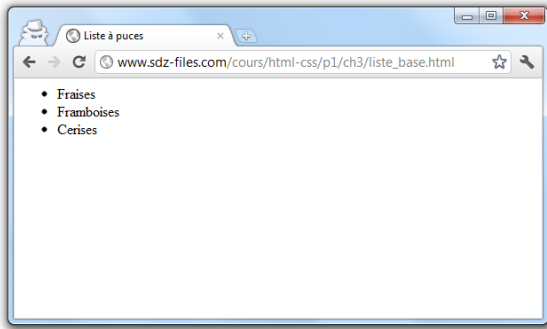
<li>Fraises</li>

<li>Framboises</li>

<li>Cerises</li>
```

</ul>

Le résultat se trouve à la figure suivante.



*Une liste non ordonnée*

Notez que la liste doit être placée à l'intérieur de `<body></body>`. À partir de maintenant, je ne mets pas tout le code de la page pour rester lisible.

Retenez donc ces deux balises :

- `<ul></ul>` délimite toute la liste ;
- `<li></li>` délimite un élément de la liste (une puce).

Vous pouvez mettre autant d'éléments que vous voulez dans la liste à puces, vous n'êtes pas limités à trois éléments.

Et voilà, vous savez créer une liste à puces ! Pas si dur une fois qu'on a compris comment imbriquer les balises.

Pour ceux qui ont besoin de faire des listes complexes, sachez que vous pouvez *imbriquer* des listes à puces (créer une liste à puces **dans** une liste à puces). Si vous voulez faire ça, ouvrez une seconde balise `<ul>` **à l'intérieur** d'un élément `<li></li>`.

Si vous fermez les balises dans le bon ordre, vous n'aurez pas de problème. Attention néanmoins, cette technique est un peu compliquée à maîtriser.

### *Liste ordonnée*

Une liste ordonnée fonctionne de la même façon, seule une balise change : il faut remplacer `<ul></ul>` par `<ol></ol>`. À l'intérieur de la liste, on ne change rien : on utilise toujours des balises `<li></li>` pour délimiter les éléments.

L'ordre dans lequel vous placez les éléments de la liste est important. Le premier `<li></li>` sera l'élément n° 1, le second sera le n°2 etc...

Comme c'est particulièrement intuitif, je vous laisse admirer la simplicité de cet exemple (résultat à la figure suivante) :

```
<h1>Ma journée</h1>

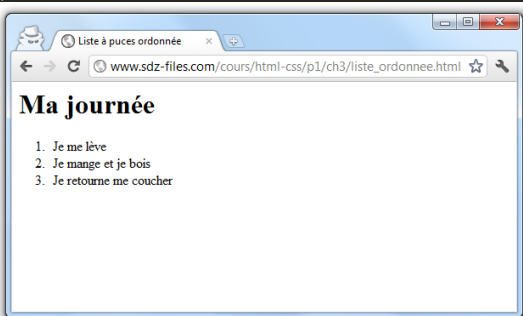
<ol>

  <li>Je me lève.</li>

  <li>Je mange et je bois.</li>

  <li>Je retourne me coucher.</li>

</ol>
```



*Une liste ordonnée*

Par rapport à l'exemple précédent, tout ce qu'on a eu à changer est donc la balise `<ol>`.

Pour information, il existe un troisième type de liste, beaucoup plus rare : la liste de définitions. Elle fait intervenir les balises `<dl>` (pour délimiter la liste), `<dt>` (pour délimiter un terme) et `<dd>` (pour délimiter la définition de ce terme).



## En résumé

- Le HTML comporte de nombreuses balises qui nous permettent d'organiser le texte de notre page. Ces balises donnent des indications comme « Ceci est un paragraphe », « Ceci est un titre », etc.
  - Les paragraphes sont définis par la balise<p> </p>et les sauts de ligne par la balise<br />.
  - Il existe six niveaux de titre, de<h1> </h1>à<h6> </h6>, à utiliser selon l'importance du titre.
  - On peut mettre en valeur certains mots avec les balises<strong>,<em>et<mark>.
  - Pour créer des listes, on doit utiliser la balise<ul>(liste à puces, non ordonnée) ou<ol>(liste ordonnée). À l'intérieur, on insère les éléments avec une balise<li>pour chaque item.
- 

## Créer des liens

Au chapitre précédent, vous avez appris à créer une page HTML toute simple. D'accord, elle n'était pas franchement magnifique, mais c'était une vraie page HTML quand même.

Comme vous le savez, un site web est composé de plusieurs pages. Comment faire pour aller d'une page vers une autre ? À l'aide de liens pardi ! Dans ce chapitre, nous allons justement apprendre à créer des liens entre nos pages.

Je suppose que chacun d'entre vous sait ce qu'est un lien : il s'agit d'un texte sur lequel on peut cliquer pour se rendre sur une autre page.

On peut faire un lien d'une page a.html vers une page b.html, mais on peut aussi faire un lien vers un autre site (par exemple,http://www.siteduzero.com). Dans les deux cas, nous allons voir que le fonctionnement est le même.

### Partie 1 :Un lien vers un autre site

Il est facile de reconnaître les liens sur une page : ils sont écrits d'une façon différente (par défaut, en bleu et soulignés) et un curseur en forme de main apparaît lorsqu'on pointe dessus.

Je vous propose d'essayer de coder le lien qui amène vers OpenClassrooms, comme à la figure suivante.

Bonjour. Vous souhaitez visiter [OpenClassrooms](#) ?  
C'est un bon site ;o)



*Lien vers OpenClassrooms*

Pour faire un lien, la balise que nous allons utiliser est très simple à retenir :<a>. Il faut cependant lui ajouter un attribut,href, pour indiquer vers quelle page le lien doit conduire.

Par exemple, le code ci-dessous est un lien qui amène vers OpenClassrooms, situé à l'adressehttps://openclassrooms.com:

```
<a href="https://openclassrooms.com">OpenClassrooms</a>
```

Nous allons placer ce lien au sein d'un paragraphe. Voici donc comment reproduire l'exemple de la figure précédente :

```
<p>Bonjour. Vous souhaitez visiter <a href="https://openclassrooms.com">OpenClassrooms</a>  
?<br />
```

```
C'est un bon site ;o)</p>
```

Par défaut, le lien s'affiche en bleu souligné. Si vous avez déjà ouvert la page, le lien s'affiche en violet.

Nous verrons comment changer cette apparence lorsque nous étudierons le CSS.

Si vous voulez faire un lien vers un autre site, il suffit donc de copier son adresse (on parle d'URL) en http://. Notez que certains liens commencent parfois parhttps://(sites sécurisés, comme OpenClassrooms) ou d'autres préfixes (ftp://,...).

Si vous faites un lien vers un site qui comporte une adresse un peu bizarre avec des &, comme

:http://www.site.com/?data=15&name=mateo21, vous devrez remplacer tous les « & » par « &amp; » dans votre lien comme ceci :http://www.site.com/?data=15&amp;name=mateo21.

Vous ne verrez pas la différence, mais cela est nécessaire pour avoir une page web correctement construite en HTML5.

Les liens que nous venons de voir sont appelés **liens absolus** car on indique l'adresse complète. Nous allons maintenant voir que l'on peut écrire les liens d'une façon un peu différente, ce qui va nous être utile pour faire des liens entre les pages de notre site.

## Partie 2 : Un lien vers une autre page de son site

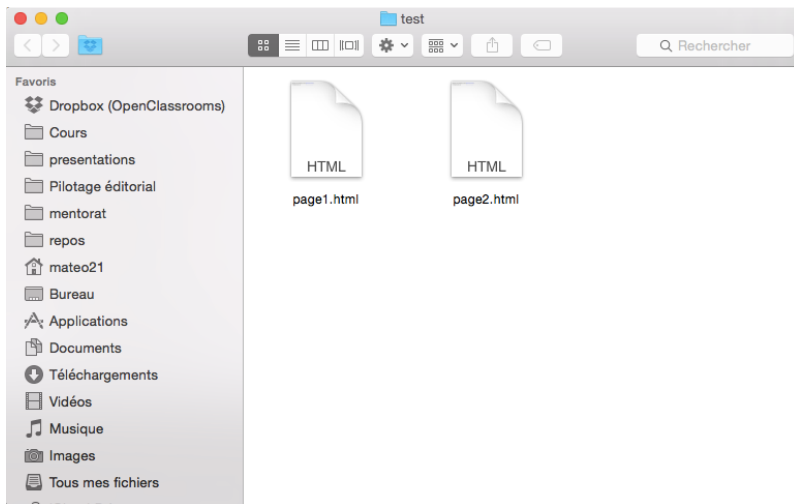
Nous venons d'apprendre à créer des liens vers des sites existants. Mais je suis sûr que vous aimeriez faire des liens entre les différentes pages de votre site, non ?

Oui, justement, comment je fais pour faire un lien vers une autre page de mon site ? Je ne connais pas son adresse en `http://...`, je commence à peine à créer mon site là ! Je n'ai pas d'adresse.

En effet, pour le moment, vous êtes en train de créer votre site sur votre ordinateur. Vous êtes le seul à pouvoir le voir et il n'a pas encore « d'adresse web » qui commence en `http://` comme la plupart des sites. Heureusement, cela ne va pas nous empêcher de travailler.

## Partie 3 : Deux pages situées dans un même dossier

Pour commencer, nous allons créer deux fichiers correspondant à deux pages HTML différentes. Comme je suis très inspiré, je vous propose de les appeler `page1.html` et `page2.html`. Nous aurons donc ces deux fichiers sur notre disque *dans le même dossier* (figure suivante).



*Plusieurs fichiers HTML dans un même dossier*

Comment faire un lien de la page 1 vers la page 2, sans avoir d'adresse en `http://` ? En fait, c'est facile : si les deux fichiers sont situés dans le même dossier, il suffit d'écrire comme cible du lien le nom du fichier vers lequel on veut amener. Par exemple : `<a href="page2.html">`. On dit que c'est un **lien relatif**.

Voici le code que nous allons utiliser dans nos fichiers `page1.html` et `page2.html`.

`page1.html`

```
<p>Bonjour. Souhaitez-vous consulter <a href="page2.html">la page 2</a> ?</p>
page2.html
```

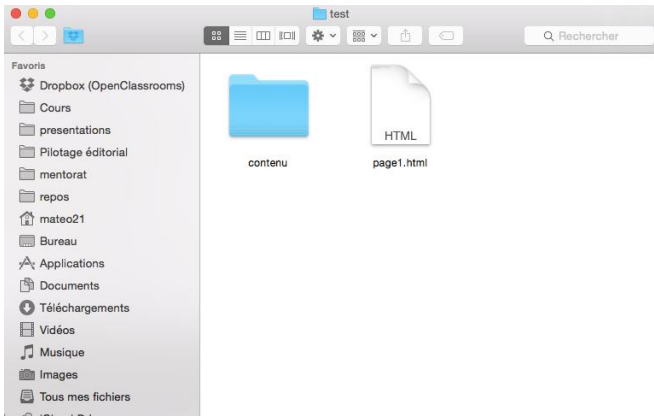
La page 2 (page d'arrivée) affichera simplement un message pour indiquer que l'on est bien arrivé sur la page 2 :

```
<h1>Bienvenue sur la page 2 !</h1>
```

## Partie 4 : Deux pages situées dans des dossiers différents

Les choses se corsent un petit peu si les pages sont situées dans des dossiers différents. Idéalement, elles ne devraient pas être trop loin l'une de l'autre (dans un sous-dossier par exemple).

Imaginons que `page2.html` se trouve dans un sous-dossier appelé `contenu`, comme à la figure suivante.



*Le fichier page2.html se trouve à l'intérieur du dossier contenu*

Dans ce cas de figure, le lien doit être rédigé comme ceci :

```
<a href="contenu/page2.html">
```

S'il y avait plusieurs sous-dossiers, on écrirait ceci :

```
<a href="contenu/autredossier/page2.html">
```

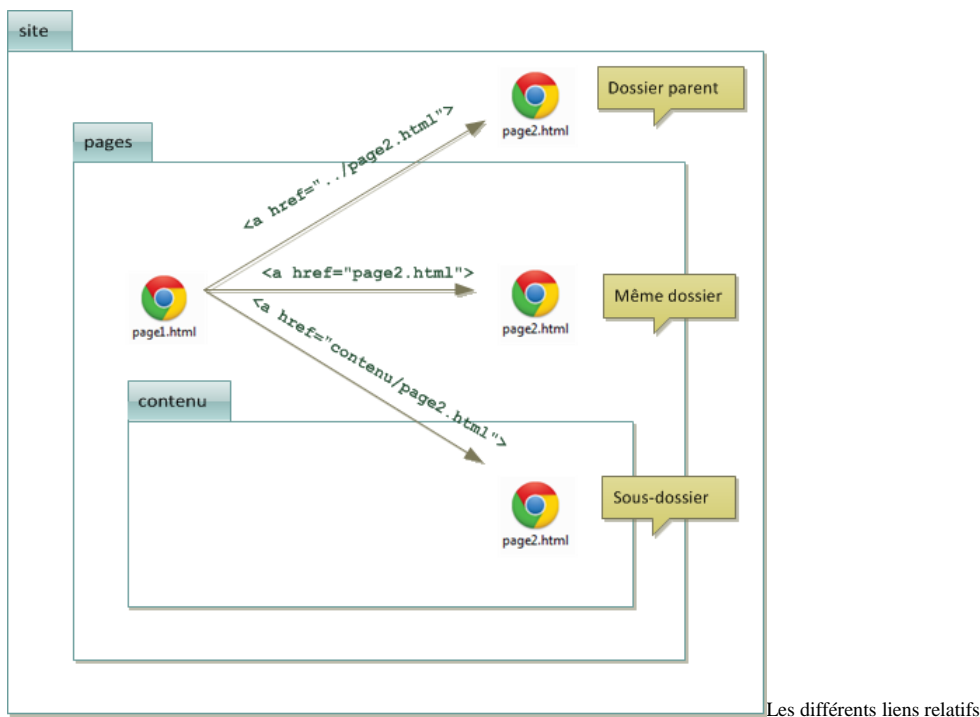
Et si le fichier ne se trouve pas dans un sous-dossier mais dans un dossier « parent », on fait comment ?

Si votre fichier cible est placé dans un dossier qui se trouve « plus haut » dans l'arborescence, il faut écrire deux points comme ceci :

```
<a href="../page2.html">
```

### Résumé en images

Les liens relatifs ne sont pas bien compliqués à utiliser une fois qu'on a compris le principe. Il suffit de regarder dans quel « niveau de dossier » se trouve votre fichier cible pour savoir comment écrire votre lien. La figure suivante fait la synthèse des différents liens relatifs possibles.



### Partie 5 : Un lien vers une ancre

Une **ancre** est une sorte de point de repère que vous pouvez mettre dans vos pages HTML lorsqu'elles sont très longues. En effet, il peut alors être utile de faire un lien amenant plus bas dans la même page pour que le visiteur puisse sauter directement à la partie qui l'intéresse.

Pour créer une ancre, il suffit de rajouter l'attribut `id` à une balise qui va alors servir de repère. Ce peut être n'importe quelle balise, un titre par exemple.

Utilisez l'attribut `id` pour donner un nom à l'ancre. Cela nous servira ensuite pour faire un lien vers cette ancre.

Par exemple :

```
<h2 id="mon_ancre">Titre</h2>
```

Ensuite, il suffit de créer un lien comme d'habitude, mais cette fois l'attribut `href` contiendra un dièse (#) suivi du nom de l'ancre. Exemple :

```
<a href="#mon_ancre">Aller vers l'ancre</a>
```

Normalement, si vous cliquez sur le lien, cela vous amènera plus bas dans la même page (à condition que la page comporte suffisamment de texte pour que les barres de défilement se déplacent automatiquement).

Voici un exemple de page comportant beaucoup de texte et utilisant les ancres (j'ai mis n'importe quoi dans le texte pour remplir) :

```
<h1>Ma grande page</h1>

<p>

    Aller directement à la partie traitant de :<br />

    <a href="#cuisine">La cuisine</a><br />

    <a href="#rollers">Les rollers</a><br />

    <a href="#arc">Le tir à l'arc</a><br />

</p>

<h2 id="cuisine">La cuisine</h2>

<p>... (beaucoup de texte) ...</p>

<h2 id="rollers">Les rollers</h2>

<p>... (beaucoup de texte) ...</p>

<h2 id="arc">Le tir à l'arc</h2>

<p>... (beaucoup de texte) ...</p>
```

S'il ne se passe rien quand vous cliquez sur les liens, c'est qu'il n'y a pas assez de texte. Dans ce cas, vous pouvez soit rajouter du blabla dans la page pour qu'il y ait (encore) plus de texte, soit réduire la taille de la fenêtre de votre navigateur pour faire apparaître les barres de défilement sur le côté.

L'attribut `id` sert à donner un nom « unique » à une balise, pour s'en servir de repère. Et, croyez-moi, vous n'avez pas fini d'entendre parler de cet attribut. Ici, on s'en sert pour faire un lien vers une ancre mais, en CSS, il nous sera très utile pour « repérer » une balise précise, vous verrez.

Évitez cependant de créer des `id` avec des espaces ou des caractères spéciaux, utilisez simplement, dans la mesure du possible, des lettres et chiffres pour que la valeur soit reconnue par tous les navigateurs.

## [Partie 6 : Lien vers une ancre située dans une autre page](#)

L'idée, c'est de faire un lien qui ouvre une autre page ET qui amène directement à une ancre située plus bas sur cette page. En pratique c'est assez simple à faire : il suffit de taper le nom de la page, suivi d'un dièse (#), suivi du nom de l'ancre.

Par exemple : `<a href="ancres.html#rollers">`

... vous amènera sur la page `ancres.html`, directement au niveau de l'ancre appelée `rollers`.

Voici une page qui contient trois liens, chacun amenant vers une des ancres de la page de l'exemple précédent :

```
<h1>Le Mégamix</h1>
```

```
<p>
```

```
Rendez-vous quelque part sur une autre page :<br />
```

```
<a href="ancres.html#cuisine">La cuisine</a><br />
```

```
<a href="ancres.html#rollers">Les rollers</a><br />
```

```
<a href="ancres.html#arc">Le tir à l'arc</a><br />
```

```
</p>
```

### Cas pratiques d'utilisation des liens

Je vais essayer de vous montrer ici quelques cas pratiques d'utilisation des liens. Par exemple, saviez-vous qu'il est très facile de faire des liens qui lancent un téléchargement ? Qui créent un nouvel e-mail ? Qui ouvrent une nouvelle fenêtre ?

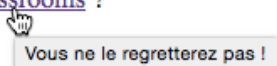
Non ? Eh bien nous allons voir tout cela ici.

#### Un lien qui affiche une infobulle au survol

Vous pouvez utiliser l'attribut `title` qui affiche une bulle d'aide lorsqu'on pointe sur le lien. Cet attribut est facultatif.

Vous aurez un résultat ressemblant à la figure suivante.

Souhaitez-vous visiter [OpenClassrooms](#) ?



*Une infobulle*

La bulle d'aide peut être utile pour informer le visiteur avant même qu'il n'ait cliqué sur le lien.

Voici comment reproduire ce résultat :

```
<p>Bonjour. Souhaitez-vous visiter <a href="https://openclassrooms.com" title="Vous ne le regretterez pas !">OpenClassrooms</a> ?</p>
```

#### Un lien qui ouvre une nouvelle fenêtre

Il est possible de « forcer » l'ouverture d'un lien dans une nouvelle fenêtre. Pour cela, on rajoutera `target="_blank"` à la balise `<a>`:

```
<p>Bonjour. Souhaitez-vous visiter <a href="https://openclassrooms.com" title="Vous ne le regretterez pas !" target="_blank">OpenClassrooms</a> ?</p>
```

Selon la configuration du navigateur, la page s'affichera dans une nouvelle fenêtre ou un nouvel onglet. Vous ne pouvez pas choisir entre l'ouverture d'une nouvelle fenêtre ou d'un nouvel onglet.

Notez cependant qu'il est déconseillé d'abuser de cette technique car elle perturbe la navigation. Le visiteur lui-même peut décider s'il veut ouvrir le lien dans une nouvelle fenêtre. Il fera **Ma**j + Clic sur le lien pour ouvrir dans une nouvelle fenêtre ou **C**trl + Clic pour ouvrir dans un nouvel onglet.

#### Un lien pour envoyer un e-mail

Si vous voulez que vos visiteurs puissent vous envoyer un e-mail, vous pouvez utiliser des liens de type `mailto`. Rien ne change au niveau de la balise, vous devez simplement modifier la valeur de l'attribut `href` comme ceci :

```
<p><a href="mailto:votrenom@bidule.com">Envoyez-moi un e-mail !</a></p>
```

Il suffit donc de faire commencer le lien par `mailto` : et d'écrire l'adresse e-mail où on peut vous contacter. Si vous cliquez sur le lien, un nouveau message vide s'ouvre, prêt à être envoyé à votre adresse e-mail.

#### Un lien pour télécharger un fichier

Beaucoup d'entre vous se demandent comment cela se passe pour le téléchargement d'un fichier... En fait, il faut procéder exactement comme si vous faisiez un lien vers une page web, mais en indiquant cette fois le nom du fichier à télécharger.

Par exemple, supposez que vous vouliez faire télécharger `monfichier.zip`. Placez simplement ce fichier dans le même dossier que votre page web (ou dans un sous-dossier) et faites un lien vers ce fichier :

```
<p><a href="monfichier.zip">Télécharger le fichier</a></p>
```

C'est tout ! Le navigateur, voyant qu'il ne s'agit pas d'une page web à afficher, va lancer la procédure de téléchargement lorsqu'on cliquera sur le lien.

## En résumé

- Les liens permettent de changer de page et sont, par défaut, écrits en bleu et soulignés.
  - Pour insérer un lien, on utilise la balise `<a>` avec l'attribut `href` pour indiquer l'adresse de la page cible. Exemple : `<a href="https://openclassrooms.com">`.
  - On peut faire un lien vers une autre page de son site simplement en écrivant le nom du fichier : `<a href="page2.html">`.
  - Les liens permettent aussi d'amener vers d'autres endroits sur la même page. Il faut créer une ancre avec l'attribut `id` pour « marquer » un endroit dans la page, puis faire un lien vers l'ancre comme ceci : `<a href="#ancre">`.
- 

## Insérer une image dans une page web ?

Vous allez voir, c'est d'une facilité déconcertante... Enfin presque. Il existe différents **formats** d'image que l'on peut utiliser sur des sites web, et on ne doit pas les choisir au hasard. En effet, les images sont parfois volumineuses à télécharger, ce qui ralentit le temps de chargement de la page (beaucoup plus que le texte !).

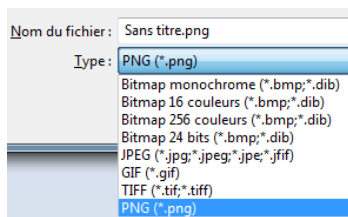
Pour faire en sorte que vos pages restent lisibles et rapides à télécharger, suivez donc activement mes conseils !

### Partie 1 : Les différents formats d'images

Savez-vous ce qu'est un format d'image ?

Quand vous avez une image « entre les mains », vous avez la possibilité de l'enregistrer dans plusieurs « formats » différents. Le poids (en Ko, voire en Mo) de l'image sera plus ou moins élevé selon le format choisi et la qualité de l'image va changer.

Par exemple, le logiciel de dessin Paint (même si c'est loin d'être le meilleur) vous propose de choisir entre plusieurs formats lorsque vous enregistrez une image (figure suivante).



*Différents formats d'image proposés par Paint*

Certains formats sont plus adaptés que d'autres selon l'image (photo, dessin, image animée...). Notre but ici est de faire le tour des différents formats utilisés sur le Web pour que vous les connaissiez et sachiez choisir celui qui convient le mieux à votre image. Rassurez-vous, il n'y a pas beaucoup de formats différents, cela ne sera donc pas bien long.

Toutes les images diffusées sur Internet ont un point commun : elles sont **compressées**. Cela veut dire que l'ordinateur fait des calculs pour qu'elles soient moins lourdes et donc plus rapides à charger.

### Le JPEG

Les images au format JPEG (*Joint Photographic Expert Group*) sont très répandues sur le Web. Ce format est conçu pour réduire le poids des photos (c'est-à-dire la taille du fichier associé), qui peuvent comporter plus de 16 millions de couleurs différentes. La figure suivante est une photo enregistrée au format JPEG.



*Une photo de montagne en JPEG*

Les images JPEG sont enregistrées avec l'extension `.jpg` ou `.jpeg`.

Notez que le JPEG détériore un peu la qualité de l'image, d'une façon généralement imperceptible. C'est ce qui le rend si efficace pour réduire le poids des photos.

Quand il s'agit d'une photo, on ne peut généralement pas détecter la perte de qualité. Par contre, si ce n'est pas une photo, vous risquez de voir l'image un peu « baver ». Dans ce cas, il vaut mieux utiliser le format PNG.

### **Le PNG**

Le format PNG (*Portable Network Graphics*) est le plus récent de tous. Ce format est adapté à la plupart des graphiques (je serais tenté de dire « à tout ce qui n'est pas une photo »). Le PNG a deux gros avantages : il peut être rendu transparent et il n'altère pas la qualité de l'image.

Le PNG a été inventé pour concurrencer un autre format, le GIF, à l'époque où il fallait payer des royalties pour pouvoir utiliser des GIF. Depuis, le PNG a bien évolué et c'est devenu le format le plus puissant pour enregistrer la plupart des images.

Le PNG existe en deux versions, en fonction du nombre de couleurs que doit comporter l'image :

- PNG 8 bits : 256 couleurs ;
- PNG 24 bits : 16 millions de couleurs (autant qu'une image JPEG).

La figure suivante est une image PNG en 24 bits, représentant Zozor, qui sera notre mascotte tout au long de ce cours.



*Zozor en PNG*

Au fait, si le PNG 24 bits peut afficher autant de couleurs qu'une image JPEG, et qu'en plus il peut être rendu transparent sans modifier la qualité de l'image... quel est l'intérêt du JPEG ?

La compression du JPEG est plus puissante sur les photos. Une photo enregistrée en JPEG se chargera toujours beaucoup plus vite que si elle était enregistrée en PNG. Je vous conseille donc toujours de réserver le format JPEG aux photos.

### **Le GIF**

C'est un format assez vieux, qui a été néanmoins très utilisé (et qui reste très utilisé par habitude). Aujourd'hui, le PNG est globalement bien meilleur que le GIF : les images sont généralement plus légères et la transparence est de meilleure qualité. Je vous recommande donc d'utiliser le PNG autant que possible.

Le format GIF est limité à 256 couleurs (alors que le PNG peut aller jusqu'à plusieurs millions de couleurs).

Néanmoins, le GIF conserve un certain avantage que le PNG n'a pas : il peut être animé. D'où l'explosion ces dernières années des GIF animés sur le web (aussi appelé "*reaction gifs*").

Un GIF animé



## *Il existe un format adapté à chaque image*

Si on résume, voici quel format adopter en fonction de l'image que vous avez :

- **Une photo** : utilisez un JPEG.
- **N'importe quel graphique avec peu de couleurs** (moins de 256) : utilisez un PNG 8 bits ou éventuellement un GIF.
- **N'importe quel graphique avec beaucoup de couleurs** : utilisez un PNG 24 bits.
- **Une image animée** : utilisez un GIF animé.

## *Les erreurs à éviter !!!!*

Bannissez les autres formats

Les autres formats non cités ici, comme le format BITMAP (\*.bmp) sont à bannir car bien souvent ils ne sont pas compressés, donc trop gros. Ils ne sont pas du tout adaptés au Web. On peut en mettre sur son site mais le chargement sera vraiment *extrêmement* long !

Choisissez bien le nom de votre image

Si vous voulez éviter des problèmes, prenez l'habitude d'enregistrer vos fichiers avec des noms en minuscules, sans espace ni accent, par exemple : mon\_image.png.

Vous pouvez remplacer les espaces par le caractère *underscore* (« \_ ») comme je l'ai fait ici.

## Partie 2 : Insérer une image

Revenons maintenant au code HTML pour découvrir comment placer des images dans nos pages web !

### Insertion d'une image

Quelle est la fameuse balise qui va nous permettre d'insérer une image ? Il s'agit de... `<img />`!

C'est une balise de type **orpheline** (comme `<br />`). Cela veut dire qu'on n'a pas besoin de l'écrire en deux exemplaires comme la plupart des autres balises que nous avons vues jusqu'ici. En effet, nous n'avons pas besoin de délimiter une portion de texte, nous voulons juste insérer une image à un endroit précis.

La balise doit être accompagnée de deux attributs obligatoires :

- **src** : il permet d'indiquer où se trouve l'image que l'on veut insérer. Vous pouvez soit mettre un chemin absolu (ex. : `http://www.site.com/fleur.png`), soit mettre le chemin en relatif (ce qu'on fait le plus souvent). Ainsi, si votre image est dans un sous-dossier `images`, vous devrez taper : `src="images/fleur.png"`
- **alt** : cela signifie « texte alternatif ». On doit *toujours* indiquer un texte alternatif à l'image, c'est-à-dire un court texte qui décrit ce que contient l'image. Ce texte sera affiché à la place de l'image si celle-ci ne peut pas être téléchargée (cela arrive), ou dans les navigateurs de personnes handicapées (non-voyants) qui ne peuvent malheureusement pas « voir » l'image. Cela aide aussi les robots des moteurs de recherche pour les recherches d'images. Pour la fleur, on mettrait par exemple : `alt="Une fleur"`.

Les images doivent se trouver obligatoirement à l'intérieur d'un paragraphe (`<p></p>`). Voici un exemple d'insertion d'image :

```
<p>
Voici une photo que j'ai prise lors de mes dernières vacances à la montagne :<br />

</p>
```

Bref, l'insertion d'image est quelque chose de très facile pour peu qu'on sache indiquer où se trouve l'image, comme on avait appris à le faire avec les liens.

La plus grosse « difficulté » (si on peut appeler cela une difficulté) consiste à choisir le bon format d'image. Ici, c'est une photo donc c'est évidemment le format JPEG qu'on utilise.

Je le répète : évitez à tout prix les accents, majuscules et espaces dans vos noms de fichiers et de dossiers. Voici un chemin qui va poser problème :

"Images du site/Image toute bête.jpg".

Il faudrait supprimer les espaces (ou les remplacer par le symbole « \_ »), supprimer les accents et tout mettre en minuscules comme ceci :

"images\_du\_site/image\_toute\_bete.jpg".

Sachez donc que, si votre image ne s'affiche pas, c'est très certainement parce que le chemin est incorrect ! Simplifiez au maximum vos noms de fichiers et de dossiers, et tout ira bien.

### Partie 3 : Ajouter une infobulle

L'attribut permettant d'afficher une bulle d'aide est le même que pour les liens : il s'agit de `title`. Cet attribut est facultatif (contrairement à `alt`).

Voici ce que cela peut donner :

```
<p>

Voici une photo que j'ai prise lors de mes dernières vacances à la montagne :<br />



</p>
```

Surveillez la photo avec la souris pour voir l'infobulle apparaître.

### Partie 4 : Miniature cliquable

Si votre image est très grosse, il est conseillé d'en afficher la miniature sur votre site. Ajoutez ensuite un lien sur cette miniature pour que vos visiteurs puissent afficher l'image en taille originale.

De nombreux sites permettent de redimensionner des images, comme [ResizeImage.net](https://resizeimage.net) par exemple. Je vais ainsi disposer de deux versions de ma photo, comme à la figure suivante) : la miniature et l'image d'origine.



Je place toutes les deux dans un dossier appelé par exemple `img`. J'affiche la version `montagne_mini.jpg` sur ma page et je fais un lien vers `montagne.jpg` pour que l'image agrandie s'affiche lorsqu'on clique sur la miniature.

Voici le code HTML que je vais utiliser pour cela :

```
<p>

Vous souhaitez voir l'image dans sa taille d'origine ? Cliquez dessus !<br />

<a href="img/montagne.jpg"></a>

</p>
```

Parfois, certains navigateurs choisissent d'afficher un cadre bleu (ou violet) pas très esthétique autour de votre image cliquable.

Heureusement, nous pourrions retirer ce cadre dans peu de temps grâce au CSS.

### Partie 5 : Les figures

Au cours de la lecture de ce livre, vous avez déjà rencontré plusieurs fois des **figures**. Ce sont des éléments qui viennent enrichir le texte pour compléter les informations de la page.

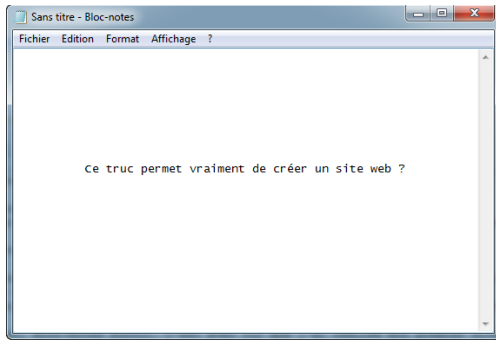
Les figures peuvent être de différents types :

- images ;
- codes source ;
- citations ;
- etc.

Bref, tout ce qui vient *illustrer* le texte est une figure. Nous allons ici nous intéresser aux images mais, contrairement à ce qu'on pourrait croire, les figures ne sont pas *forcément* des images : un code source aussi illustre le texte.

#### Création d'une figure

Reprenons par exemple cette capture d'écran du premier chapitre, représentée à la figure suivante.



*Le logiciel Bloc-Notes*

En HTML5, on dispose de la balise `<figure>`. Voici comment on pourrait l'utiliser :

```
<figure>

</figure>
```

Une figure est le plus souvent accompagnée d'une légende. Pour ajouter une légende, utilisez la balise `<figcaption>` à l'intérieur de la balise `<figure>`, comme ceci :

```
<figure>

  <figcaption>Le logiciel Bloc-Notes</figcaption>

</figure>
```

### **Bien comprendre le rôle des figures**

Un peu plus tôt dans ce chapitre, je vous ai dit que les images devaient être situées dans des paragraphes (placées à l'intérieur d'une balise `<p></p>`). Ce n'est pas tout à fait vrai.

Si vous faites de votre image une figure, l'image peut être située en-dehors d'un paragraphe.

```
<p>Connaissez-vous le logiciel Bloc-Notes ? On peut faire des sites web avec !</p>

<figure>

  <figcaption>Le logiciel Bloc-Notes</figcaption>

</figure>
```

Je ne vois pas vraiment de changement. Quand dois-je placer mon image dans un paragraphe et quand dois-je la placer dans une figure ?

Bonne question ! Tout dépend de ce que votre image apporte au texte :

- Si elle n'apporte aucune information (c'est juste une illustration pour décorer) : placez l'image dans un paragraphe.
- Si elle apporte une information : placez l'image dans une figure.

La balise `<figure>` a un rôle avant tout **sémantique**. Cela veut dire qu'elle indique à l'ordinateur que l'image a du sens et qu'elle est importante pour la bonne compréhension du texte. Cela peut permettre à un programme de récupérer toutes les figures du texte et de les référencer dans une table des figures, par exemple.

Enfin, sachez qu'une figure peut très bien comporter plusieurs images. Voici un cas où cela se justifie :

```
<figure>

  <figcaption>Logos des différents navigateurs</figcaption>

</figure>
```

### En résumé





- Il existe plusieurs formats d'images adaptées au Web :
  - JPEG : pour les photos ;
  - PNG : pour toutes les autres illustrations ;
  - GIF : similaire au PNG, plus limité en nombre de couleurs mais qui peut être animé.
- On insère une image avec la balise `<img />`. Elle doit obligatoirement comporter au moins ces deux attributs : `src`(nom de l'image) et `alt`(courte description de l'image).
- Si une image illustre le texte (et n'est pas seulement décorative), il est conseillé de la placer au sein d'une balise `<figure>`. La balise `<figcaption>` permet d'écrire la légende de l'image.

## QCM

Maintenant, il est temps d'évaluer vos connaissances sur cette première partie !  
Bon courage !





### • Question 1

Quel est le rôle du navigateur web ?

- ☐  Ecrire du code en HTML et CSS
- ☐  Traduire le code HTML et CSS en un résultat visuel
- ☐  Se connecter au réseau Wifi
- ☐  Fournir votre site web à vos visiteurs





### • Question 2

Lequel de ces éléments rencontre-t-on en premier dans un code HTML ?

- ☐  `<html>`
- ☐  `<head>`
- ☐  `<!-- BEGIN -->`
- ☐  `<!DOCTYPE html>`

### • Question 3

Avec lequel de ces langages peut-on définir le fond du menu de navigation d'un site en gris ?

- ☐  HTML
- ☐  CSS
- ☐  JavaScript
- ☐  PHP

### • Question 4





Le contenu de l'une de ces balises correspond au titre principal de votre page. Il est très important pour les moteurs de recherche. Lequel ?

- ☐  `<title>`
- ☐  `<meta>`
- ☐  `<h1>`
- ☐  `<h2>`

## • Question 5

---

Laquelle de ces balises indique le contenu de la plus grande importance ?

- ☐  <strong>
- ☐  <p>
- ☐  <i>
- ☐  <em>

## • Question 6

---





L'un de ces formats d'image est particulièrement adapté au stockage des photos. Lequel ?

- ☐  PNG
- ☐  JPEG
- ☐  GIF
- ☐  BMP

## • Question 7

---





Pourquoi l'attribut **alt** est-il obligatoire pour les images ?

- ☐  Parce qu'il indique la taille de l'image
- ☐  Parce qu'il permet de conserver la compatibilité avec les vieux navigateurs
- ☐  Parce qu'il indique ce que contient l'image pour les non-voyants et les moteurs de recherche
- ☐  Cet attribut n'est pas obligatoire : seul **src** est requis

## • Question 8

---





Comment fait-on télécharger un fichier à un visiteur en HTML ?

- ☐  On fait un lien vers le fichier avec la balise <a>
- ☐  On utilise la balise <download>
- ☐  On insère le contenu du fichier à télécharger directement dans le code HTML
- ☐  On ne peut pas faire cela en HTML

## • Question 9

---


Quelle balise délimite une liste à puce numérotée ?




- ☐  <dl>
- ☐  <ol>
- ☐  <li>
- ☐  <ul>

## • Question 10

---

Lequel de ces liens permet d'amener le visiteur à un endroit précis de la même page web ayant l'identifiant details ?

- ☐  <a href=".details">

- ☐  <a href="details">
- ☐  <a href="/details">
- ☐  <a href="#details">

## À vous de jouer !

### Exercice 1 : p 46 Hatier

A l'aide de la page web et image associée, retrouver les différentes parties du langage html.

Utiliser des couleurs différentes pour repérer commentaires, corps de la page, titres par ordre d'importance, listes, paragraphes et images.

### Exercice 2 :

Pour cet exercice, vous allez devoir créer votre première page web en HTML à partir de ce que vous venez d'apprendre dans la première partie du cours.

L'objectif est de créer **votre CV en ligne**. Il ne remportera pas le prix du design de l'année pour le moment, mais attendez de voir la suite du cours qu'on l'améliore !

Voici ce que je souhaite voir apparaître sur votre CV :

- Votre nom, en titre principal
- Une courte phrase qui vous décrit, ou une citation, en-dessous du nom
- Une photo miniature, sur laquelle on pourra cliquer pour avoir une version agrandie
- 3 sections délimitées par des titres différents :
  - Mon expérience
  - Mes compétences
  - Ma formation

Pensez à bien utiliser des balises de titre, de paragraphe et des listes à puces lorsque cela vous semble nécessaire.

## Mettre en place le CSS

Après avoir passé toute une première partie du cours à ne travailler que sur le HTML, nous allons maintenant découvrir le CSS. Le CSS n'est pas plus compliqué que le HTML. Il vient le compléter pour vous aider à mettre en forme votre page web.

Dans ce premier chapitre sur le CSS, nous allons voir la théorie sur le CSS : qu'est-ce que c'est ? À quoi cela ressemble-t-il ? Où est-ce qu'on écrit du code CSS ?

Ces aspects théoriques ne sont pas bien compliqués mais vous devez obligatoirement les connaître car c'est la base du CSS. C'est d'ailleurs la seule chose que je vous demanderai de retenir par cœur en CSS, vous pourrez retrouver le reste dans le mémo en annexe.

### Partie 1 : La petite histoire du CSS

Nous avons déjà bien entamé notre découverte du HTML, même s'il reste encore de nombreuses choses à apprendre (nous y reviendrons dans quelques chapitres). En revanche, il est temps maintenant de nous intéresser au CSS.

CSS (*Cascading Style Sheets*), c'est cet autre langage qui vient compléter le HTML. Vous vous souvenez de son rôle ? *Gérer la mise en forme de votre site.*

### Partie 2 : Petit rappel : à quoi sert CSS ?

CSS ? C'est lui qui vous permet de choisir la couleur de votre texte.  
Lui qui vous permet de sélectionner la police utilisée sur votre site.  
Lui encore qui permet de définir la taille du texte, les bordures, le fond...

Et aussi, c'est lui qui permet de faire la mise en page de votre site. Vous pourrez dire : je veux que mon menu soit à gauche et occupe telle largeur, que l'en-tête de mon site soit calé en haut et qu'il soit toujours visible, etc.

Souvenez-vous de ce petit comparatif que nous avons vu dès le premier chapitre (figure suivante).

HTML  
(pas de CSS)



HTML + CSS



La même page HTML, sans et avec CSS

([www.csszengarden.com](http://www.csszengarden.com))

Grâce au HTML, nous avons pu rédiger le contenu de notre site mais il est brut de décoffrage. Le CSS vient compléter ce code pour mettre en forme tout cela et donner au contenu l'apparence que l'on souhaite.

### Partie 3 : CSS : des débuts difficiles

Il faut savoir qu'aux débuts du Web, CSS n'existait pas. En fait, il n'y avait initialement que le langage HTML. Le HTML est né en 1991 et CSS en 1996.

Alors, vous vous dites sûrement : comment faisait-on la mise en forme de 1991 à 1996 ?

Eh bien, uniquement en HTML ! Il y avait en effet des balises HTML dédiées à la mise en forme. `<fontcolor="#aabl1c3">`, par exemple, permettait de définir la couleur du texte.

Cependant, les pages HTML commençaient à devenir assez complexes. Il y avait de plus en plus de balises et c'était un joyeux mélange entre le fond et la forme, qui rendait la mise à jour des pages web de plus en plus complexe. C'est pour cela que l'on a créé le langage CSS.

Cependant, le CSS n'a pas été adopté immédiatement par les webmasters, loin de là. Il fallait se défaire de certaines mauvaises habitudes et cela a pris du temps. Encore aujourd'hui, on peut trouver des sites web avec des balises HTML de mise en forme, anciennes et obsolètes, comme `<font>` !

### Partie 4 : CSS : la prise en charge des navigateurs

Tout comme le HTML, le CSS a évolué. Je vous avais indiqué qu'il y avait quatre versions importantes de CSS :

- CSS 1 ;
- CSS 2.0 ;
- CSS 2.1 ;
- CSS 3.

Ce sont les navigateurs web qui font le travail le plus complexe : ils doivent *lire* le code CSS et *comprendre* comment afficher la page.

Au début des années 2000, Internet Explorer était le navigateur le plus répandu mais sa gestion du CSS est longtemps restée assez médiocre (pour ne pas dire carrément mauvaise). C'était la grande époque de la version 6 (IE6).

Depuis, de nombreux navigateurs sont arrivés et ont chahuté Internet Explorer : Mozilla Firefox bien sûr, mais aussi Google Chrome. Et je ne vous parle pas du succès des Mac et iPhone avec leur navigateur Safari. Cela a incité Microsoft à réagir et publier (après une longue période d'inactivité) IE 7, puis IE 8 et IE 9, 10, 11... Et maintenant Edge.

Bon, ton cours d'histoire, c'est bien joli mais en quoi cela me concerne-t-il aujourd'hui ?



*Que faut-il retenir de tout cela ?* Que les navigateurs ne connaissent pas toutes les propriétés CSS qui existent. Plus le navigateur est vieux, moins il connaît de fonctionnalités CSS.

Nous allons voir un certain nombre de fonctionnalités de CSS qui ne marchent pas forcément sur les navigateurs les plus vieux. Je ne peux pas l'éviter, c'est comme cela : *aucun navigateur ne connaît parfaitement toutes les fonctionnalités CSS* de toute façon ! Au pire, si le navigateur ne connaît pas une propriété CSS, il l'ignore et ne met pas en forme, mais cela ne fait pas planter votre page : celle-ci sera donc toujours lisible.

Je vous recommande fortement de mettre dans vos favoris le site [www.caniuse.com](http://www.caniuse.com) qui propose des tables de compatibilité des fonctionnalités de HTML et CSS sur différents navigateurs (et sur leurs différentes versions) :

# CSS3 Border images - CR

Method of using images for borders

Global: 83.02% + 10.59% = 93.61%  
unprefixed: 83.01%  
France: 90.88% + 3.81% = 94.69%  
unprefixed: 90.88%

Current aligned Usage relative Show all

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8			31					4.1	
9		38	43					4.3	
10		39	44			7.1		4.4.4	
11	12	40	45	8	31	8.4	8	44	44
	13	41	46	9	32	9			
		42	47		33				
		43	48						

Notes Known issues (1) Resources (3) Feedback

Note that both the `border-style` and `border-width` must be specified (not set to `none` or `0`) for border-images to work according to spec, though older implementations may not have this requirement. Partial support refers to supporting the shorthand syntax, but not the individual properties (`border-image-source`, `border-image-slice`, etc).

Table de compatibilité CSS sur [caniuse.com](http://caniuse.com)

## Partie 5 : Où écrit-on le CSS ?

Vous avez le choix car on peut écrire du code en langage CSS à trois endroits différents :

- dans un fichier `.css` (méthode la plus recommandée) ;
- dans l'en-tête `<head>` du fichier HTML ;
- directement dans les balises du fichier HTML via un attribut `style` (méthode la moins recommandée).

Je vais vous présenter ces trois méthodes mais sachez d'ores et déjà que la première... est la meilleure.

## Partie 6 : Dans un fichier .css (recommandé)

Comme je viens de vous le dire, on écrit le plus souvent le code CSS dans un fichier spécial ayant l'extension `.css` (contrairement aux fichiers HTML qui ont l'extension `.html`). C'est la méthode la plus pratique et la plus souple. Cela nous évite de tout mélanger dans un même fichier. J'utiliserai cette technique dans toute la suite de ce cours.

Commençons à pratiquer dès maintenant ! Nous allons partir du fichier HTML suivant :

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

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

    <title>Premiers tests du CSS</title>

  </head>

  <body>

    <h1>Mon super site</h1>
```

```

<p>Bonjour et bienvenue sur mon site !</p>

<p>Pour le moment, mon site est un peu <em>vide</em>. Patientez encore un peu !</p>

</body>

</html>

```

Vous noterez le contenu de la ligne 5, `<link rel="stylesheet" href="style.css" />`: c'est elle qui indique que ce fichier HTML est associé à un fichier appelé `style.css` et chargé de la mise en forme.

Enregistrez ce fichier sous le nom que vous voulez (par exemple `page.html`). Pour le moment, rien d'extraordinaire à part la nouvelle balise que nous avons ajoutée.

Maintenant, créez un *nouveau* fichier vide dans votre éditeur de texte (par exemple Sublime Text) et copiez-y ce bout de code CSS (rassurez-vous, je vous expliquerai tout à l'heure ce qu'il veut dire) :

```

p
{
    color: blue;
}

```

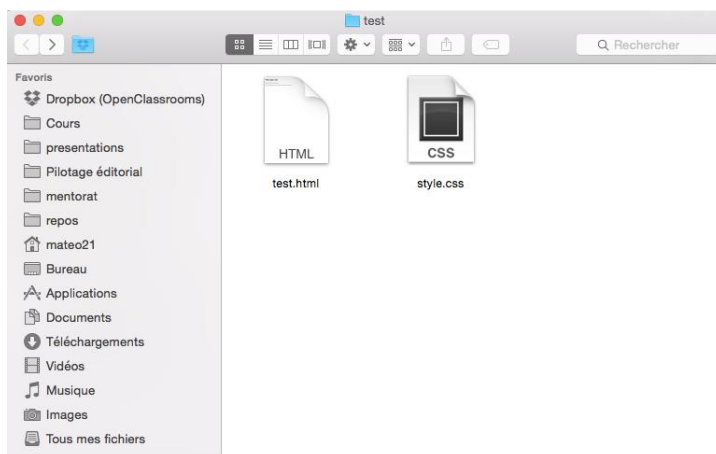
Pour obtenir la coloration du code dans Sublime Text, enregistrez bien votre fichier avec l'extension `.css` d'abord. Enregistrez le fichier en lui donnant un nom qui se termine par `.css`, comme `style.css`. Placez ce fichier `.css` dans le même dossier que votre fichier `.html`.

Dans Sublime Text, vous devriez observer quelque chose de similaire à la figure suivante.



*Fichiers HTML et CSS dans Sublime Text*

Dans votre explorateur de fichiers, vous devriez les voir apparaître côte à côte. D'un côté le `.html`, de l'autre le `.css`, comme à la figure suivante.



*Fichiers HTML et CSS dans l'explorateur de fichiers*

Ouvrez maintenant votre fichier `page.html` dans votre navigateur pour le tester, comme vous le faites d'habitude. Regardez, c'est magique : vos paragraphes sont écrits en bleu, comme dans la figure suivante !



*Le texte est écrit en bleu*

Il est inutile d'ouvrir directement le fichier `style.css` dans le navigateur. Il faut ouvrir le fichier `page.html` (il fera automatiquement appel au fichier `style.css`).

### Partie 7 : Dans l'en-tête `<head>` du fichier HTML

Il existe une autre méthode pour utiliser du CSS dans ses fichiers HTML : cela consiste à insérer le code CSS directement dans une balise `<style>` à l'intérieur de l'en-tête `<head>`.

Voici comment on peut obtenir exactement le même résultat avec un seul fichier `.html` qui contient le code CSS (lignes 5 à 10) :

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <style>

      p
      {
        color: blue;
      }

    </style>

    <title>Premiers tests du CSS</title>

  </head>

  <body>

    <h1>Mon super site</h1>

    <p>Bonjour et bienvenue sur mon site !</p>

    <p>Pour le moment, mon site est un peu <em>vide</em>. Patientez encore un peu !</p>

  </body>

</html>
```

Testez, vous verrez que le résultat est le même.

### Partie 7 : Directement dans les balises (non recommandé)

Dernière méthode, à manipuler avec précaution : vous pouvez ajouter un attribut `style` à n'importe quelle balise. Vous insérerez votre code CSS directement dans cet attribut :

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <title>Premiers tests du CSS</title>

  </head>

  <body>

    <h1>Mon super site</h1>

    <p style="color: blue;">Bonjour et bienvenue sur mon site !</p>

    <p>Pour le moment, mon site est un peu <em>vide</em>. Patientez encore un peu !</p>

  </body>

</html>
```

Cette fois, seul le texte du premier paragraphe (ligne 11), dont la balise contient le code CSS, sera coloré en bleu (figure suivante).



*Le premier paragraphe est écrit en bleu*

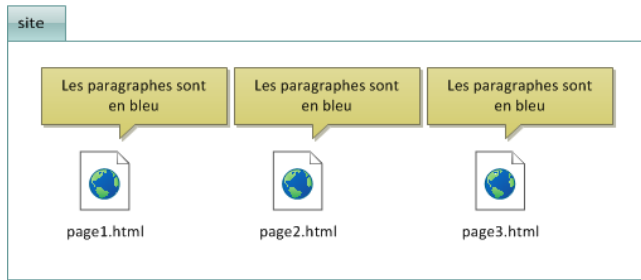
### Partie 8 : Quelle méthode choisir ?

Je trouve que la première méthode que tu recommandes est plus compliquée que les deux autres ! Pourquoi nous conseilles-tu de créer deux fichiers, j'étais bien, moi, avec juste un fichier `.html` !

Je vous recommande fortement de prendre l'habitude de travailler avec la première méthode parce que c'est celle utilisée par la majorité des webmasters... Pourquoi ?

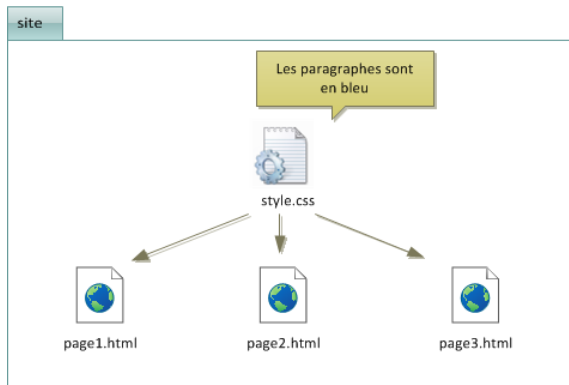
Pour le moment, vous faites vos tests sur un seul fichier HTML. Cependant, votre site sera plus tard constitué de plusieurs pages HTML, on est d'accord ?

Imaginez : si vous placez le code CSS directement dans le fichier HTML, il faudra copier ce code dans tous les fichiers HTML de votre site ! Et si demain vous changez d'avis, par exemple si vous voulez que vos paragraphes soient écrits en rouge et non en bleu, il faudra modifier chaque fichier HTML un à un, comme le montre la figure suivante.



*Le code CSS est répété dans chaque fichier HTML*

Si vous travaillez avec un fichier CSS externe, vous n'aurez besoin d'écrire cette instruction qu'une seule fois pour tout votre site, comme le montre la figure suivante.



*Le code CSS est donné une fois pour toutes dans un fichier CSS*

## Partie 9 : Appliquer un style : sélectionner une balise

Maintenant que nous savons où placer le code CSS, intéressons-nous de plus près à ce code. Je vous ai donné, sans vous l'expliquer, un premier bout de code CSS :

```
p
{
  color: blue;
}
```

Dans un code CSS comme celui-ci, on trouve trois éléments différents :

- **Des noms de balises** : on écrit les noms des balises dont on veut modifier l'apparence. Par exemple, si je veux modifier l'apparence de tous les paragraphes `<p>`, je dois écrire `p`.
- **Des propriétés CSS** : les « effets de style » de la page sont rangés dans des propriétés. Il y a par exemple la propriété `color` qui permet d'indiquer la couleur du texte, `font-size` qui permet d'indiquer la taille du texte, etc. Il y a beaucoup de propriétés CSS et, comme je vous l'ai dit, je ne vous obligerai pas à les connaître toutes par cœur.
- **Les valeurs** : pour chaque propriété CSS, on doit indiquer une valeur. Par exemple, pour la propriété `color`, il faut indiquer le nom de la couleur. Pour `font-size`, il faut indiquer quelle taille on veut, etc.

Schématiquement, une feuille de style CSS ressemble donc à cela :

```
balise1
{
  propriete1: valeur1;
  propriete2: valeur2;
  propriete3: valeur3;
}

balise2
{
  propriete1: valeur1;
  propriete2: valeur2;
```

```

propriete3: valeur3;

propriete4: valeur4;
}

balise3
{

    propriete1: valeur1;
}

```

Vous repérez dans cet extrait de code les balises, propriétés et valeurs dont je viens de vous parler.

Comme vous le voyez, on écrit le nom de la balise (par exemple `h1`) et on ouvre des accolades pour, à l'intérieur, mettre les propriétés et valeurs que l'on souhaite. On peut mettre autant de propriétés que l'on veut à l'intérieur des accolades. Chaque propriété est suivie du symbole « deux-points » ( `:` ) puis de la valeur correspondante. Enfin, chaque ligne se termine par un point-virgule ( `;` ).

Je vous apprendrai de nombreuses propriétés dans les chapitres suivants. Pour le moment, dans les exemples, on va juste changer la couleur pour s'entraîner.

Le code CSS que nous avons utilisé jusqu'ici :

```

p
{

    color: blue;
}

```

... signifie donc en français : « *Je veux que tous mes paragraphes soient écrits en bleu.* ». Le résultat est visible à la figure suivante.



*Paragraphes écrits en bleu*

Essayez de changer le nom de la balise affectée par le code CSS. Par exemple, si j'écris `h1`, c'est le titre qui sera écrit en bleu. Modifiez votre fichier `style.css` comme ceci :

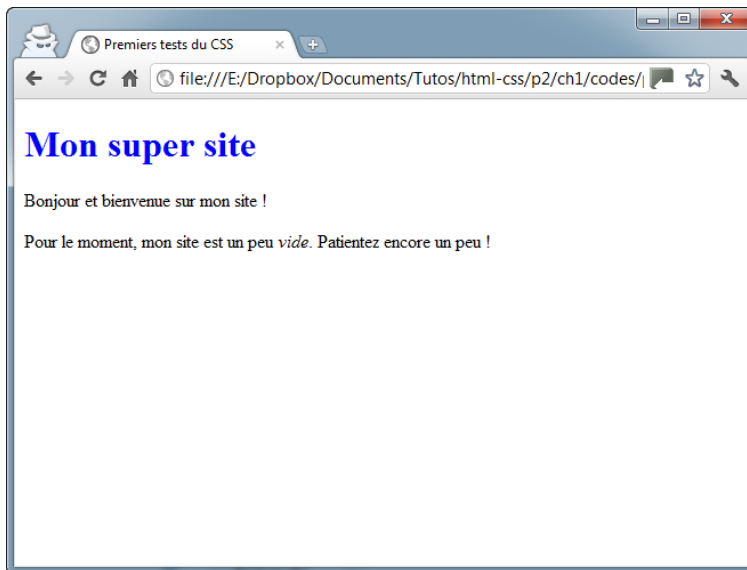
```

h1
{

    color: blue;
}

```

Maintenant, ouvrez à nouveau votre page HTML (souvenez-vous, c'est la page HTML qu'on ouvre dans le navigateur, pas le fichier CSS !) : vous devriez voir son titre s'afficher en bleu (figure suivante) !



*Titre écrit en bleu*

### Partie 10 : Appliquer un style à plusieurs balises

Prenons le code CSS suivant :

```
h1
{
    color: blue;
}

em
{
    color: blue;
}
```

Il signifie que nos titres `<h1>` et nos textes importants `<em>` doivent s'afficher en bleu. Par contre, c'est un peu répétitif, vous ne trouvez pas ?

Heureusement, il existe un moyen en CSS d'aller plus vite si les deux balises doivent avoir la même présentation. Il suffit de combiner la déclaration en séparant les noms des balises par une virgule, comme ceci :

```
h1, em
{
    color: blue;
}
```

Le résultat se trouve à la figure suivante.





*Titre et texte important écrits en bleu*

Cela signifie : « Je veux que le texte de mes `<h1>` et `<em>` soit écrit en bleu ».

Vous pouvez indiquer autant de balises à la suite que vous le désirez.

### Partie 11 : Des commentaires dans du CSS

Comme en HTML, il est possible de mettre des commentaires. Les commentaires ne seront pas affichés, ils servent simplement à indiquer des informations pour vous, par exemple pour vous y retrouver dans un looong fichier CSS.

D'ailleurs, vous allez vous en rendre compte, en général le fichier HTML est assez court et la feuille CSS assez longue (si elle contient tous les éléments de style de votre site, c'est un peu normal). Notez qu'il est possible de créer plusieurs fichiers CSS pour votre site si vous ressentez le besoin de séparer un peu votre code CSS (en fonction des différentes sections de votre site, par exemple).

... De quoi on parlait déjà ? Ah oui, les commentaires en CSS.

Donc, pour faire un commentaire, c'est facile ! Tapez `/*`, suivi de votre commentaire, puis `*/` pour terminer votre commentaire.

Vos commentaires peuvent être écrits sur une ou plusieurs lignes. Par exemple :

```
/*
style.css
-----

Par Mathieu Nebra
*/

p
{
    color: blue; /* Les paragraphes seront en bleu */
}
```

Il est possible que j'utilise les commentaires dans la suite du cours, pour vous donner des explications à l'intérieur même des fichiers `.css`.

### Partie 12 : Appliquer un style : class et id

Ce que je vous ai montré jusqu'ici a quand même un défaut : cela implique par exemple que TOUS les paragraphes possèdent la même présentation (ici, ils seront donc tous écrits en bleu).

Comment faire pour que certains paragraphes seulement soient écrits d'une manière différente ? On pourrait placer le code CSS dans un attribut `style` sur la balise que l'on vise (c'est la technique que je vous ai présentée un peu plus tôt) mais, comme je vous l'ai dit, ce n'est pas recommandé (il vaut mieux utiliser un fichier CSS externe).

Pour résoudre le problème, on peut utiliser ces attributs spéciaux *qui fonctionnent sur toutes les balises* :

- l'attribut `class`;
- l'attribut `id`.

Que les choses soient claires dès le début : les attributs `class` et `id` sont quasiment identiques. Il y a seulement une petite différence que je vous dévoilerai plus bas.

Pour le moment, et pour faire simple, on ne va s'intéresser qu'à l'attribut `class`.

Comme je viens de vous le dire, c'est un attribut que l'on peut mettre sur n'importe quelle balise, aussi bien titre que paragraphe, image, etc.

```
<h1 class=""> </h1>

<p class=""> </p>

<img class="" />
```

Oui mais que met-on comme valeur à l'attribut `class`?

En fait, vous devez écrire un nom qui sert à identifier la balise. Ce que vous voulez, du moment que le nom commence par une lettre.

Par exemple, je vais associer la classe `introduction` à mon premier paragraphe (ligne 12) :

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

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

    <title>Premiers tests du CSS</title>

  </head>

  <body>

    <h1>Mon super site</h1>

    <p class="introduction">Bonjour et bienvenue sur mon site !</p>

    <p>Pour le moment, mon site est un peu <em>vide</em>. Patientez encore un peu !</p>

  </body>

</html>
```

Maintenant que c'est fait, votre paragraphe est identifié. Il a un nom : `introduction`. Vous allez pouvoir réutiliser ce nom dans le fichier CSS pour dire : « *Je veux que seules les balises qui ont comme nom 'introduction' soient affichées en bleu* ».

Pour faire cela en CSS, indiquez le nom de votre classe en commençant par un point, comme ci-dessous :

```
.introduction
{
  color: blue;
}
```

Testez le résultat : seul votre paragraphe appelé `introduction` va s'afficher en bleu (figure suivante) !



Seul le premier paragraphe s'affiche en bleu

Et l'attribut `id` alors ?

Lui, il fonctionne exactement de la même manière que `class`, à un détail près : il ne peut être utilisé *qu'une fois* dans le code.

Quel intérêt ? Il y en a assez peu pour tout vous dire, cela vous sera utile si vous faites du JavaScript plus tard pour reconnaître certaines balises. D'ailleurs, nous avons déjà vu l'attribut `id` dans le chapitre sur les liens (pour réaliser des ancres). En pratique, nous ne mettrons des `id` que sur des éléments qui sont uniques dans la page, comme par exemple le logo :

```

```

Si vous utilisez des `id`, lorsque vous définirez leurs propriétés dans le fichier CSS, il faudra faire précéder le nom de l'`id` par un dièse (#) :

```
#logo
{
    /* Indiquez les propriétés CSS ici */
}
```

Je ne vous propose pas de le tester, cela fonctionne exactement comme `class`.

Si vous vous emmêlez les pinceaux entre `class` et `id` retenez que deux balises peuvent avoir le même nom avec l'attribut `class`. Un nom d'`id` doit en revanche être unique dans la page HTML.

### Partie 13 : Les balises universelles

Il arrivera parfois que vous ayez besoin d'appliquer une `class` (ou un `id`) à certains mots qui, à l'origine, ne sont pas entourés par des balises.

En effet, le problème de `class`, c'est qu'il s'agit d'un attribut. Vous ne pouvez donc en mettre que sur une balise. Si, par exemple, je veux modifier uniquement « bienvenue » dans le paragraphe suivant :

```
<p>Bonjour et bienvenue sur mon site !</p>
```

Cela serait facile à faire s'il y avait une balise autour de « bienvenue » mais, malheureusement il n'y en a pas. Par chance, on a inventé... la balise-qui-ne-sert-à-rien.

En fait, on a inventé deux balises dites **universelles**, qui n'ont aucune signification particulière (elles n'indiquent pas que le mot est important, par exemple). Il y a une différence minime (mais significative !) entre ces deux balises :

- `<span> </span>`: c'est une balise de type **inline**, c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte, pour sélectionner certains mots uniquement. Les balises `<strong>` et `<em>` sont de la même famille. Cette balise s'utilise donc au milieu d'un paragraphe et c'est celle dont nous allons nous servir pour colorer « bienvenue ».
- `<div> </div>`: c'est une balise de type **block**, qui entoure un bloc de texte. Les balises `<p>`, `<h1>`, etc. sont de la même famille. Ces balises ont quelque chose en commun : elles créent un nouveau « bloc » dans la page et provoquent donc obligatoirement un retour à la ligne. `<div>` est une balise fréquemment utilisée dans la construction d'un design, comme nous le verrons plus tard.

Pour le moment donc, nous allons utiliser plutôt la balise `<span>`. On la met autour de « bienvenue », on lui ajoute une classe (du nom qu'on veut), on crée le CSS et c'est gagné !

```
<p>Bonjour et <span class="salutations">bienvenue</span> sur mon site !</p>

.salutations
{
    color: blue;
}
```

Vous pouvez voir le résultat à la figure suivante.



*Le mot « bienvenue » est écrit en bleu*

### Appliquer un style : les sélecteurs avancés

En CSS, le plus difficile est de savoir cibler le texte dont on veut changer la forme. Pour cibler (on dit « sélectionner ») les éléments de la page à modifier, on utilise ce qu'on appelle des **sélecteurs**. Vous en avez déjà utilisé quelques-uns un peu plus tôt dans ce chapitre, résumons-les pour commencer.

### Les sélecteurs que vous connaissez déjà

Ces sélecteurs, que nous avons vus précédemment, sont de loin les plus couramment utilisés. Il faut les connaître par cœur. Commençons par la base de la base :

```
p
{
}
}
```

... signifie « Je veux toucher tous les paragraphes ». Après, c'est à vous de dire ce que vous faites à ces paragraphes (vous les écrivez en bleu, par exemple).

Nous avons aussi vu :

```
h1, em
{
}
}
```

... qui signifie « Tous les titres et tous les textes importants ». Nous avons sélectionné deux balises d'un coup.

Et enfin, nous avons vu comment sélectionner des balises précises à qui nous avons donné un nom grâce aux attributs `class` et `id`:

```
.class
{
}
}
```

```
#id
```

```
{  
  
}
```

Vous savez quoi ? Il existe des dizaines d'autres façons de cibler des balises en CSS ! Nous n'allons pas toutes les voir car il y en a beaucoup et certaines sont complexes, mais voici déjà de quoi vous permettre d'être plus efficaces en CSS !

### Les sélecteurs avancés

\* : sélecteur universel

```
*  
  
{  
  
}
```

Sélectionne toutes les balises sans exception. On l'appelle le sélecteur universel.

A B : une balise contenue dans une autre

```
h3 em  
  
{  
  
}
```

Sélectionne toutes les balises `<em>` situées à l'intérieur d'une balise `<h3>`. Notez qu'il n'y a pas de virgule entre les deux noms de balises.

Exemple de code HTML correspondant :

```
<h3>Titre avec <em>texte important</em></h3>
```

A + B : une balise qui en suit une autre

```
h3 + p  
  
{  
  
}
```

Sélectionne la première balise `<p>` située après un titre `<h3>`.

Exemple :

```
<h3>Titre</h3>
```

```
<p>Paragraphe</p>
```

A[attribut] : une balise qui possède un attribut

```
a[title]  
  
{  
  
}
```

Sélectionne tous les liens `<a>` qui possèdent un attribut `title`.

Exemple :

```
<a href="http://site.com" title="Infobulle">
```

A[attribut="Valeur"] : une balise, un attribut et une valeur exacte

```
a[title="Cliquez ici"]  
  
{
```

```
}
```

Idem, mais l'attribut doit en plus avoir exactement pour valeur « Cliquez ici ».

Exemple :

```
<a href="http://site.com" title="Cliquez ici">
```

A[attribut\*="Valeur"] : une balise, un attribut et une valeur

```
a[title*="ici"]
```

```
{
```

```
}
```

Idem, l'attribut doit cette fois contenir dans sa valeur le mot « ici » (peu importe sa position).

Exemple :

```
<a href="http://site.com" title="Quelque part par ici">
```

[D'autres sélecteurs existent !](#)

Je ne vous ai présenté ici qu'une partie des sélecteurs CSS mais sachez qu'il en existe beaucoup d'autres. Si vous voulez une liste complète, vous pouvez vous renseigner directement à la source : sur le [site du W3C](#) ! C'est très complet.

Sachez que nous découvrirons certains de ces autres sélecteurs dans la suite de ce cours !

### En résumé

- CSS est un autre langage qui vient compléter le HTML. Son rôle est de mettre en forme votre page web.
- Il faut être vigilant sur la compatibilité des navigateurs avec certaines fonctionnalités récentes de CSS3. Quand un navigateur ne connaît pas une instruction de mise en forme, il l'ignore simplement.
- On peut écrire le code CSS à plusieurs endroits différents, le plus conseillé étant de créer un fichier séparé portant l'extension `.css` (exemple : `style.css`).
- En CSS, on sélectionne quelles portions de la page HTML on veut modifier et on change leur présentation avec des propriétés CSS :

```
balise1
```

```
{
```

```
    propriete1: valeur1;
```

```
    propriete2: valeur2;
```

```
}
```

- Il existe de nombreuses façons de sélectionner la portion de la page que l'on veut mettre en forme. Par exemple, on peut viser :
  - toutes les balises d'un même type, en écrivant simplement leur nom (`h1` par exemple) ;
  - certaines balises spécifiques, auxquelles on a donné des noms à l'aide des attributs `class` ou `id` (`.nomclasse` ou `#nomid`) ;
  - uniquement les balises qui se trouvent à l'intérieur d'autres balises (`h3 em`).
  - etc.

---

## Activité 7 : Mini projet : créer son Site WEB

### Cahier de charges :

L'objectif est de créer un mini-site WEB écrit en langage HTML

*Le site doit contenir :*

- Au moins 3 pages reliées entre elles par des liens (par exemple [suivant](#) , [retour](#) ).
  - Veillez à ce que la première page soit nommée **index**
- Une ou plusieurs images.

- Un lien vers un moteur de recherche (Possibilité d'intégrer la barre de recherche d'un moteur de recherche)
- Un ou plusieurs lien(s) externe(s) vers un site web ou vers une vidéo complétant le sujet traité.

**Mise en page :**

La mise en forme du site sera gérée à l'aide d'un fichier CSS.

Il n'est pas attendu une mise en forme graphique trop ambitieuse !!!

**Choix du thème à traiter :** Le site WEB doit contenir une réflexion autour d'un des thèmes suivant :

- **L'évolution des métiers induite par le numérique.**
- **Les réseaux sociaux**

*Histoire, caractéristiques, nombre d'abonnés, propriété de l'information, etc ...*

- **Intelligence artificielle**

*Apprentissage automatique, impact sur l'homme, etc...*

- **Données dans les nuages « Cloud »**

*Support de stockage, partage de données, consommation énergétique, etc ...*

- **L'évolution des outils et des supports d'écriture et de lecture**

*Les machines, la réalité augmentée, etc ...*

- **Réparation et augmentation des capacités du corps humain**

### **III. Moteurs de recherche et publication d'informations**

Un moteur de recherche est un service qui permet aux internautes de rechercher du contenu via le web. Un utilisateur y entre des mots ou phrases clés et reçoit une liste de résultats de sites web, d'images, de vidéos ou d'autres données en ligne. La liste du contenu renvoyé à un utilisateur par un moteur de recherche est appelée « page de résultats ».

Chaque jour, nous effectuons des recherches en ligne, mais pour faire des recherches efficaces, il est préférable d'utiliser une variété de stratégies de recherche plutôt qu'une seule source d'informations et ainsi exercer son esprit critique par rapport aux résultats.

**Activité :** chercher un lieu à visiter ou un événement culturel pour une sortie de classe.

Trouver un lieu gratuit ou peu coûteux, qui possède les caractéristiques suivantes : lieu ou événement culturel, ouvert pendant les jours et heures de classe, groupes scolaires acceptés, accessible aux handicapés.

### **Aide :**

**S**électionner des questions et des outils de recherche : élaborer les questions essentielles de la recherche et choisir les outils de recherche.

**E**xtraire des mots clés et des termes : trouver des mots clés efficaces en surlignant les principaux termes de leurs questions.

**A**ppliquer des stratégies de recherche : appliquer certaines des stratégies de recherche, par exemple, mettre des mots entre guillemets ou préciser le type d'information désiré.

**R**echercher : effectuer la recherche avec plusieurs sources.

**C**hoisir de consigner la recherche : pour éviter des recherches déjà faites, consigner le sujet cherché et les sources consultées.

### **Indexation du web**

Les sociétés qui proposent des moteurs de recherche doivent disposer de serveurs très puissants et de capacités mémoire très importantes, car l'indexation demande d'enregistrer une copie de toutes les pages web accessibles au niveau mondial.

**Indexation** : traitement qui consiste à analyser des pages pour y détecter des mots clés utilisés fréquemment dans les demandes des internautes, puis à fabriquer un index permettant de trouver rapidement des pages à partir de mots clés.

Tout ceci est calculé à l'avance, ce qui permet aux moteurs de recherche de répondre en un temps très court quand un nouvel utilisateur effectue une nouvelle demande. La demande est en fait rarement nouvelle : d'autres l'avaient déjà probablement effectuée avant.

### **Calcul de la popularité :**

La principale difficulté des moteurs de recherche consiste à classer, de la manière la plus pertinente possible l'ensemble des pages contenant les mots clés demandés, pour choisir quelles pages présenter en premier.

- Les moteurs de recherche se font concurrence sur ces algorithmes de classement.
- L'algorithme de PageRank calcule la popularité d'une page à partir de la popularité des pages qui la citent. L'idée est de faire confiance aux auteurs des pages web qui citent une autre page.

**Popularité** : plus une page est citée, plus la popularité est grande.

### **Activité :**

**JEU : ACTIVITE 10 P61**

**Calcul de la popularité d'une page web : TP 2 HATIER p 47**

## **IV. Sécuriser la navigation**

Un navigateur internet (ou navigateur web) est un logiciel informatique permettant de consulter des pages Web. Il s'agit donc d'un client HTTP. Les navigateurs les plus populaires (en terme de part de marché) sont, Google Chrome, Mozilla Firefox, Internet Explorer, Safari et Opera.



Les navigateurs peuvent bloquer l'historique de navigation de l'utilisateur via des fichiers que l'on nomme cookies.

**Activité 1:** maîtriser les réglages les plus importants concernant la sécurité et la confidentialité d'un navigateur.

Choisir 2 des 5 navigateurs pour mener vos recherches.

Trouver le chemin pour supprimer les cookies du navigateur et détailler les différences dans la procédure.

Navigateur1 :

.....

Navigateur 2 :

.....

Différences :

.....

Quels intérêts présentent ces différences ?

.....

**Activité 2: Mesurer l'impact des cookies**

Comment découvrir la face cachée de sa propre navigation ? les experts de la commission nationale de l'informatique et des libertés (CNIL), dans le cadre d'un des premiers projets du laboratoire d'innovation, ont développé cookieviz. C'est un outil de visualisation pour mesurer l'impact des cookies et autres traqueurs lors de la navigation de l'internaute.

Une fois cookieviz installé, il suffit de visiter un site d'information, un réseau social ou une plateforme de e-commerce et cookieviz identifiera point par point les acteurs du web qui auront accès aux traces laissées.

1. Aller sur le site <https://github.com/LINCnil/CookieViz/releases>

Et télécharger le fichier CookieViz\_1.1\_setup.exe

Suivre le tutoriel en ligne.

Lancer cookieviz, puis naviguer sur 2 sites internet différents.

Retourner dans cookieviz et visualiser les graphiques.

2. Comparer les 2 visualisations et décrire les différences.

[www.education.gouv.fr](http://www.education.gouv.fr) et [www.amazon.fr](http://www.amazon.fr)

3. Indiquer quelles sortes d'informations peuvent être transmises aux acteurs avec liens cachés ?

a. si l'internaute est logué sur le compte

b. si l'internaute n'a pas ouvert de compte.

4. Quelles peuvent être les implications de l'usage des cookies par les sites ?

**Activité 3:** Gérer ses traces en ligne

Dès qu'un internaute saisit de l'information dans un site web –en se créant un profil ou en faisant une recherche- il donne des informations au site.

Ainsi, le site recueille ces données et les utilise pour personnaliser le contenu qu'il offre, plus particulièrement la publicité en fonction des intérêts de l'utilisateur.

Il faut donc avoir conscience du fait que les entreprises recueillent plusieurs types d'information sur les internautes et que cela peut avoir des avantages et des risques.

Faire des recherches en ligne pour répondre aux questions :

- Comment limiter ses traces en ligne ?

- Quelles sont les informations que Google est susceptible d amasser sur moi ?
- Quels peuvent être les avantages et les risques de la collecte de données par les sites web, les réseaux sociaux et les moteurs de recherche ?

## **V. Enjeux et débats**

Bordas p 62 à 65 : lire et répondre aux questions