

Cours Technologies Web et SIG

Dr SADOUNI Salheddine

Chapitre 1

Les notions de base du réseau Internet

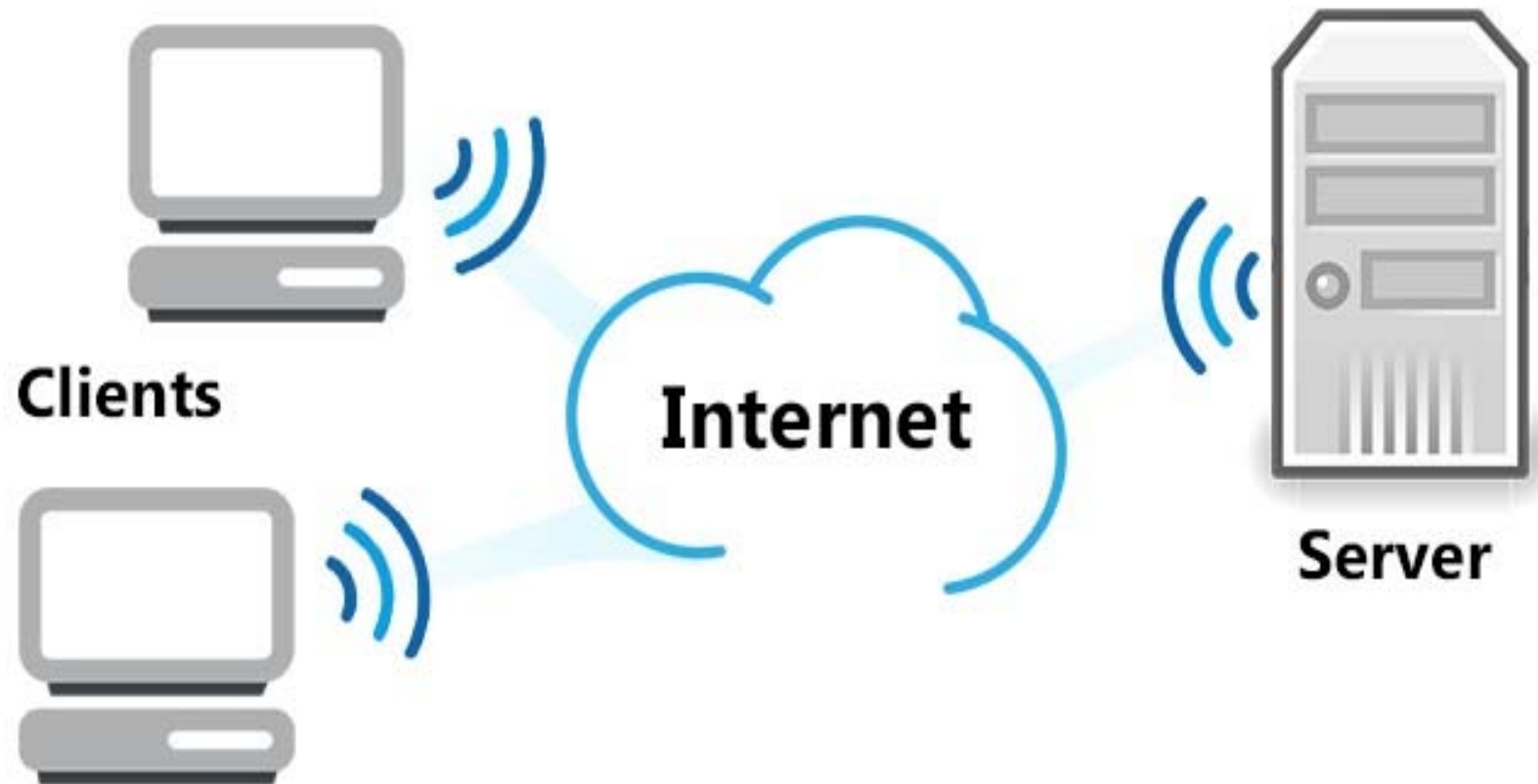
Objectifs de ce cours

- *Initiation aux **différents aspects d'internet** :*
- *URL, DNS, protocole http...etc*
- Connaître les principales **techniques pour la création de sites web**
- Se familiariser avec les langages du web
- **HTML, CSS**, JavaScript, **PHP, MySQL**
- Rendre **dynamique le contenu** des pages côté client ou côté serveur
- **modélisation de l'information géographique**

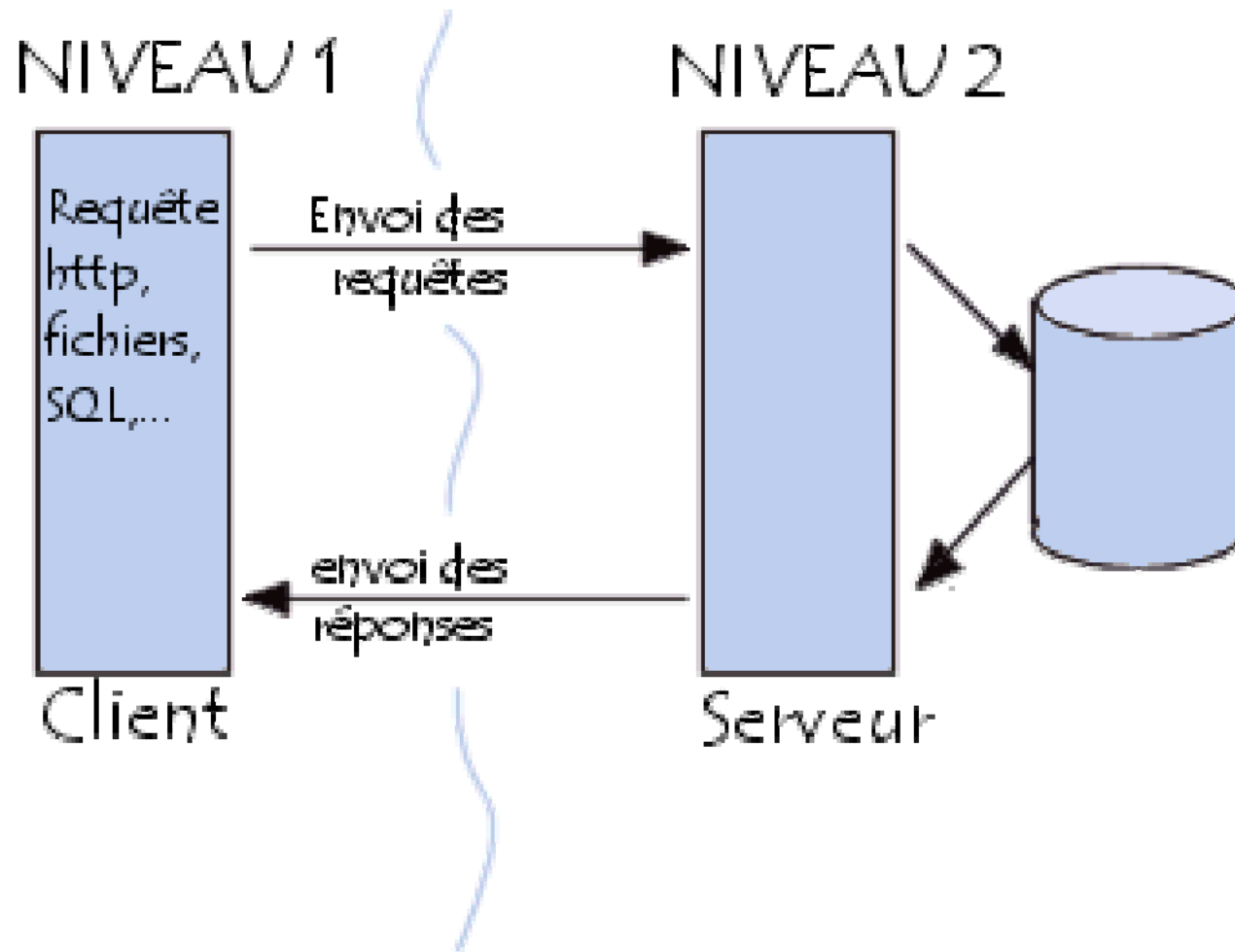
Internet

- Internet= *Interconnected Networks*
- Première utilisation remonte à **octobre1972**
- Réseau **informatique mondial** constitué d'un ensemble de réseaux nationaux, régionaux et privés(**Réseau de réseaux**)
- qui sont reliés par le protocole de communication **TCP/IP**
- et qui coopèrent dans le but d'offrir une **interface unique à leurs utilisateurs.**

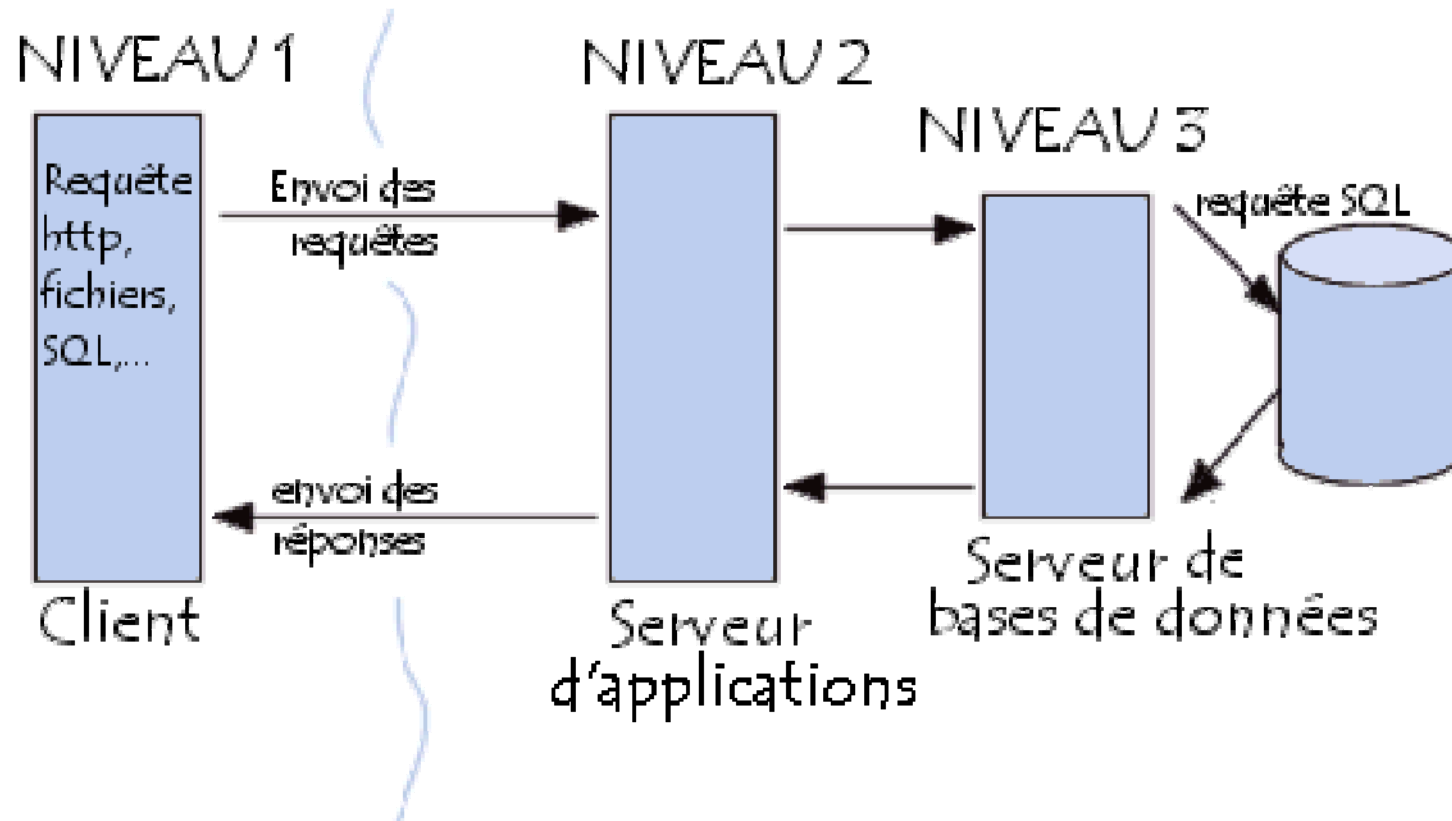
Mécanisme client-serveur par internet



Architecture Client/serveur à 2 niveaux

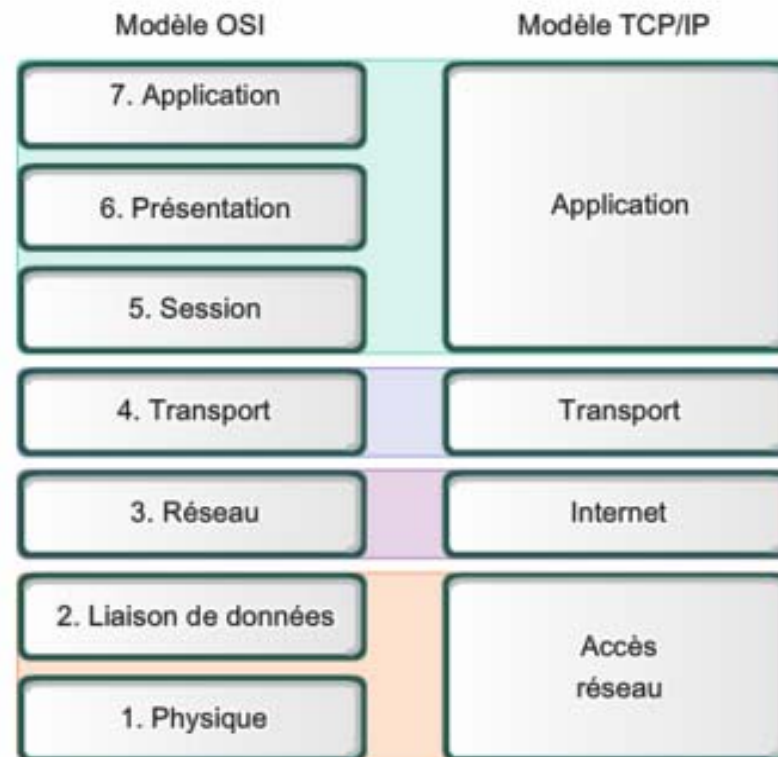


Architecture Client/serveur à 3 niveaux



Protocole TCP/IP

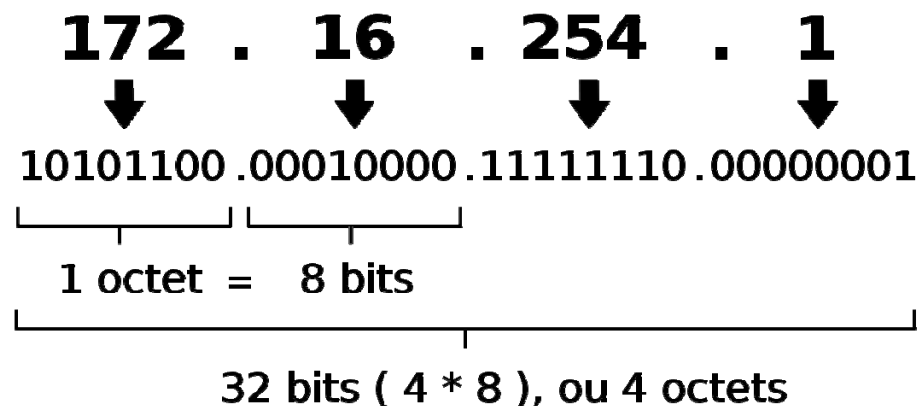
- Protocole TCP/IP (**Transmission Control Protocol/Internet Protocol**)
- TCP/IP représente **l'ensemble des règles de communication sur internet et se base sur la notion d'adressage IP.**



Adresse IP

- Sur Internet, les ordinateurs communiquent entre eux grâce au **protocole IP (Internet Protocol)**
- qui utilise des adresse numériques appelés **@IP. de 4 octets** séparés par des points.
- Un octet correspond à 8 bits de l'@ ce qui est fait une **adresse IP est constituée de 32 bits**.

Une adresse IPv4 (notation décimale à point)



DNS (Domain Name Server)

- il est pratiquement **impossible aux humains de connaître les adresses (IP)** des machines auxquelles ils veulent accéder.
- Afin de résoudre ce problème, le système des noms de domaine ou **DNS (Domain Name System)** a été introduit.
- Le DNS est le système d'adressage des serveurs mis au point pour **établir une relation entre l'adresse numérique d'un serveur (adresse IP) et une adresse logique.**
- Les **adresses IP des serveurs peuvent être converties en un nom de domaine et inversement.**
- Le nom de domaine est plus facilement lisible : **www.google.com** est le nom de domaine correspondant **216.58.208.238**
- **Le mécanisme consistant à trouver l'adresse IP correspondant au nom d'un hôte est appelé « résolution de nom de domaine ».**
- L'application permettant de réaliser cette opération est appelée « **résolveur** » (en anglais « **resolver** »).

Les applications internet

- L'utilisateur d'Internet, "l'internaute", a accès à de très **nombreux services** tels que :
 - ✓ **le courrier électronique,**
 - ✓ **transfert de fichiers,**
 - ✓ **groupes de discussions,**
 - ✓ **la recherche d'informations ...**, etc.
- La **recherche d'informations** ou bien le **Web** est **un moyen d'accès et de partage d'information** incontournable.
- Il utilise le protocole HTTP (**Hypertext Transfer Protocol**) **pour l'échange d'information entre le logiciel client et le serveur.**

L'*Hypertext Transfer Protocol* HTTP

- **protocole de transfert hypertexte** est un **protocole de communication client-serveur** développé pour le *World Wide Web*
- **HTTPS** (avec S pour *secured*, soit « **sécurisé** ») est la variante du HTTP *sécurisée* par l'usage des protocoles **SSL** ou **TLS**.
- **HTTP est un protocole de la couche application.**
- Il peut fonctionner sur n'importe quelle **connexion fiable**, dans les faits on utilise le **protocole TCP** comme couche de **transport**. Un serveur HTTP utilise alors par défaut **le port 80 (443 pour HTTPS)**.
- **Les clients HTTP les plus connus sont les navigateurs Web**
- permettant à un **utilisateur d'accéder à un serveur contenant les données.**
- Il existe aussi des systèmes pour **recupérer automatiquement le contenu d'un site** tel que les **aspirateurs de site Web** ou **les robots d'indexation.**

Le World Wide Web

- Le **World Wide Web** (WWW), communément appelé le **Web**, et est un **système hypertexte public fonctionnant sur Internet**.
- Le Web permet **de consulter, avec un navigateur, des pages accessibles sur des sites**.
- L'image de la toile d'araignée vient des hyperliens qui lient les pages web entre elles.
- **Le Web n'est qu'une des applications d'Internet**
- Le Web a été inventé par *Tim Berners-Lee* et *Robert Cailliau* plusieurs années après Internet,
- mais c'est lui qui a rendu les médias grand public attentifs à Internet. Depuis, le Web est fréquemment confondu avec Internet

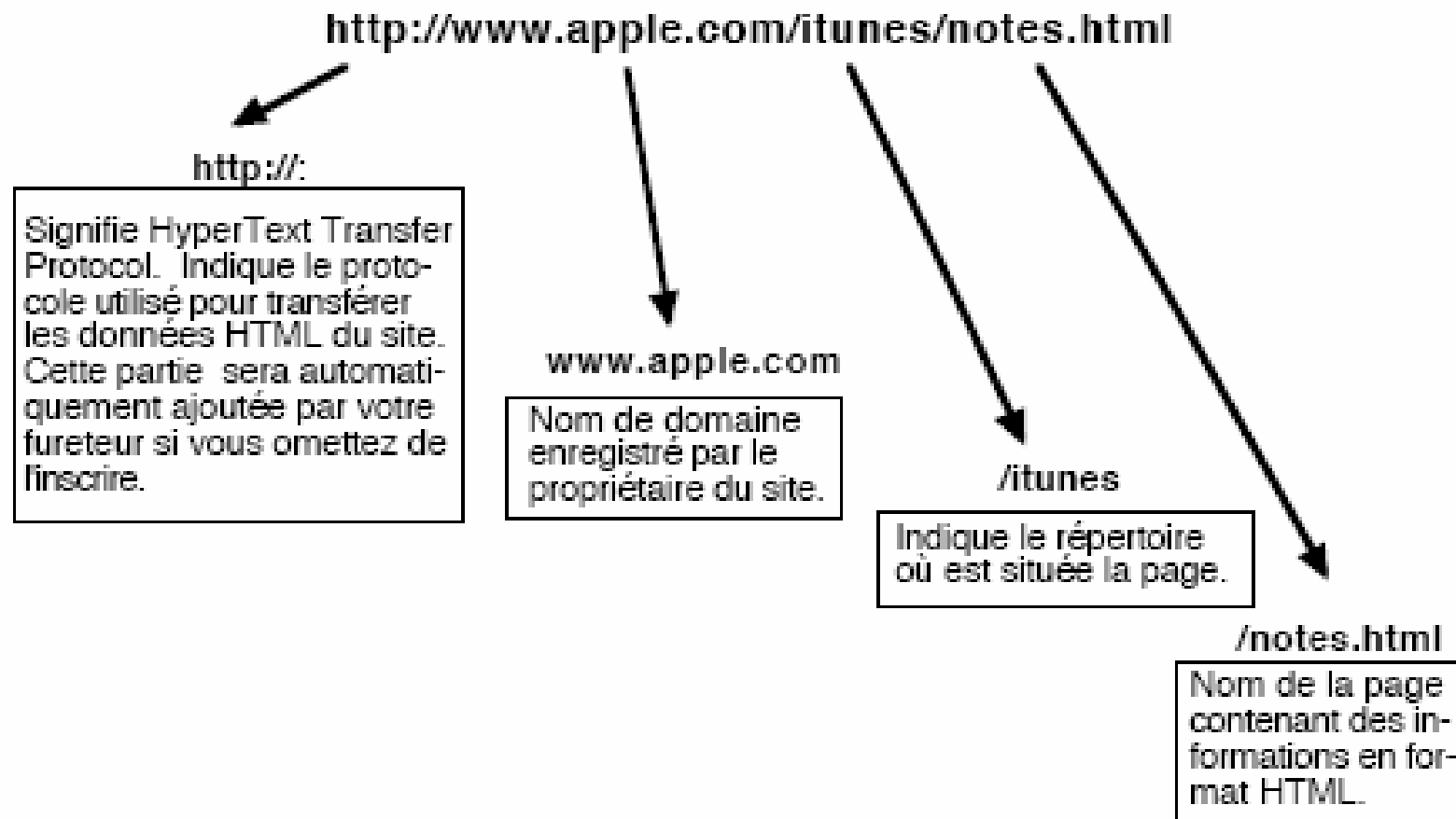


Fonctionnement du Web



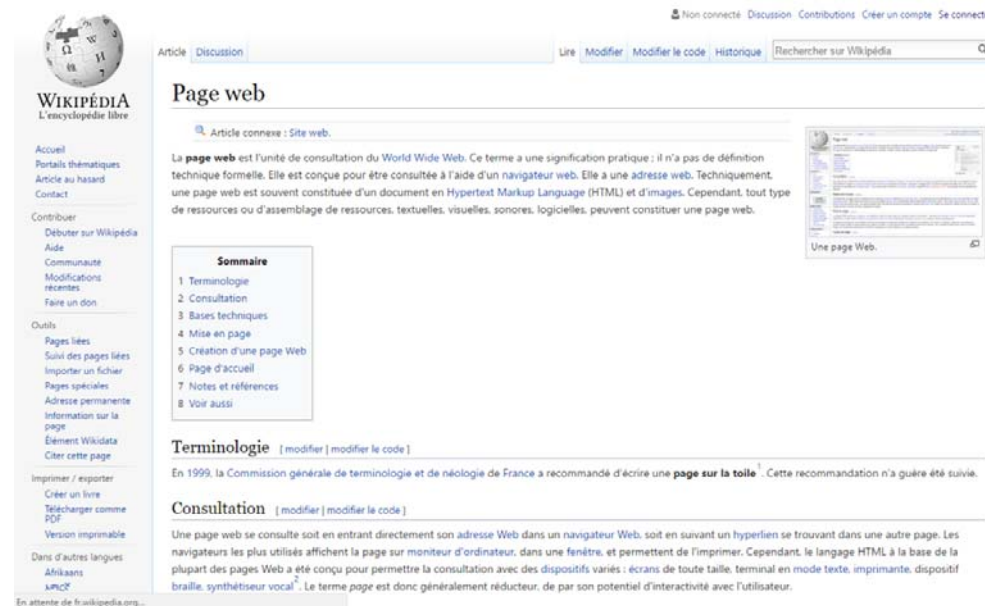
URL Uniform Resource Locator

Une adresse URL (Uniform Resource Locator) peut-être composée de plusieurs parties. En voici les principales.



Une Page web

- **Une ressource** du World Wide Web
- Créée par des webmasters à l'aide des **langages HTML/et CSS**
- Possède **une adresse Web**.
- Peut contenir du **texte**, des **images**, des **tableaux**, des **formulaires et autres éléments multimédias**
- **Visualisée** par les internautes **grâce à des navigateurs Web**



The screenshot shows the French Wikipedia page for "Page web". The page title is "Page web" and it is part of the "WIKIPÉDIA L'encyclopédie libre" project. The page content includes a definition of a "page web" as a unit of consultation on the World Wide Web, a table of contents with 8 sections (Terminologie, Consultation, Bases techniques, Mise en page, Création d'une page Web, Page d'accueil, Notes et références, Voir aussi), and sections for "Terminologie" and "Consultation". The "Terminologie" section mentions a 1999 recommendation from the French Commission générale de terminologie et de néologie. The "Consultation" section explains how a page web is accessed via a browser or hyperlinks. The page also features a sidebar with navigation links, a search bar, and a footer with the URL "En attente de fr.wikipedia.org...".

Site web

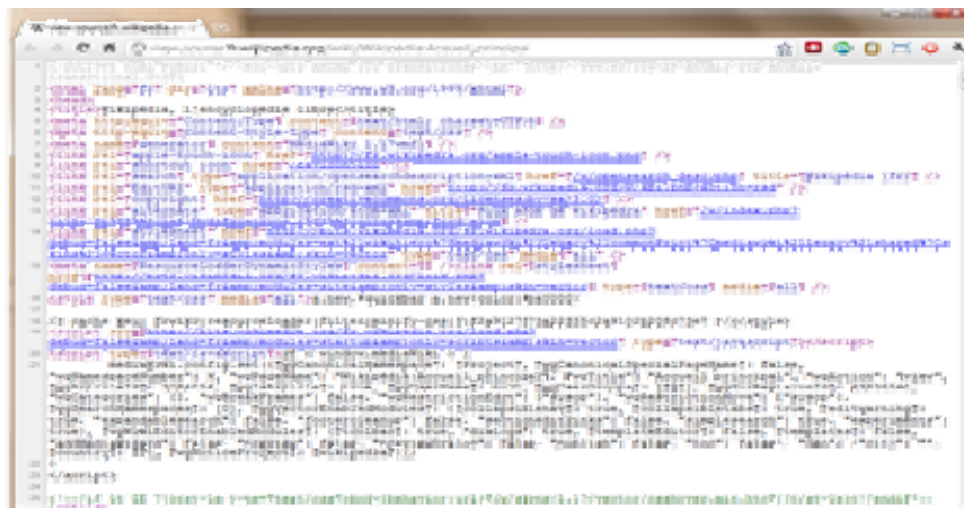
- **Un site web** (aussi appelé **site internet**) est un **ensemble de fichiers HTML stockés** sur un **ordinateur connecté en permanence à internet**
- Et hébergeant **les pages web (serveur web)**.
- Un site web est habituellement **architecturé autour d'une page centrale, appelée «page d'accueil» et proposant des liens vers un ensemble d'autres pages hébergées sur le même serveur,**
- et parfois des liens dits «**externes**», c'est-à-dire de pages **hébergées par un autre serveur.**



Le fonctionnement des sites 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 et CSS sont deux « langues de programmation » qu'il faut connaître 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.

Langages HTML et CSS



A screenshot of a web browser window displaying raw HTML and CSS code. The code is color-coded, with HTML tags in blue and CSS styles in red. The browser's address bar shows a URL starting with 'http://www.wikiquote.org/'. The code includes various HTML elements like <div>, <p>, , and , along with CSS rules for styling.

*Traduction par
l'ordinateur*



Résultat visible
à l'écran



HTML et CSS : deux langages pour créer un site web

- Pour créer un site web, on doit donner des instructions à l'ordinateur.
- Il ne suffit pas simplement de taper le texte qui devra figurer dans le site (comme on le ferait dans un traitement de texte Word, par exemple),
- il faut aussi indiquer
 - ✓ où placer ce texte,
 - ✓ insérer des images,
 - ✓ faire des liens entre les pages...etc.

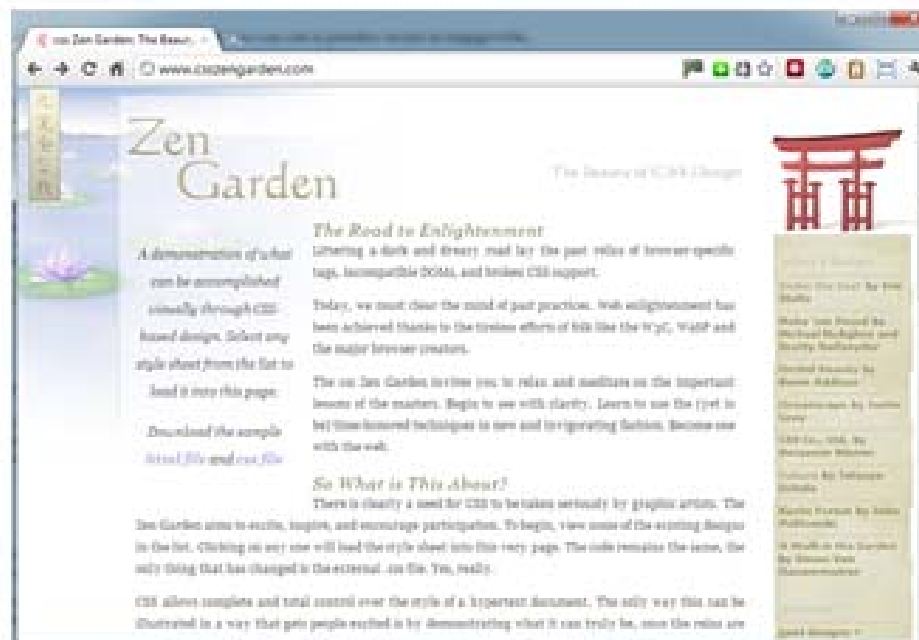
Le rôle de HTML et CSS

- **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...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 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)



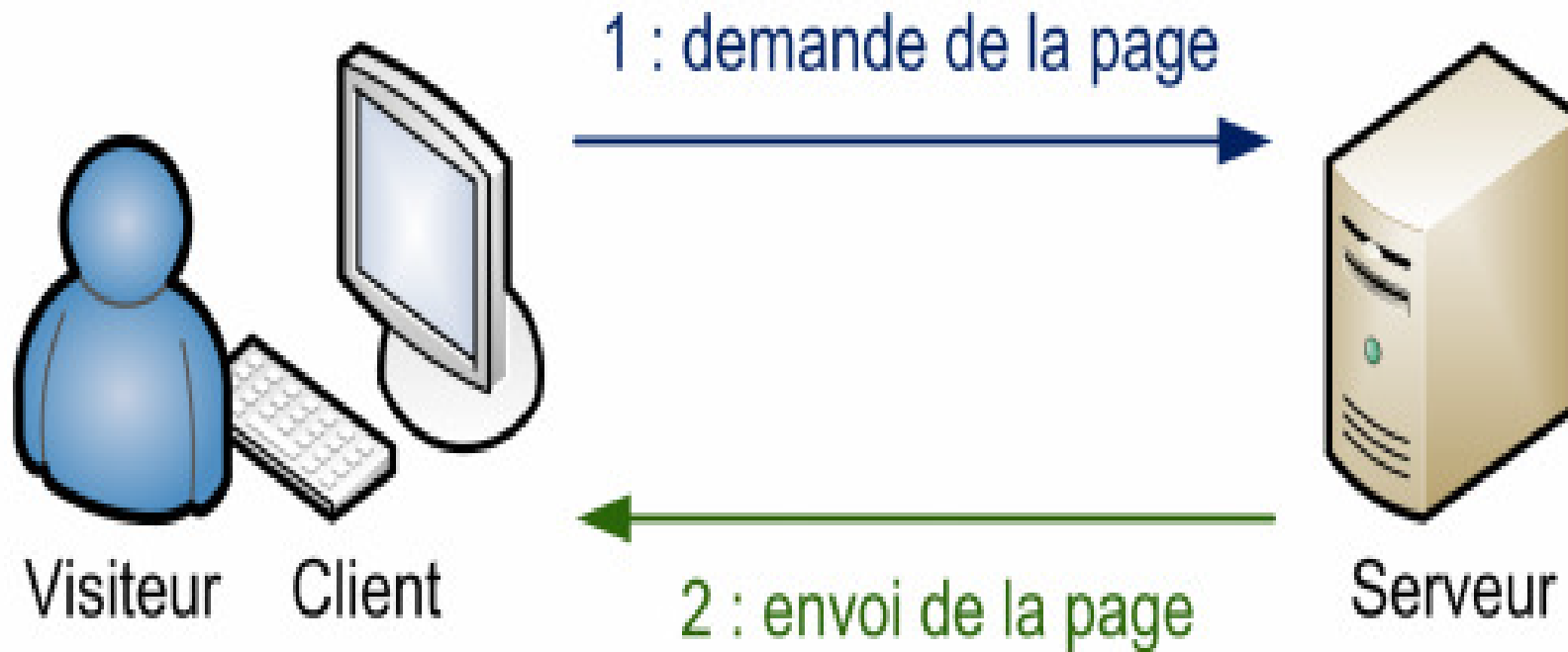
HTML + CSS



Le rôle de HTML et CSS

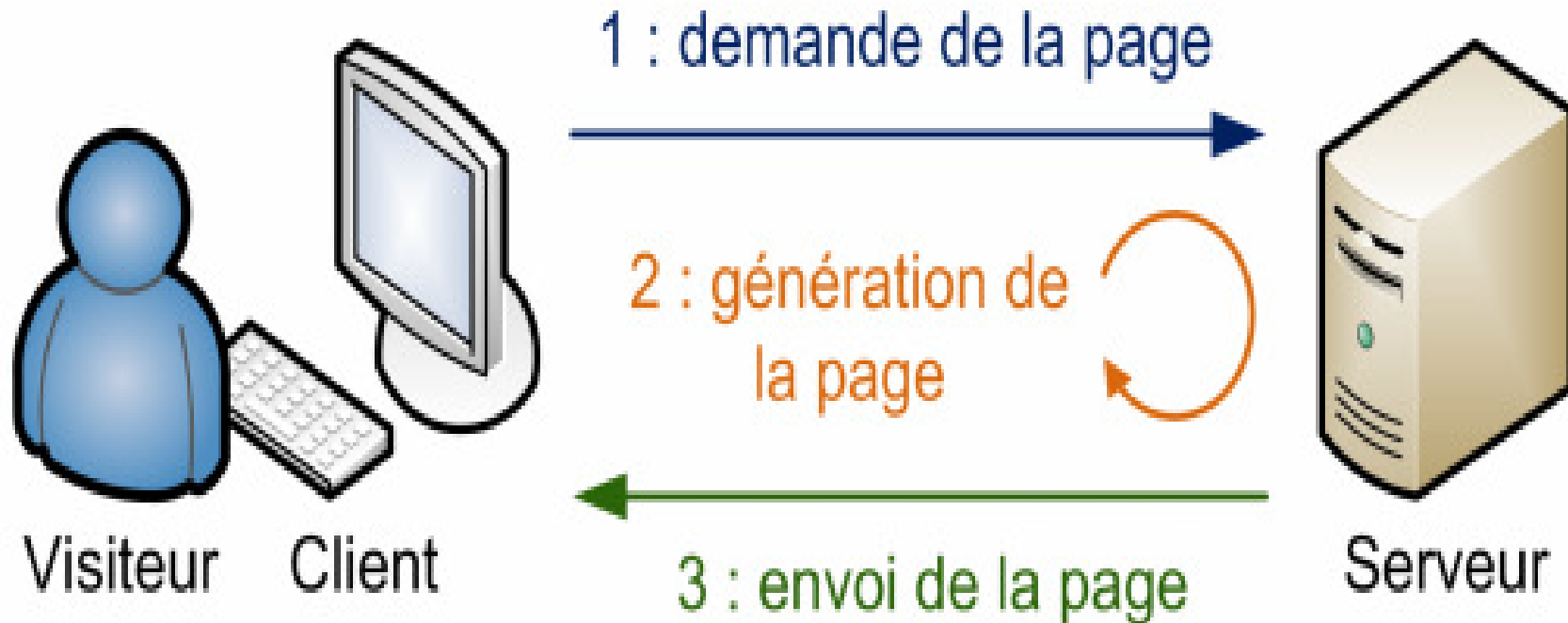
- Le **HTML** définit le contenu.
- Le **CSS** permet, lui, d'arranger le contenu et de définir la présentation : **couleurs, image de fond, marges, taille du texte...**
- le **CSS a besoin d'une page HTML pour fonctionner.**
- C'est pour cela que nous allons d'abord apprendre les bases du HTML avant de nous occuper du CSS.
- **Historiquement parlant**, nous avons **05 versions du langage HTML et 03 versions du langage CSS**
- Lors de ce cours nous travaillerons avec les dernières version de chaque langage **HTML 5 et CSS 3**

Cas d'une Page web statique



Transferts avec un site statique

Cas d'une Page dynamique



Transfert avec un site dynamique

Logiciel de création des sites web

WYSIWYG

- On peut classer les **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 **joomla** !
- 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.

Logiciel de création des sites web

éditeurs de texte Sublime Text

- **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 !
- **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.

Chapitre 2

Le langage HTML

Le langage HTML

- HTML est l'abréviation de **HyperText Markup Language**, soit en français « **langage hypertexte de balisage** ».
- Ce langage a été **créé en 1991**.
- Permet **la structuration des pages WEB**
- Permet aussi de faire le **design des page WEB (mise en forme du contenu)**
- mais on préfère pour **faciliter la mise à jour des formes des page web, utiliser le CSS**.

Les balises et leurs attributs

- En fait, pour créer une page web il ne suffit pas de taper simplement du texte.
- 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.

Les balises

- **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>**

Les balises

- 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 ».

<image />

- Notez que le **/ de fin n'est pas obligatoire**.
- **On pourrait écrire seulement<image>**.
- **pour ne pas les confondre avec le premier type de balise, les webmasters recommandent de rajouter ce/(slash) à la fin des balises orphelines.**

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">**
- la balise<image /> 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**.

Attributs pour les balises paires

- 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>
```

Structure de base d'une page HTML5

- 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**.

L'ordre des balises

- 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.

Le doctype et La balise</html>

- `<!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).
- 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.

deux balises contenues dans l'en-tête

- L'encodage (charset)

<meta charset="utf-8" />

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

- **<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)
- Il faut savoir que le titre apparaît aussi dans les résultats de recherche, comme sur Google

Les 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.
- Vous pouvez utiliser les commentaires pour laisser des indications sur le fonctionnement de votre page.
- **Insérer un commentaire**
- Un commentaire est une balise HTML avec une forme bien spéciale :

<!-- Ceci est un commentaire -->

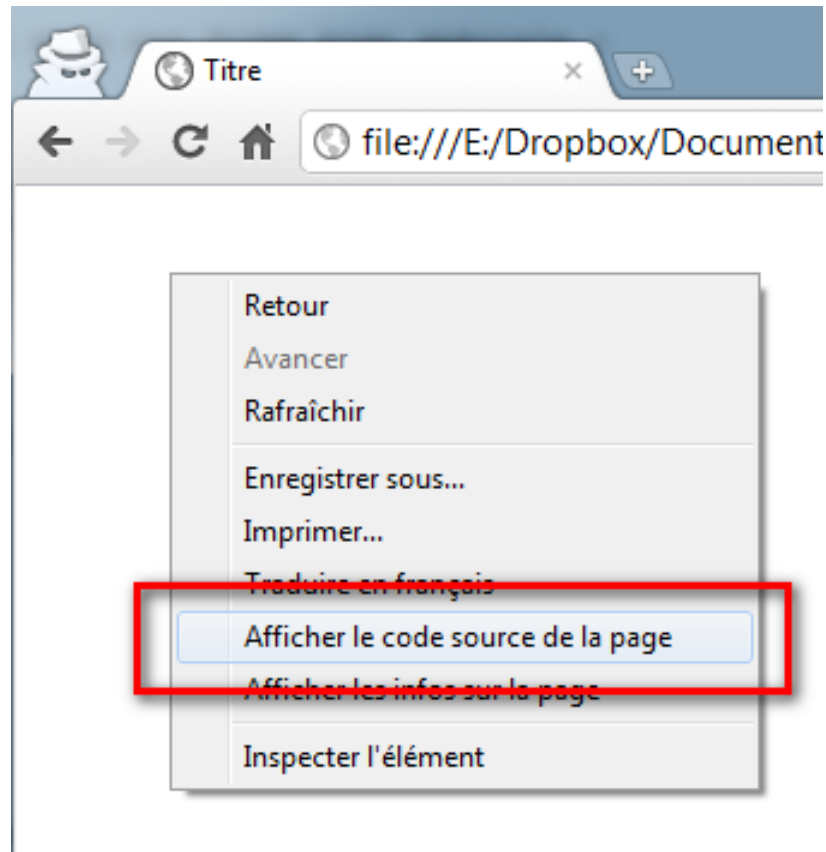
Exemple d'un programme HTML

```
<!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 votre code HTML

- 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.
- le navigateur *doit* obtenir le code HTML pour savoir ce qu'il faut afficher. **Le code HTML de tous les sites est donc public.**
- Par conséquent, ne mettez pas d'informations **sensibles comme des mots de passe dans votre code HTML et soignez votre code source**

Afficher le code source de la page



```
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>
```

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 !

`<!DOCTYPE html>`

`<html>`

`<head>`

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

`<title>Paragraphes</title>`

`</head>`

`<body>`

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

`</body>`

`</html>`

le saut de 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>
```

Plusieurs paragraphes

- Tout le texte s'affiche sur la même ligne
- 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 :

Exemple de plusieurs paragraphes

```
<!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 de notre programme



La balise Aller à la ligne

- C'est **une balise orpheline** qui sert juste à indiquer qu'on doit aller à la ligne :
- **
**
- **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>
```

L'utilisation de plusieurs balises

- Vous pouvez théoriquement mettre **plusieurs balises
** 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
- **<p> </p>: pour organiser son texte en paragraphes ;**
- **
: pour aller à la ligne.**

Les titres

- HTML utiliser six niveaux de titres différents, par ordre d'importance 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>**
- **<h4> </h4>**
- **<h5> </h5>**
- **<h6> </h6>**
- 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> à <h6> servent à créer des titres qui seront affichés dans la page web.

Exemple d'utilisation des titres

```
<!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>
```

La mise en valeur de quelques mots

- Au sein de vos paragraphes, **certains mots sont parfois plus importants** que d'autres .
- 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 **`` ``**.
- la balise **``** a ***pour conséquence de mettre le texte en italique***.
Son utilisation est très simple : encadrez les mots à mettre en valeur avec ces balises

Exemple de l'utilisation de la balise

``

```
<!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>
```

la balise

```
<!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>
```

- on utilise la **balise** qui signifie « fort », ou « important.
- Elle s'utilise exactement de la **même manière que**

L'affichage des mots importants

- utiliser la balise `` 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.
- Avec la balise `` 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 `` 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.

La balise <mark>

```
<!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.)
- **HTML pour le fond** (contenu, logique des éléments) ;
- **CSS pour la forme** il définit la forme (apparence).

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 **baliseque l'on referme un peu plus loin avec**.
Commencez donc à taper ceci :
- ****

Exemple d'une liste non ordonnée

- on va écrire chacun des éléments de la liste entre deux balises ``.
- **Chacune de ces balises doit se trouver entre `` et ``.**
exemple :

``

`Fraises`

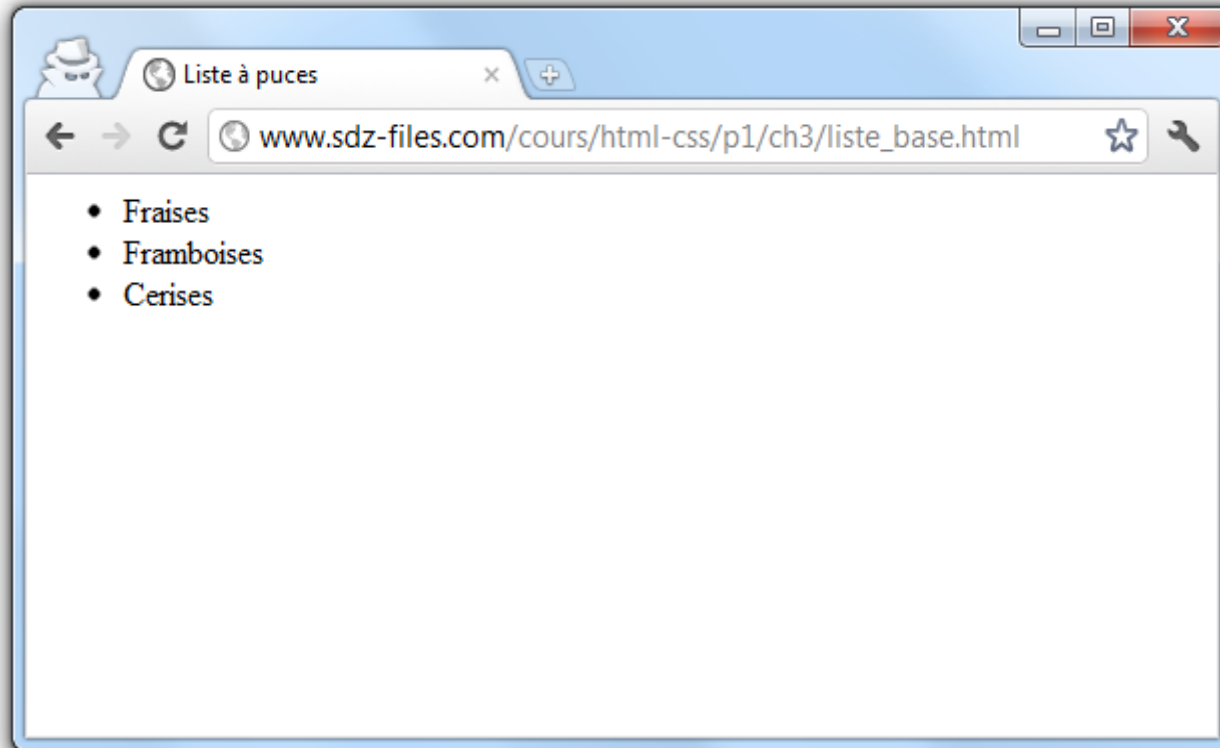
`Framboises`

`Cerises`

``

- Le résultat se trouve à la figure suivante.
- **`` délimite toute la liste ;**
- **`` délimite un élément de la liste (une puce).**

Le résultat de la liste non ordonnée



- **Pour les listes complexes**, 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 à l'intérieur d'un élément *.

Liste ordonnée

- Une liste ordonnée fonctionne de la même façon, seule une balise change : ***il faut remplacer***````***par***````.
- **À l'intérieur de la liste**, on ne change rien : on utilise toujours des ***balises***```` pour délimiter les éléments.
- L'ordre dans lequel vous placez les éléments de la liste est **important**. Le premier```` sera l'élément n° 1, le second sera le n°2 etc...

<h1>Ma journée</h1>

Je me lève.

Je mange et je bois.

Je retourne me coucher.

Le résultat de la liste ordonnée



- **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***).

Les liens

- un site web est composé de plusieurs pages. **Comment faire pour aller d'une page vers une autre ?**
- **À l'aide de liens**, Dans cette partie , nous allons justement apprendre à créer des liens entre nos pages.
- un **lien est 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.google.dz>).
- Dans les deux cas, nous allons voir que le **fonctionnement est le même.**

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 www.google.dz
- Pour faire un lien, la balise que nous allons utiliser est **<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 google.dz, situé à l'adresse <https://google.dz>:

google

placer ce lien au sein d'un paragraphe

- 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://google.dz">google</a> ?<br />  
C'est un bon moteur de recherche ;o)</p>
```

- Par défaut, le lien s'affiche en bleu souligné.
- Si vous avez déjà ouvert la page, le lien s'affiche en violet.
- 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 par **https://(sites sécurisés**
- ou d'autres **préfixes (ftp://,...)**.
- Les liens que nous venons de voir sont appelés **liens absolus car on indique l'adresse complète**.

Un lien vers une autre page de son site

- il est nécessaire aussi **de faire des liens entre les différentes pages d'un même site**
- 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.
- **Deux pages situées dans un même dossier**
- Pour commencer, nous allons créer deux fichiers correspondant à deux pages HTML différentes, **page1.html** et **page2.html**.
- Nous aurons donc ces deux fichiers sur notre disque *dans le même dossier*
- **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 : **``**.
- On dit que c'est un **lien relatif**.

Deux pages HTML dans le même dossier

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

- **page1.html**

<p>Bonjour. Souhaitez-vous consulter la page 2 ?</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>

Deux pages situées dans des dossiers différents

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

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



contenu

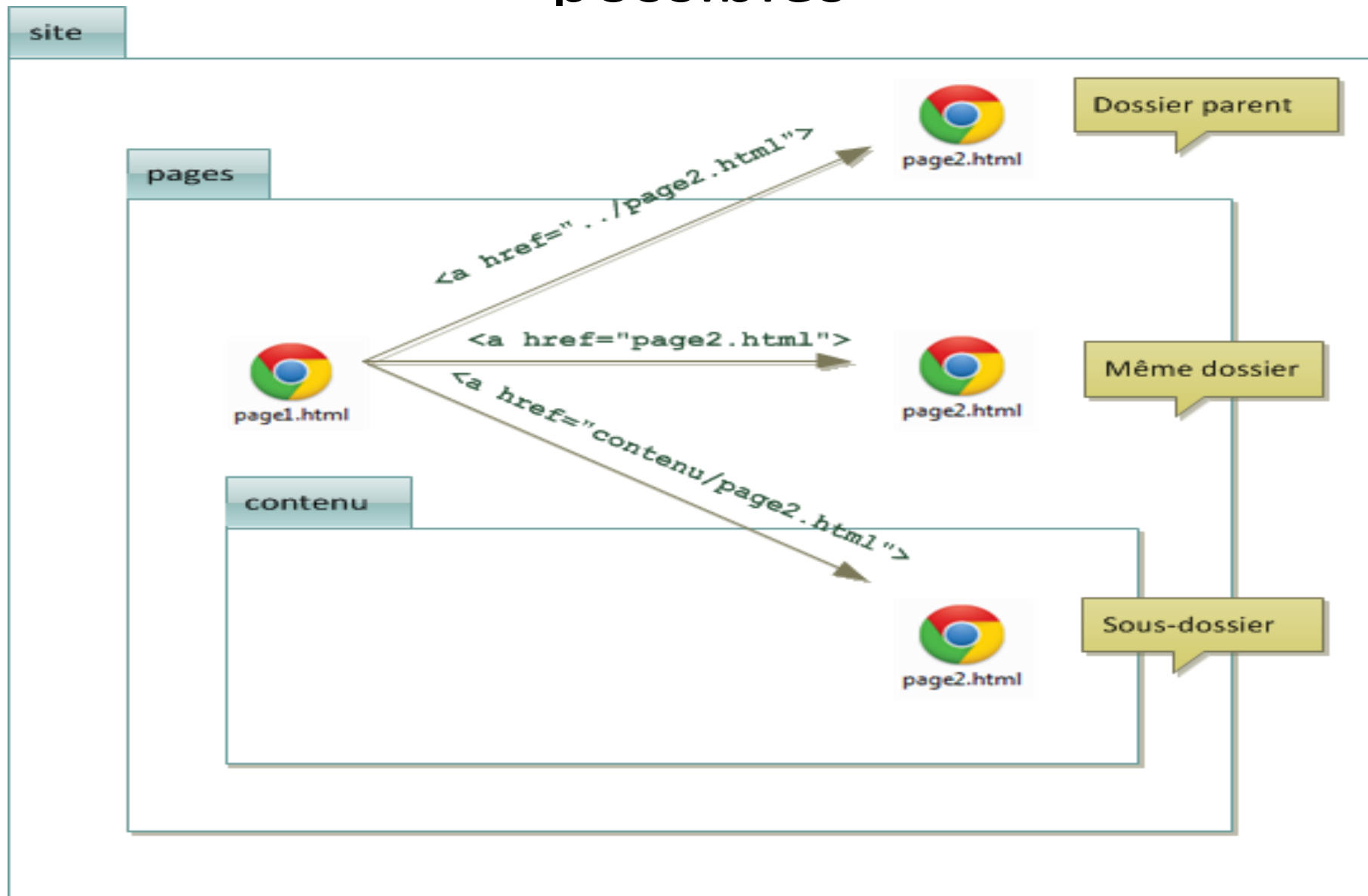


page1.html

Si le fichier se trouve dans un dossier parent

- Si votre **fichier cible est placé dans un dossier qui se trouve « plus haut » dans l'arborescence**, il faut écrire **deux points** comme ceci :
- **``**
- Les liens relatifs ne sont pas bien compliqués, 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.

synthèse des différents liens relatifs possibles



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 :

Aller vers l'ancre

- 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**).

Exemple

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

`<p>`

`Aller directement à la partie traitant de :
`

`La cuisine
`

`Les rollers
`

`Le tir à l'arc
`

`</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>`

L'attribut id

- **L'attribut id** sert à donner un **nom « unique »** à une balise, pour s'en servir de repère.
- **en CSS**, il nous sera très utile pour « repérer » une balise précise.
- **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.**

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

Exemple

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

<h1>Le exemple ancre sur une page différente </h1>

<p>

*Rendez-vous quelque part sur une autre page :
*

*La cuisine
*

*Les rollers
*

*Le tir à l'arc
*

</p>

Cas pratiques d'utilisation des liens

- liens qui lancent un **téléchargement**
- Qui créent **un nouvel e-mail**
- Qui **ouvrent une nouvelle fenêtre**

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

<p>Bonjour. Souhaitez-vous visiter google ?</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 google ?</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.

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:salah.sadouni@gmail.com">Envoyez-moi un e-mail !</a></p>
```

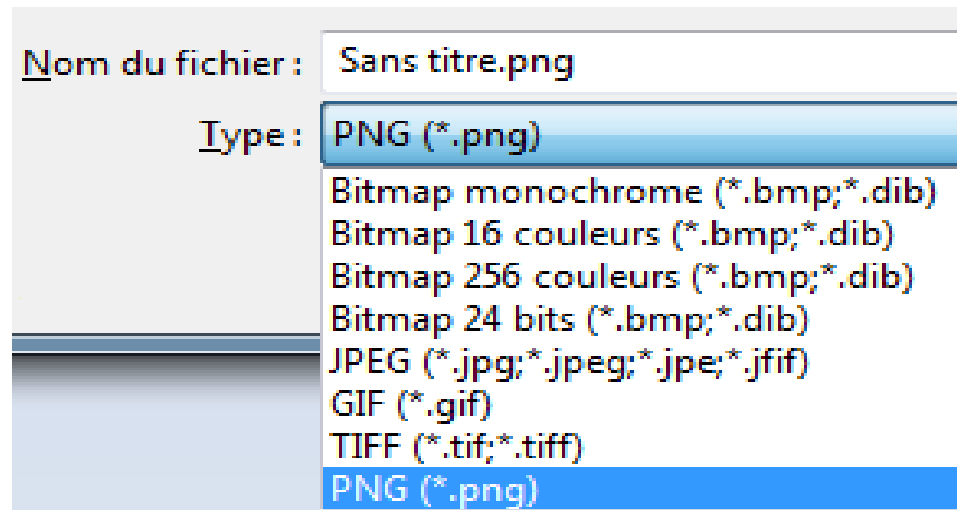
- Il suffit donc de faire commencer le lien par **mailto:** et d'écrire **l'adresse e-mail où on peut vous contacter**.
- Si vous cliquez sur le lien, un nouveau message vide s'ouvre, prêt à être envoyé à votre adresse e-mail.

Un lien pour télécharger un fichier

- 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>Télécharger le fichier</p>***
- **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**.

Les images et leurs différents formats

- Quand vous avez une image, 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 vous propose de choisir entre plusieurs formats lorsque vous enregistrez une image



Le choix des formats d'images

- Certains formats sont plus adaptés que d'autres selon l'image (photo, dessin, image animée...).
- Notre but dans cette partie 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**.
- 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 format 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**, qui peuvent comporter **plus de 16 millions de couleurs différentes**.
- 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**.

Photo JPEG



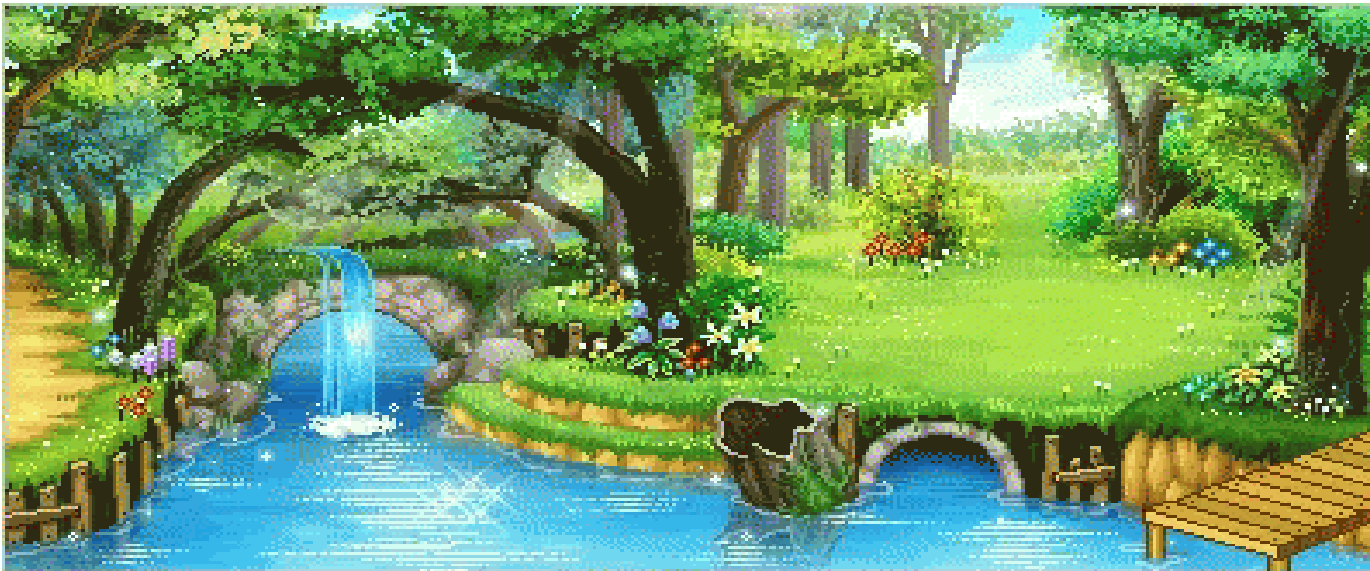
Le format PNG

- Le format **PNG** (*Portable Network Graphics*) est le **plus récent de tous**.
- Ce format est **adapté à la plupart des graphiques** (généralement « **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**
- 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**
- La figure suivante est une image PNG en 24 bits



Le Format GIF

- C'est un format assez vieux, qui a été néanmoins très utilisé (et qui reste très utilisé par habitude). **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**
- 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 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.
- **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* (« _ »)

Insérer une image

- Pour l'Insertion d'une image on utilise la balise `` type orpheline
- 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,
- ou dans les navigateurs de **personnes handicapées** (non-voyants) .
- 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>).**

exemple d'insertion d'image :

`<p>`

*Voici une photo que j'ai prise lors de mes dernières vacances à la montagne :
*

**

`</p>`

- Bref, l'insertion d'image est quelque chose de très facile pour peu qu'on sache indiquer où se trouve l'image

La plus grosse difficulté consiste à **choisir le bon format d'image**.

- Il faut **évit**ez 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".

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.

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 :
*

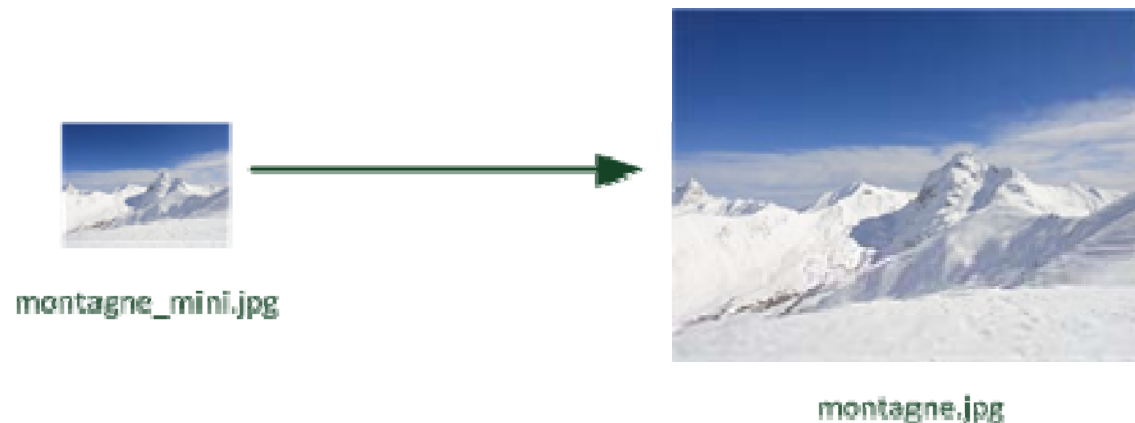
**

</p>

Survolez la photo avec la souris pour voir l'infobulle apparaître

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://www.resizeimage.net)
- Je vais ainsi disposer de deux versions de ma photo, comme à la figure suivante : la miniature et l'image d'origine.



Affichage de la photo d'origine et de la miniature

- Je les 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 !

</p>

- Parfois, certains navigateurs choisissent d'afficher un cadre bleu (ou violet) pas très esthétique autour de votre image cliquable.
- nous pourrons retirer ce cadre dans peu de temps grâce au CSS.

Les figures

- **Les figures** sont des éléments qui viennent **enrichir le texte** pour compléter les informations de chaque page.
- Les figures peuvent être de différents types :
- **images ;**
- **codes source ;**
- **citations ;**
- **etc.**
- les figures **ne sont pas forcément des images** : un code source aussi illustre le texte.

Création d'une figure

la balise <figure>

- 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>

le rôle des figures

- 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.
- **Mais 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>

- **Si l'image 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 sémantique

- La balise *<figure>* a un rôle avant tout sémantique.
- 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.
- 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>

Validation et compatibilité entre navigateurs

- À la différence de langages de programmation, un non-respect de la syntaxe de HTML **n'empêchera pas l'affichage sur le navigateur.**
- Plus généralement, le navigateur affichera **quelque chose (de non spécifié) ou n'affichera rien**

Validation

Il faut impérativement **valider** son code HTML avant de le publier.

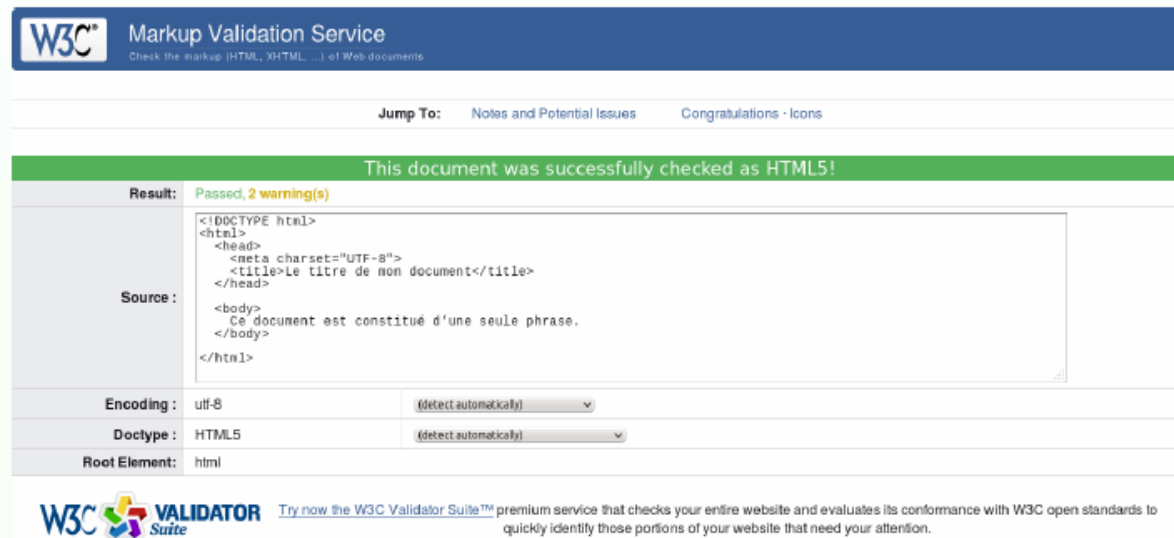
Validateur du W3C

Validateur du W3C

Un outil indispensable

- le **validateur** du W3C et de la fondation Mozilla

<http://validator.w3.org/>



The screenshot displays the W3C Markup Validation Service interface. At the top, the W3C logo and the text "Markup Validation Service" are visible, along with the tagline "Check the markup (HTML, XHTML, ...) of Web documents". Below this, there are navigation links for "Jump To: Notes and Potential Issues" and "Congratulations · Icons". A prominent green banner states "This document was successfully checked as HTML5!". The main content area shows the validation result: "Result: Passed, 2 warning(s)". The "Source" section displays the HTML code of the document being validated, which includes a DOCTYPE declaration, a head section with a meta charset and a title, and a body section with a single line of text. Below the source code, there are dropdown menus for "Encoding" (set to utf-8), "Doctype" (set to HTML5), and "Root Element" (set to html). At the bottom, there is a promotional banner for the "W3C Validator Suite" premium service.

W3C Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

Jump To: [Notes and Potential Issues](#) [Congratulations · Icons](#)

This document was successfully checked as HTML5!

Result:	Passed, 2 warning(s)
Source:	<pre><!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <title>Le titre de mon document</title> </head> <body> Ce document est constitué d'une seule phrase. </body> </html></pre>
Encoding:	utf-8 (detect automatically)
Doctype:	HTML5 (detect automatically)
Root Element:	html

W3C VALIDATOR Suite Try now the W3C Validator Suite™ premium service that checks your entire website and evaluates its conformance with W3C open standards to quickly identify those portions of your website that need your attention.

Chapitre 3

Le langage CSS

CSS (*Cascading Style Sheets*)

- C'est un autre langage qui vient compléter le HTML dans le but de *gérer la mise en forme de votre site*.
- **Petit rappel : à quoi sert CSS ?**
- CSS vous permet de **choisir la couleur de votre texte**.
- Il permet de sélectionner **la police utilisée sur votre site**.
Il permet de définir **la taille du texte, les bordures, le fond...**
- Il permet de faire **la mise en page de votre site**.

Le lancement du CSS

- 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, la mise en forme se faisait uniquement en HTML
- Il y avait en effet des **balises HTML dédiées à la mise en forme**.
- ****
- 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 cela 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.

Emplacement du code 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.**

Méthode 1 fichier.css (recommandé)

- 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 :

Exemple de programme HTML

```
<!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>
```


Application d'un code CSS sur un programme HTML

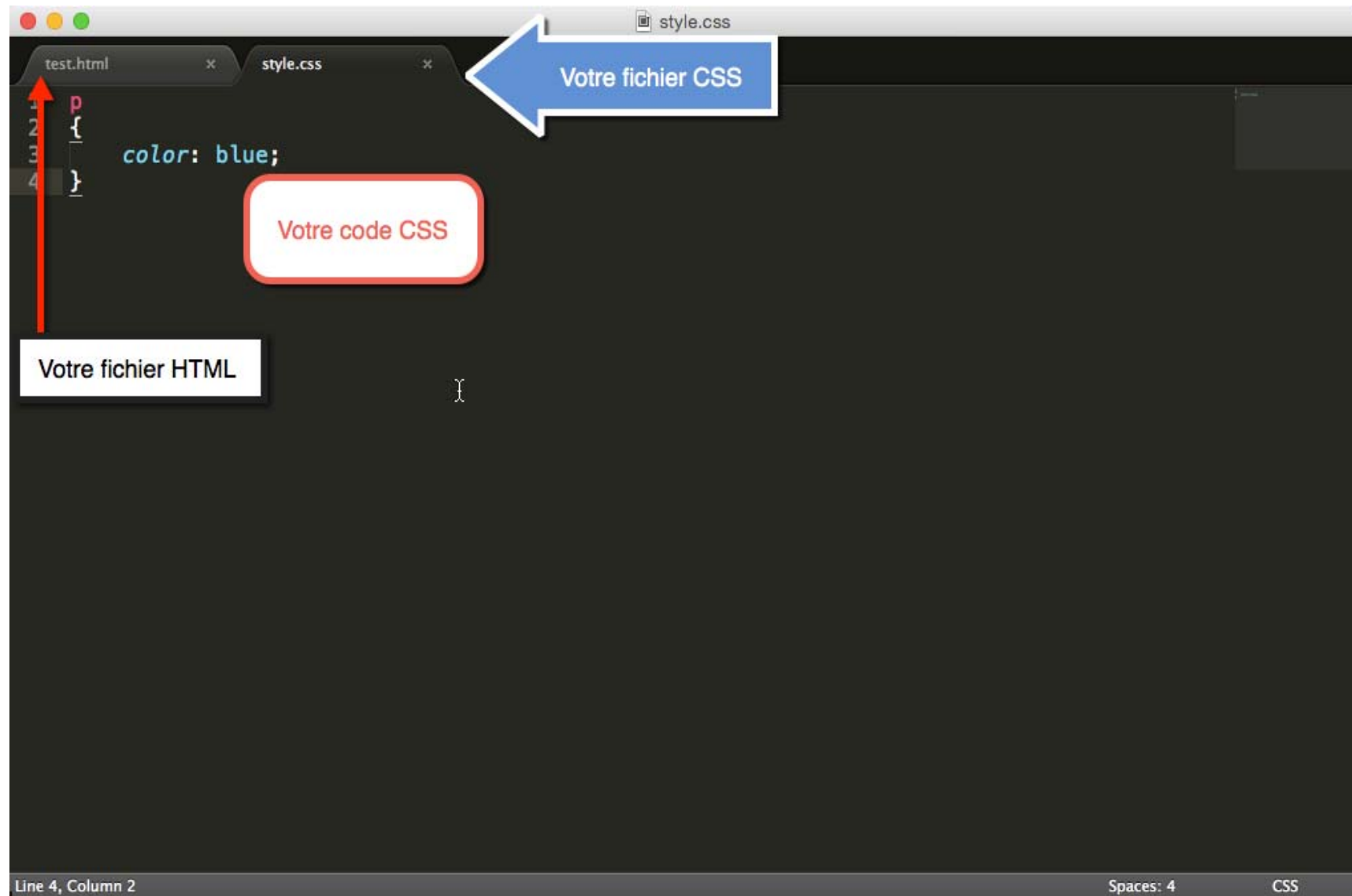
- Vous noterez le contenu de la **ligne 5**,
- **`<link rel="stylesheet" href="style.css" />`**
- c'est elle qui indique que ce **fichier HTML est associé à un fichier appelé style.css et chargé de la mise en forme.**
- Enregistrez ce fichier sous le nom que vous voulez (par exemple page.html).
- Pour 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 et copiez ce bout de code CSS**

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

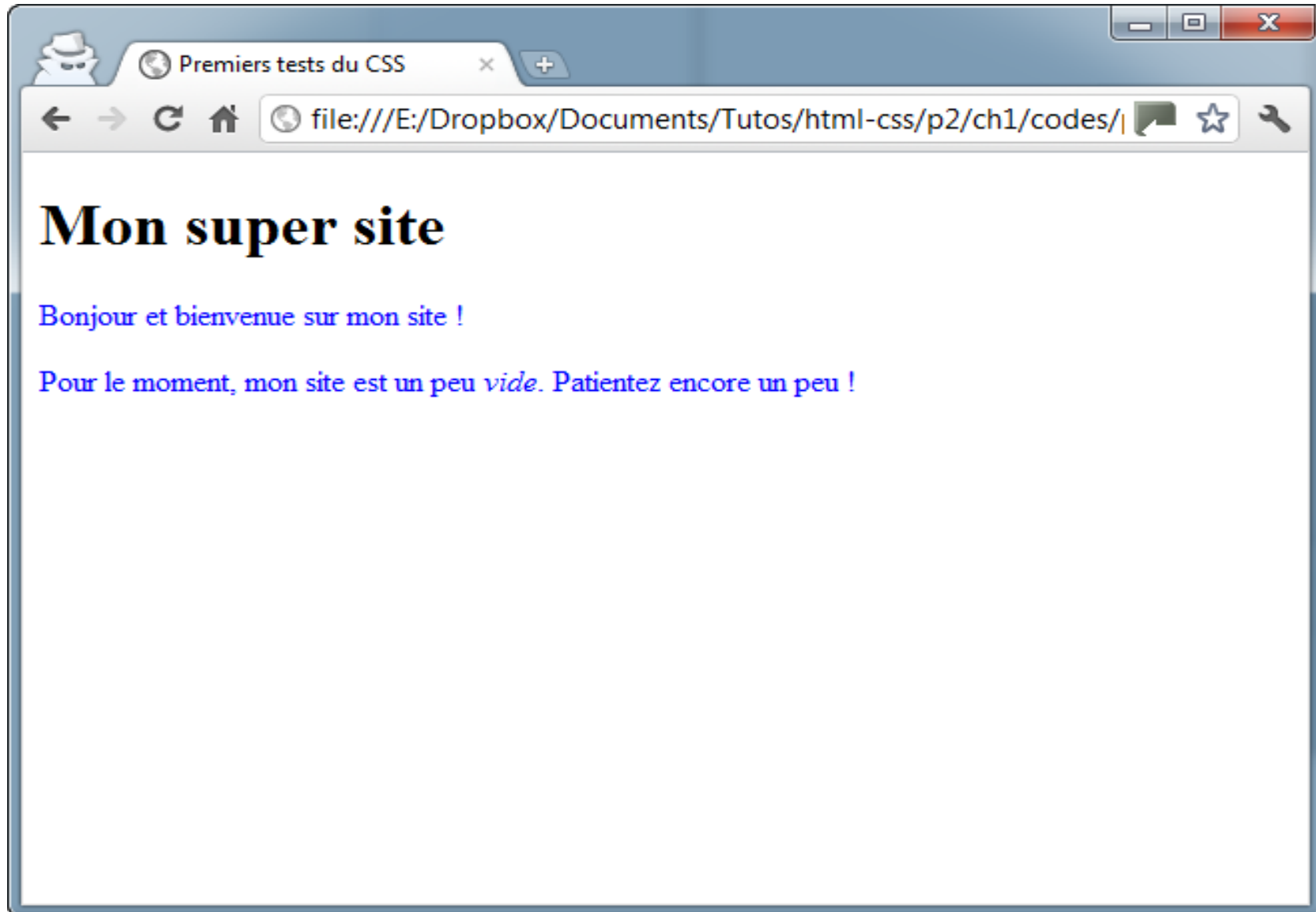
Coloration de votre texte méthode 1

- Pour obtenir la coloration du code dans Sublime Text, enregistrez bien votre fichier avec l'extension.css .
- 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 votre explorateur de fichiers, vous devriez les voir apparaître côte à côte. D'un côté le.html, de l'autre le.css.
- le fichierpage.html fera automatiquement appel au fichierstyle.css

Le résultat sur votre éditeur de text sublime text



Le résultat sur le navigateur Web



Méthode 2 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) :**
- Testez, vous verrez que le résultat est le même

Fichier HTML contenant le code CSS dans l'en-tête

```
<!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>
```

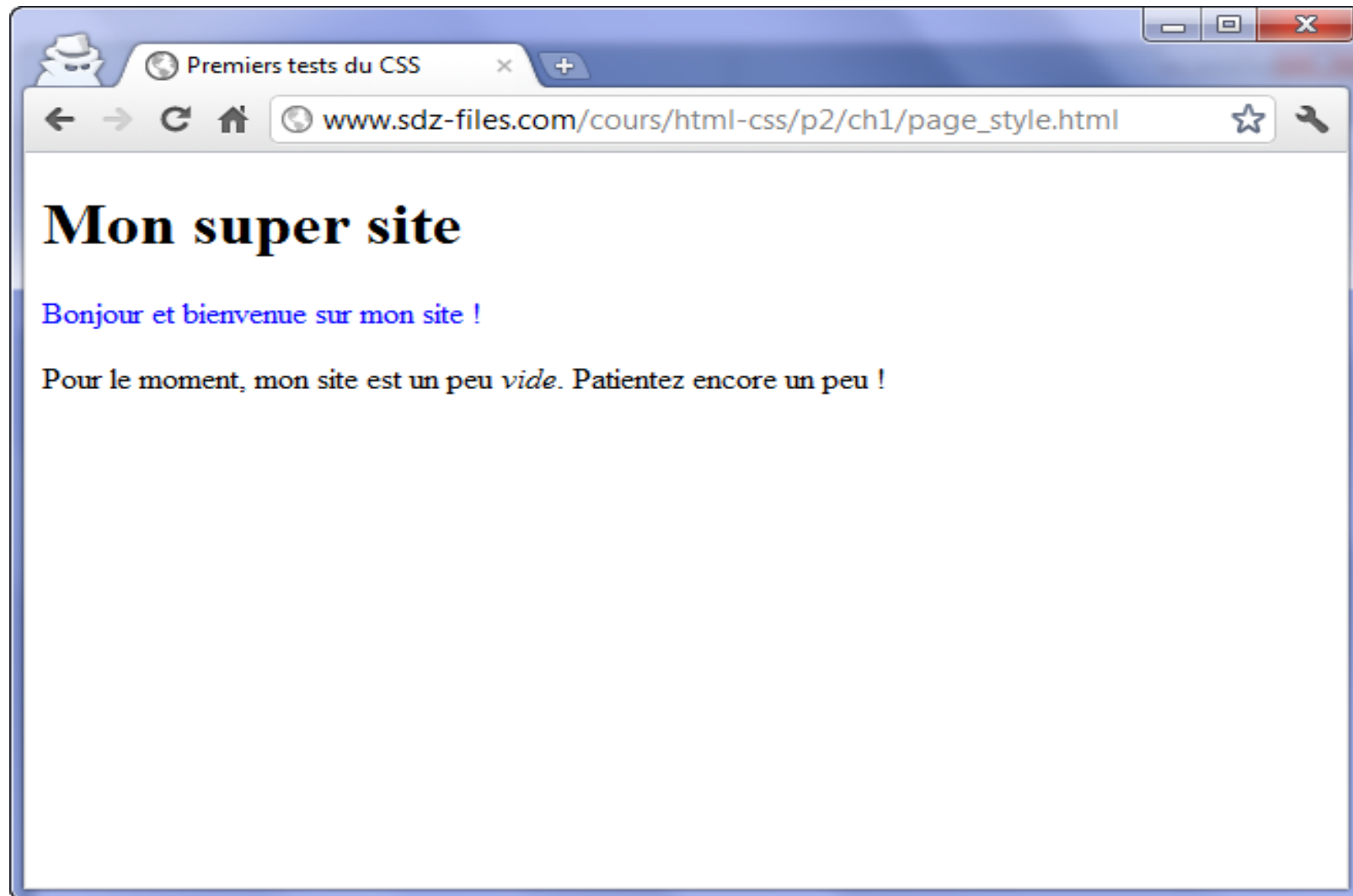
Méthode 3 Directement dans les balises (non recommandé)

- 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 :**
- Cette fois, seul le texte du premier paragraphe (ligne 11), dont la balise contient le code CSS, sera coloré en bleu

Méthode 3 Directement dans les balises

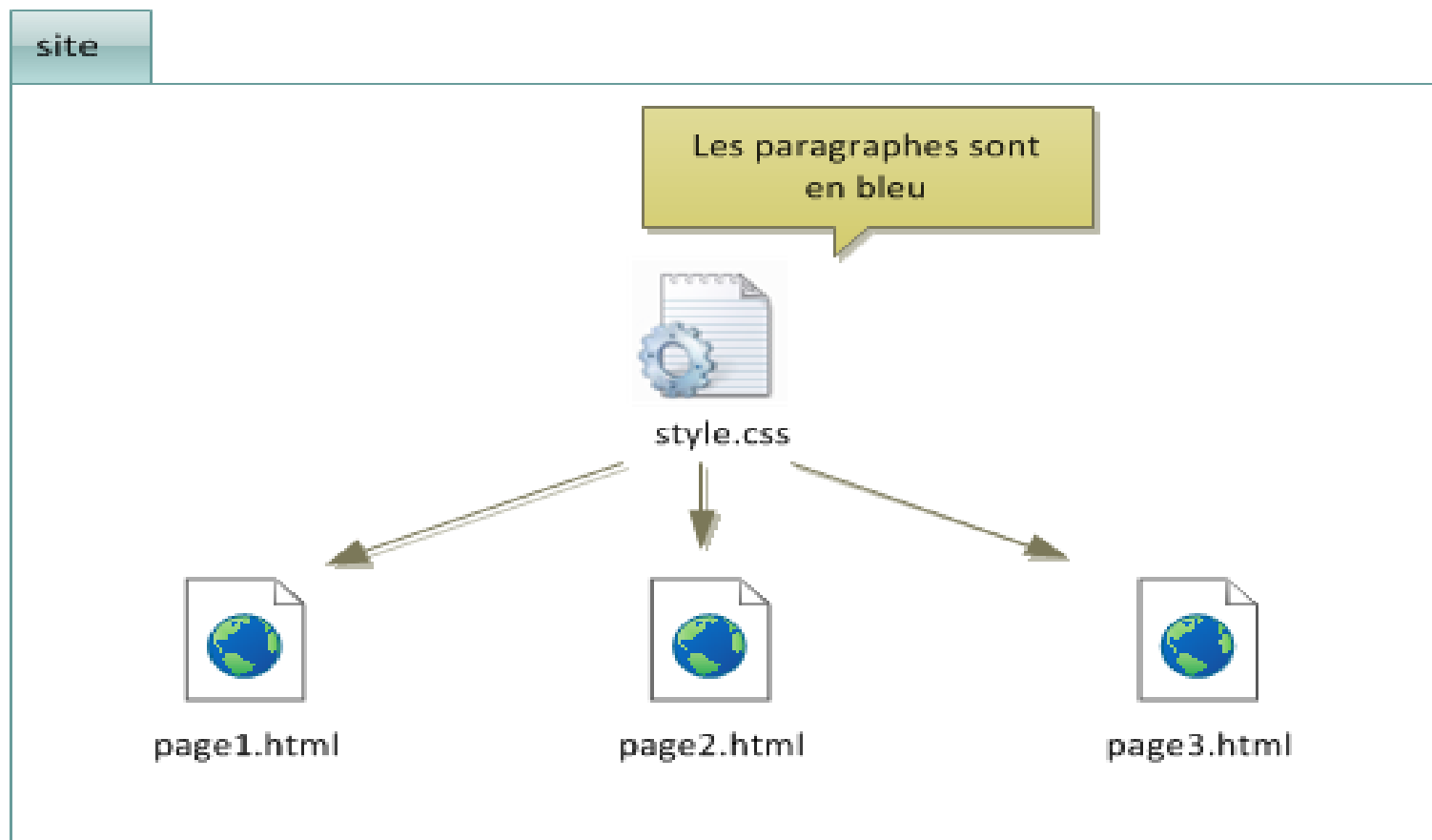
```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8" />  
    <title>Premiers tests du CSS</title>  
  </head>  
  
  <body>  
    <h1>Mon super site</h1>  
  
    <p style="color: blue;">Bonjour et bienvenue sur mon site !</p>  
    <p>Pour le moment, mon site est un peu <em>vide</em>.  
    Patiencez encore un peu !</p>  
  </body>  
</html>
```


Résultat de la méthode 3



Pour quoi choisir la méthode 1

- Votre site sera plus tard constitué de plusieurs pages HTML
- si vous placez le code CSS directement dans le fichier HTML, il faudra copier ce code dans tous les fichiers HTML de votre site
- Mais si 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,
- Si vous travaillez avec un fichier CSS externe, vous n'aurez besoin d'écrire cette instruction qu'une seule fois pour tout votre site



Appliquer un style : sélectionner une balise

```
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

Les valeurs CSS

- **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.

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;  
}
```

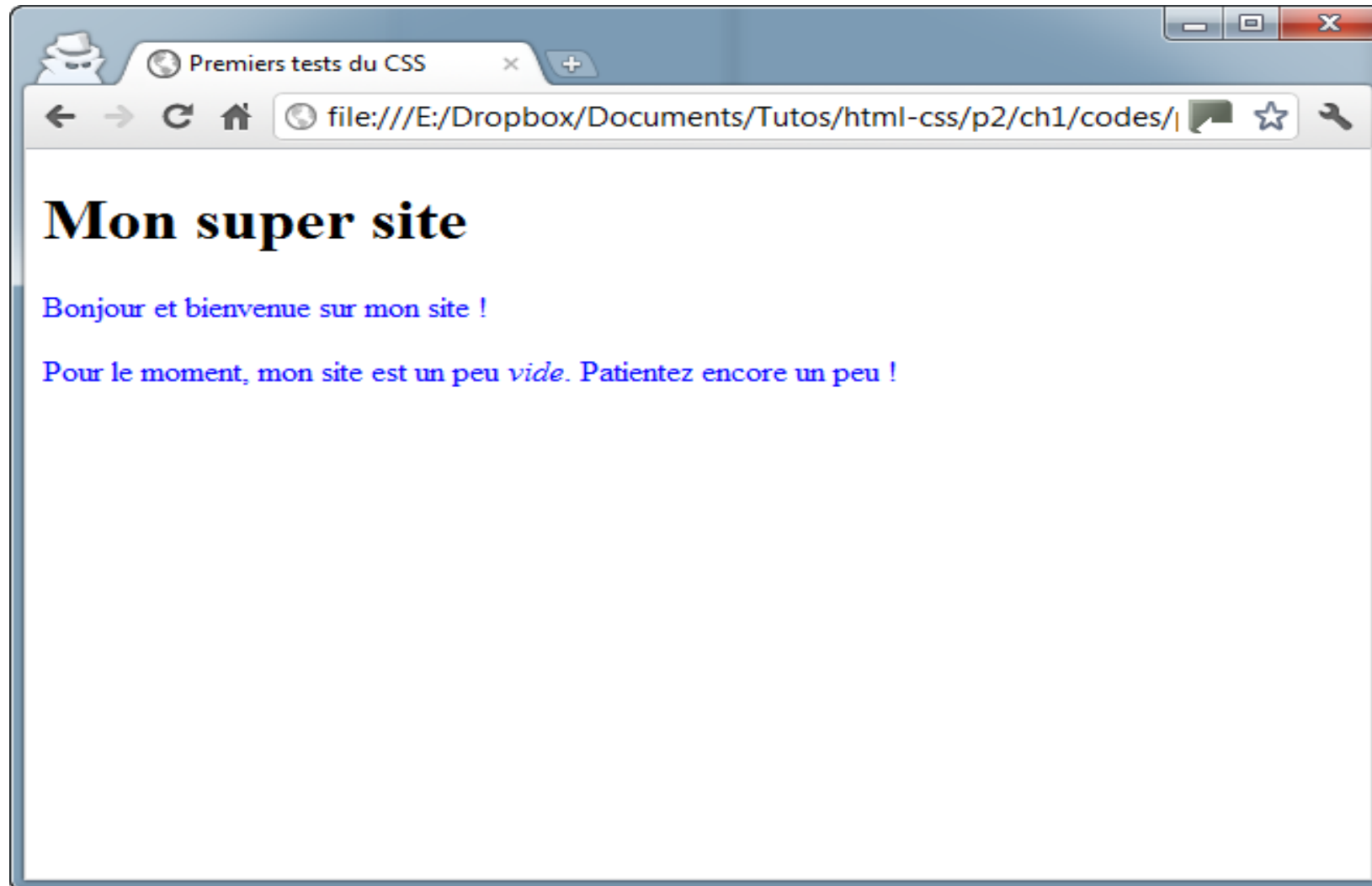
Exemple simple d'un code CSS

- 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** (`;`).
- Le code CSS que nous avons utilisé jusqu'ici :

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

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

Résultat de l'exemple



Changer le nom de balise

- 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 fichierstyle.css comme ceci :

h1

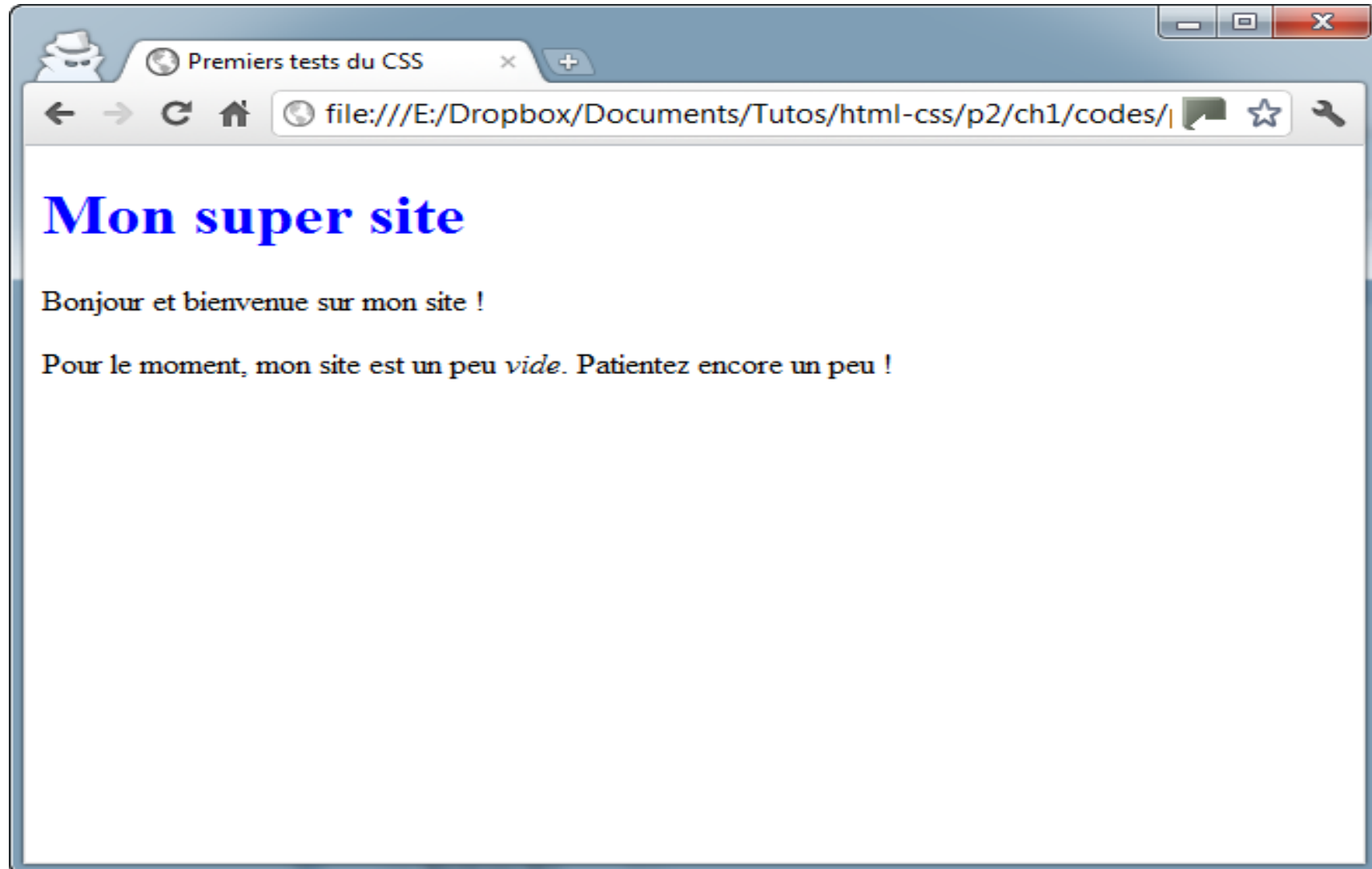
{

color: blue;

}

- Maintenant, ouvrez à nouveau votre page HTML : vous devriez voir son **titre s'afficher en bleu** (figure suivante)

Résultat du changement de la balise



Appliquer un style à plusieurs balises

- Prenons le code CSS suivant :

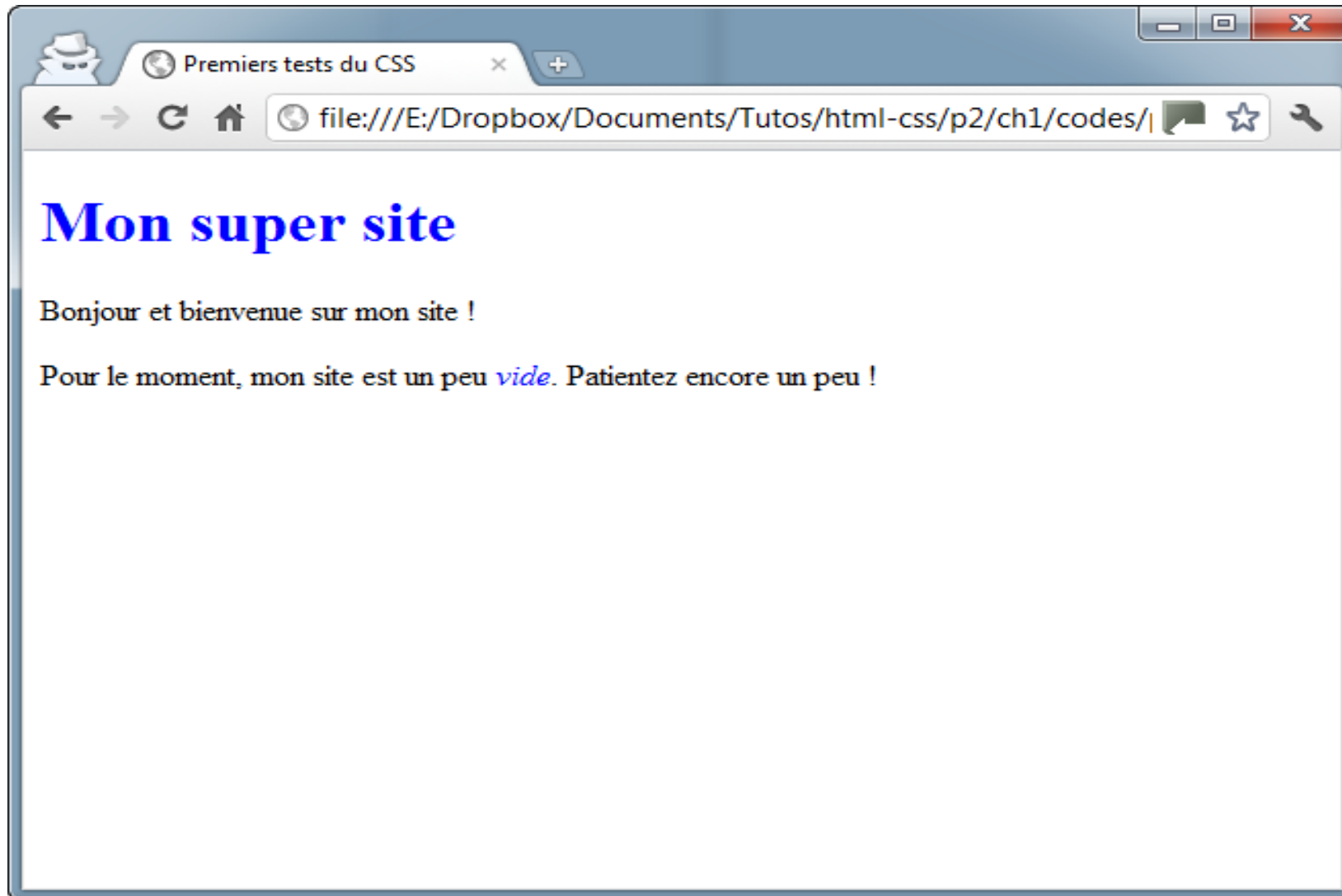
```
h1  
{  
  color: blue;  
}  
em  
{  
  color: blue;  
}
```

- Il signifie que nos titres<h1>et nos textes importantsdoivent s'afficher en bleu.
- 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;  
}
```

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

Résultat pour deux balise



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 dans un long fichier CSS.**
- le fichier HTML est assez court et la feuille CSS assez longue (elle contient tous les éléments de style de votre site).
- Notez **qu'il est possible de créer plusieurs fichiers CSS pour votre site**
- 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.**

Exemple de commentaires en CSS

```
/*  
style.css  
-----  
  
Pour le site web de la Géomatique  
*/  
  
p  
{  
    color: blue; /* Les paragraphes seront en bleu */  
}
```

Appliquer un style : **class et id**

- Le code CSS précédent implique que **TOUS les paragraphes possèdent la même présentation** ils seront tous écrits en **bleu** .
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**, ce n'est pas recommandé
- Pour résoudre le problème, on peut utiliser **ces attributs** spéciaux *qui fonctionnent sur toutes les balises* :
 - **l'attribut class;**
 - **l'attribut id.**
 - **les attributs class et id sont quasiment identiques.** Il y a seulement une petite différence qui sera dévoiler dans la suite du cours.

l'attribut class

- 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>`

``

- **La valeur à l'attribut class**
- 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) :

Exemple d'association d'une classe

```
<!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>
```

Identification de votre paragraphe

- 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

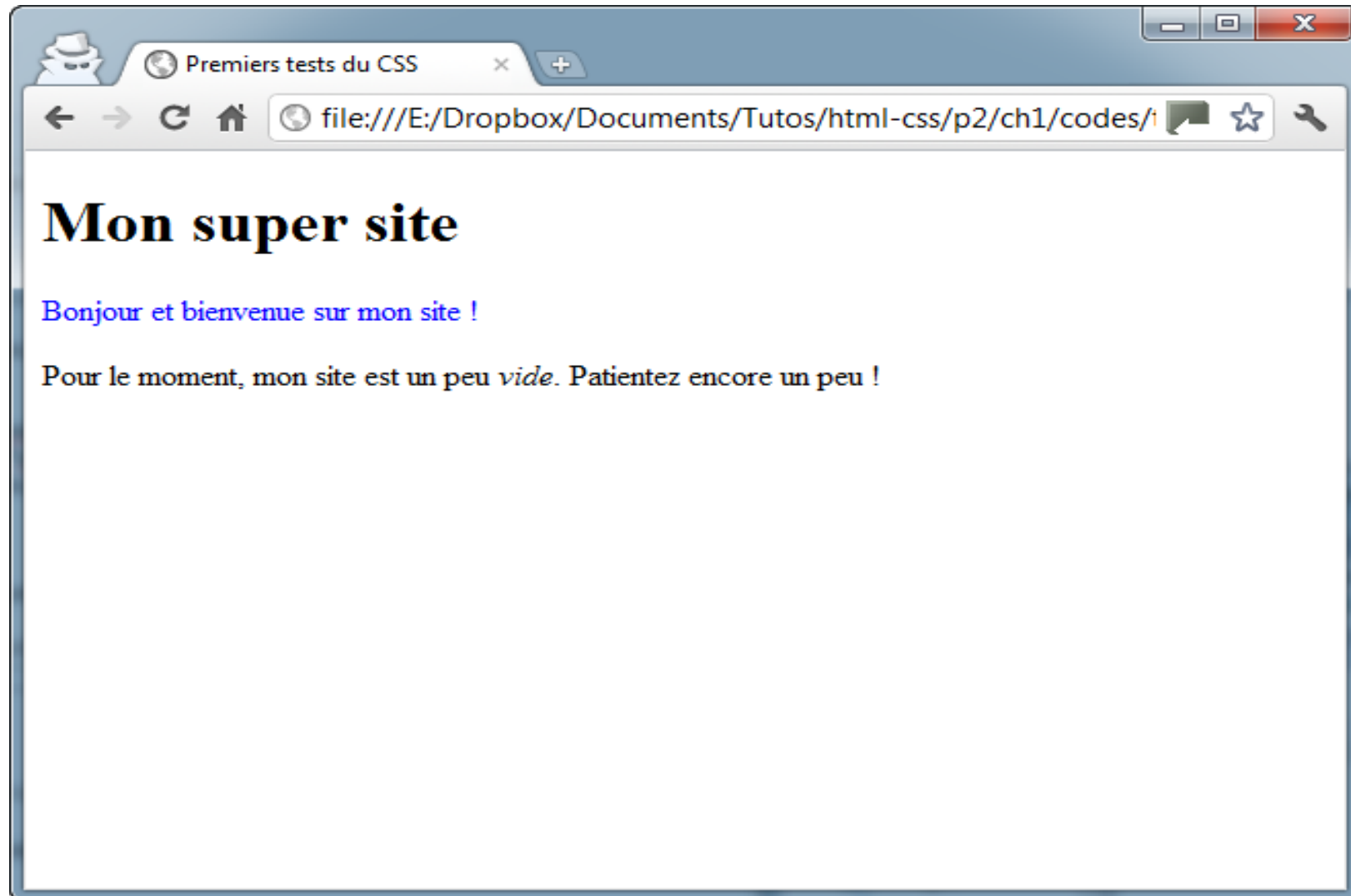
{

color: blue;

}

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

Résultat de votre exemple



l'attribut id

- 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 ?
- cela vous sera utile si vous faites du **JavaScript** plus tard **pour reconnaître certaines balises.**
- nous avons déjà vu l'attribut id dans le chapitre sur les liens (pour **réaliser des ancrés**).
- En pratique, nous **ne mettrons des id que sur des éléments qui sont uniques dans la page**, comme par exemple le logo :
- **``**

Définition des propriétés des id dans le fichier CSS

- 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 */

}

deux balises peuvent avoir le même nom avec l'attribut class.

Un nom d'id doit en revanche être unique dans la page HTML.

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.

Les balises universelles span

- En fait, il y a **deux balises dites universelles**, qui n'ont **aucune signification particulière**
- Il y a une différence minime (mais significative !) entre ces deux balises :
- **`` ``: 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 **``** et **``** 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 ».

Les balises universelles div

- **<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
- nous allons utiliser plutôt la balise ****.
- On la met autour de « bienvenue », on lui ajoute une classe (du nom qu'on veut), on crée le CSS.

Code HTML et CSS, class salutation balise span

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

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

Résultat de l'exemple



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**.

```
p  
{  
}
```

Si je veux toucher tous les paragraphes

```
h1, em  
{
```

```
}
```

Tous les titres et tous les textes importants

```
.class  
{  
}
```

```
#id  
{  
}
```

sélectionner des balises précises

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 balisessituées à l'intérieur d'une balise<h3>.
- Notez qu'il n'y a pas de virgule entre les deux noms de balises.

<h3>Titre avec texte important</h3>

Les sélecteurs avancés suite

- 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]

```
{  
}
```

- Exemple :

**

- 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 »**.

Les sélecteurs avancés suite

- **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 :

**

- 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](#) !

Formatage du texte la taille

- **Pour modifier la taille du texte**, on utilise la propriété CSS ***font-size***.
- Il existe plusieurs techniques parmi elles :
- Indiquer une **taille absolue** : en **pixels, en centimètres ou millimètres**.
- Cette méthode est très précise mais il est **conseillé de ne l'utiliser** que si c'est absolument nécessaire,
- car on risque d'indiquer **une taille trop petite pour certains lecteurs**.
- Indiquer une **taille relative** : en **pourcentage, « em » ou « ex »**, cette technique a l'avantage d'être **plus souple**.
- Elle **s'adapte plus facilement aux préférences de taille des visiteurs**.

Une taille absolue

- Pour indiquer une taille absolue, on utilise **généralement les pixels**.
- Pour avoir un texte de **16 pixels de hauteur**, vous devez donc écrire :

font-size: 16px;

- Voici un exemple d'utilisation (placez ce code dans votre fichier.css)

p
{

font-size: 14px; /* Paragraphes de 14 pixels */

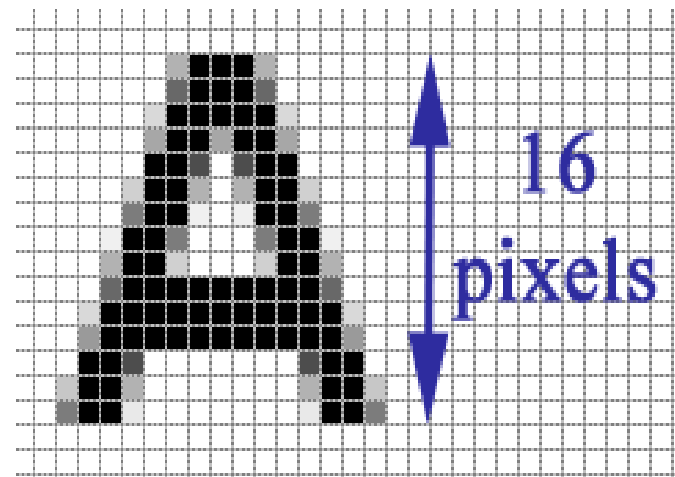
}

h1

{

font-size: 40px; /* Titres de 40 pixels */

}



Résultat de l'exemple



- Si vous le souhaitez, vous pouvez également **définir des tailles en centimètres ou millimètres**. Remplacez « px » par « **cm** » ou « **mm** ». Ces unités sont cependant moins bien adaptées aux écrans.

Une valeur relative

- C'est **la méthode recommandée** car **le texte s'adapte alors plus facilement aux préférences** de tous les visiteurs.
- Il y a **plusieurs moyens d'indiquer une valeur relative**. Vous pouvez par exemple écrire la taille avec des mots en anglais comme ceux-ci
- ***xx-small***: minuscule ;
- ***x-small***: très petit ;
- ***small***: petit ;
- ***medium***: moyen ;
- ***large***: grand ;
- ***x-large***: très grand ;
- ***xx-large***: gigantesque.
- Vous pouvez tester l'utilisation de ces valeurs dans votre code CSS :

Une valeur relative suite

```
p  
{  
  font-size: small;  
}  
h1  
{  
  font-size: large;  
}
```

- **cette technique a un défaut : il n'y a que sept tailles disponibles**
- il existe une autre technique qui consiste à **indiquer la taille en « em »**.
- Si vous écrivez **1em**, le texte a une **taille normale**.
- Si vous voulez **grossir le texte**, vous pouvez inscrire une valeur supérieure à 1, comme **1.3em**.
- Si vous voulez **réduire le texte**, inscrivez une valeur inférieure à 1, comme **0.8em**.

Une valeur relative suite

- Faites attention : pour les **nombre**s décimaux, il faut mettre un **point et non une virgule**. Vous devez donc écrire « 1.4em » et non pas « 1,4em » !
- Exemple

```
p  
{  
  font-size: 0.8em;  
}  
h1  
{  
  font-size: 1.3em;  
}
```

- D'autres unités sont disponibles. Vous pouvez essayer le « ex » (qui fonctionne sur le même principe que le em
- mais qui est plus petit de base) et le pourcentage (80%, 130%...).

La police

- pour qu'une police s'affiche correctement, il faut que tous les internautes l'aient.
- Si un internaute n'a pas la même police que vous, son navigateur prendra une police par défaut (une police standard) qui n'aura peut-être rien à voir avec ce à quoi vous vous attendiez.
- depuis CSS 3, il est possible de faire télécharger automatiquement une police au navigateur.
- **Modifier la police utilisée**
- La propriété CSS qui permet d'indiquer la police à utiliser est **font-family**.
- Vous devez écrire le nom de la police comme ceci :

balise

{

font-family: police;

}

Eviter les problèmes de police

- pour éviter les problèmes si l'internaute n'a pas la même police que vous, **on précise en général plusieurs noms de police, séparés par des virgules :**

balise

{

font-family: police1, police2, police3, police4;

}

- Le navigateur essaiera d'abord d'utiliser lapolice1. S'il ne l'a pas, il essaiera lapolice2. S'il ne l'a pas, il passera à lapolice3, et ainsi de suite.
- En général, **on indique en tout dernier serif**, ce qui correspond à une police par défaut (qui ne s'applique que si aucune autre police n'a été trouvée).
- Il existe aussi une autre **police par défaut appelée sans-serif**.
- La différence entre les deux est la présence de petites pattes de liaison en bas des lettres, que la police sans-serif n'a pas.

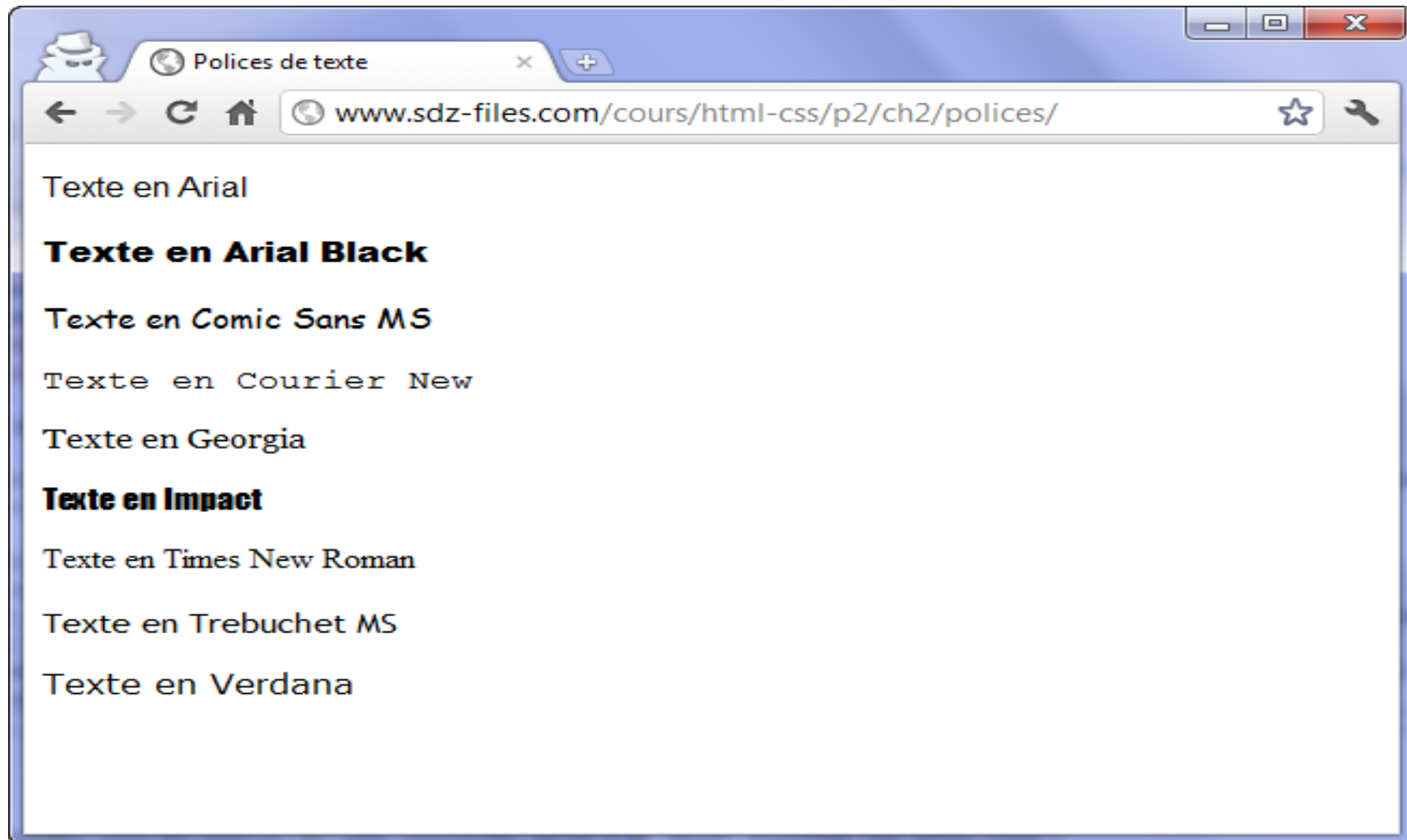


les polices les plus courantes

Voici une liste de polices qui fonctionnent bien sur la plupart des navigateurs :

- **Arial ;**
- **Arial Black ;**
- **Comic Sans MS ;**
- **Courier New ;**
- **Georgia ;**
- **Impact ;**
- **Times New Roman ;**
- **Trebuchet MS ;**
- **Verdana.**

La figure qui montre à quoi ressemblent ces polices.



Choix de la police

- Ainsi, si j'écris :

p

{

font-family: Impact, "Arial Black", Arial, Verdana, sans-serif;

}

- ... cela signifie : « Mets la police **Impact** ou, si elle n'y est pas, **Arial Black**, ou sinon **Arial**, ou sinon **Verdana**, ou si rien n'a marché, mets une police standard (sans-serif) ».
- En général, il est bien d'indiquer un choix de trois ou quatre polices (+ serif ou sans-serif) afin de s'assurer qu'au moins l'une d'entre elles aura été trouvée sur l'ordinateur du visiteur.
- Si le **nom de la police comporte des espaces**, je conseille de **l'entourer de guillemets**, comme je l'ai fait pour « Arial Black ».

italique

- ``Cela veut dire que **les mots qu'elle entoure sont assez importants**.

Pour représenter cette importance, la plupart des navigateurs choisissent d'afficher le texte en italique, mais **ce n'est pas une obligation**.

- Le **CSS lui, permet de dire réellement : « Je veux que ce texte soit en italique »**.
- en CSS, pour mettre en italique, on utilise **font-style** qui peut prendre trois valeurs :
- **italic**: le texte sera mis en **italique**.
- **oblique**: le texte sera passé en oblique (les lettres sont penchées, le résultat est légèrement différent de l'italique proprement dit).
- **normal**: le texte sera normal (par défaut).

Italique (suite)

- Cela vous permet **d'annuler une mise en italique**.
- Par exemple, si vous voulez que les textes entre `` ne **soient plus en italique**, vous devrez écrire :

em

{

font-style: normal;

}

- Ainsi, dans l'exemple suivant, je me sers de *font-style* pour mettre en italique tous mes titres `<h2>`:

h2

{

font-style: italic;

}

Gras

n'oubliez pas que ce n'est **pasqui permet de mettre en gras** (son rôle est d'indiquer que **le texte est important**, *donc* le navigateur l'affiche généralement en gras).

- La mise en gras en CSS peut par exemple s'appliquer aux titres, à certains paragraphes entiers, etc.
- La propriété CSS pour mettre en gras est ***font-weight*** et prend les valeurs suivantes :

Gras (Suite)

- ***bold***: le texte sera en gras ;
- ***normal***: le texte sera écrit normalement (par défaut).
- Voici par exemple comment écrire les titres en gras :

h1

{

font-weight: bold;

}

Soulignement et autres décorations

- La propriété CSS associée est : ***text-decoration***.
- Elle permet, entre autres, de **souligner le texte**, mais pas seulement. Voici les différentes valeurs qu'elle peut prendre :
- ***underline***: souligné.
- ***line-through***: barré.
- ***overline***: ligne au-dessus.
- ***blink***: clignotant. Ne fonctionne pas sur tous les navigateurs (Internet Explorer et Google Chrome, notamment).
- ***none***: normal (par défaut).
- Ce CSS va vous permettre de tester les effets de text-decoration:

Exemple de text-decoration

```
h1  
{  
  text-decoration: blink;  
}  
.souligne  
{  
  text-decoration: underline;  
}  
.barre  
{  
  text-decoration: line-through;  
}  
.ligne_dessus  
{  
  text-decoration: overline;  
}
```

Le résultat de l'exemple



L'alignement

- Le langage CSS nous permet de faire **tous les alignements connus : à gauche, centré, à droite et justifié.**
- On utilise la propriété ***text-align*** et on indique l'alignement désiré :
- ***left***: le texte sera aligné à gauche (c'est le réglage par défaut).
- ***center***: le texte sera centré.
- ***right***: le texte sera aligné à droite.
- ***justify***: le texte sera « justifié ».
- **Vous ne pouvez pas modifier l'alignement du texte d'une balise *inline* (comme ``, `<a>`, ``, ``...).**
- L'alignement ne **fonctionne que sur des balises de type *block* (`<p>`, `<div>`, `<h1>`, `<h2>`, ...)** C'est donc en général le paragraphe entier qu'il vous faudra aligner.
- Regardez les différents alignements sur cet exemple :

Exemple de l'alignement

```
h1  
{  
  text-align: center;  
}
```

```
p  
{  
  text-align: justify;  
}
```

```
.signature  
{  
  text-align: right;  
}
```

Résultat de l'exemple









La couleur et le fond

- Nous allons nous intéresser dans cette section aux **propriétés des couleurs**.
- comment changer la **couleur du texte** ;
- comment mettre **une couleur ou une image d'arrière-plan** ;
- comment **ajouter des ombres** ;
- comment **jouer avec les niveaux de transparence**.

Couleur du texte

- la propriété qui permet de modifier **la couleur du texte** : est ***color***
- Nous allons nous intéresser aux différentes façons d'indiquer la couleur, car **il y en a plusieurs**.
- **Indiquer le nom de la couleur**
- La méthode **la plus simple et la plus pratique pour choisir une couleur consiste à taper son nom**.
- Le seul défaut de cette méthode est qu'il n'existe que **seize couleurs dites « standard »**.
- D'autres couleurs officielles existent mais, comme elles ne fonctionneront pas forcément de la même manière sur tous les navigateurs, je vais éviter de vous les montrer.
- La figure suivante vous montre les seize couleurs que vous pouvez utiliser en tapant simplement leur nom

La liste des couleurs

white	
silver	
gray	
black	
red	
maroon	
lime	
green	
yellow	
olive	
blue	
navy	
fuchsia	
purple	
aqua	
teal	

Exemple de l'utilisation de color

- Pour passer tous les titres en bordeaux, on peut donc écrire :

```
h1
```

```
{
```

```
color: maroon;
```

```
}
```



La notation hexadécimale

- il existe en CSS plusieurs façons de choisir une couleur parmi toutes celles qui existent.
- La première est **la notation hexadécimale**. Elle est **couramment utilisée sur le Web**.
- Un **nom de couleur en hexadécimal**, cela ressemble à : **#FF5A28**. Pour faire simple, c'est une **combinaison de lettres et de chiffres** qui indiquent une couleur.

On doit toujours commencer par écrire **un dièse (#)**, suivi de **six lettres ou chiffres allant de 0 à 9 et de A à F**.

La notation hexadécimal suite

- Ces **lettres ou chiffres fonctionnent deux par deux**.
- **Les deux premiers indiquent une quantité de rouge,**
- **les deux suivants une quantité de vert**
- **et les deux derniers une quantité de bleu.**
- En **mélangeant ces quantités** (qui sont les composantes Rouge-Vert-Bleu de la couleur) on peut **obtenir la couleur qu'on veut**.
- **#000000 correspond à la couleur noire et**
- **#FFFFFF à la couleur blanche.**
- Certains logiciels de dessin, comme Photoshop, [Gimp](#) et [Paint.NET](#), vous indiquent **les couleurs en hexadécimal**. Il vous est alors facile de copier-coller le code hexadécimal d'une couleur dans votre fichier CSS.

La notation raccourcie

p

{

color: #FFFFFF;

}

- Notez qu'il existe **une notation raccourcie** : on peut écrire **une couleur avec seulement trois caractères**.
- Par exemple : **#FA3** équivaut à écrire **#FFAA33**.

La méthode RGB

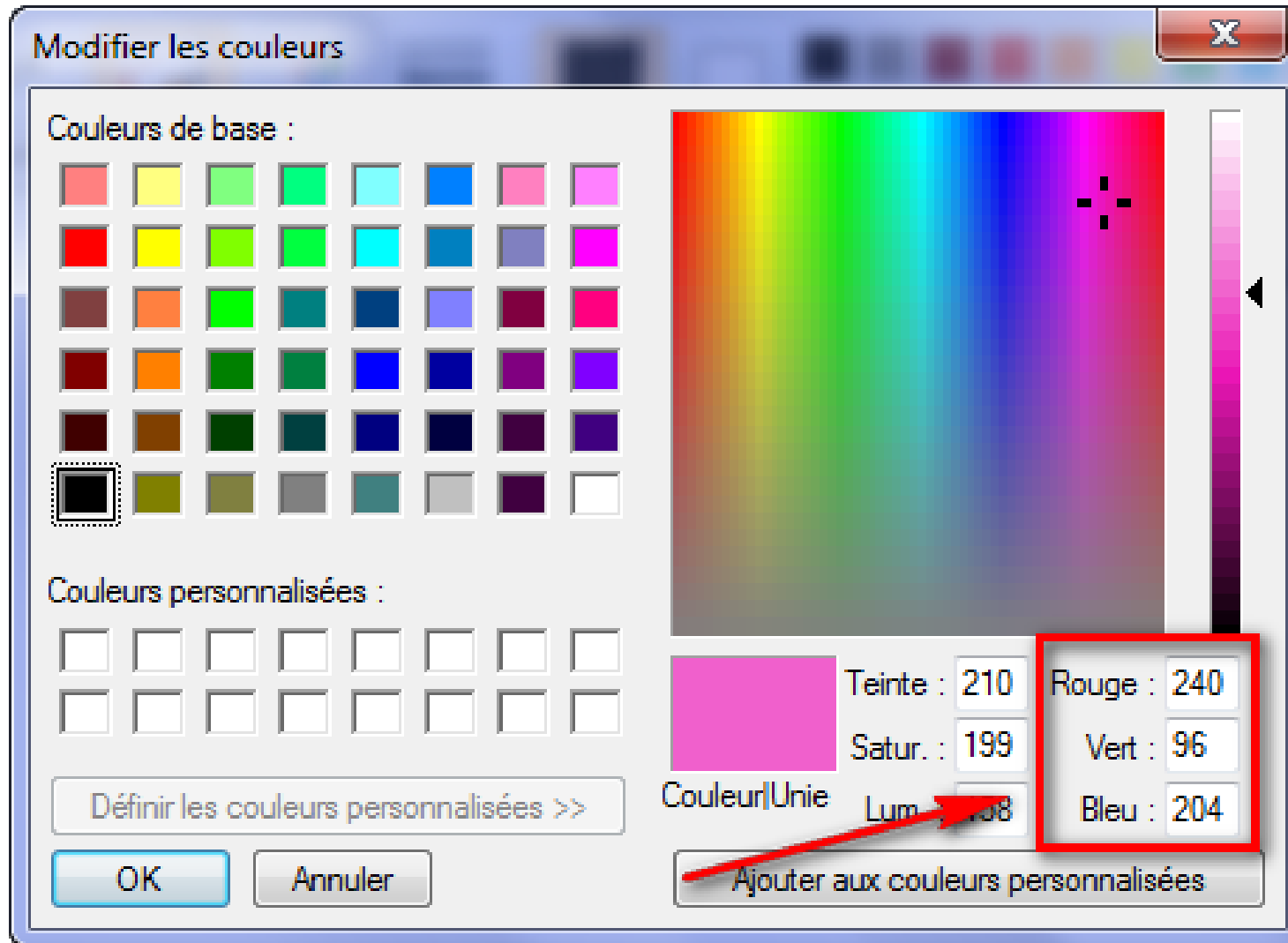
- Comme avec la notation hexadécimale, pour choisir une couleur, on doit **définir une quantité de rouge, de vert et de bleu**.
- Lancez le logiciel **Paint** depuis le menu Démarrer.
- Rendez-vous dans la section Modifier les couleurs, comme indiqué à la figure suivante.
- Une fenêtre s'ouvre. Dans la zone qui apparaît à droite, faites bouger les curseurs pour sélectionner la couleur qui vous intéresse.
- Sélectionnez la couleur dans la fenêtre, comme à la figure suivante.
- **Relevez les quantités de Rouge-Vert-Bleu** correspondantes, indiquées en bas à droite de la fenêtre (**ici 240-96-204**). **Recopiez ces valeurs dans cet ordre** dans le fichier CSS, comme dans le code ci-dessous.

```
p  
{
```

```
color: rgb(240,96,204);
```

```
}
```

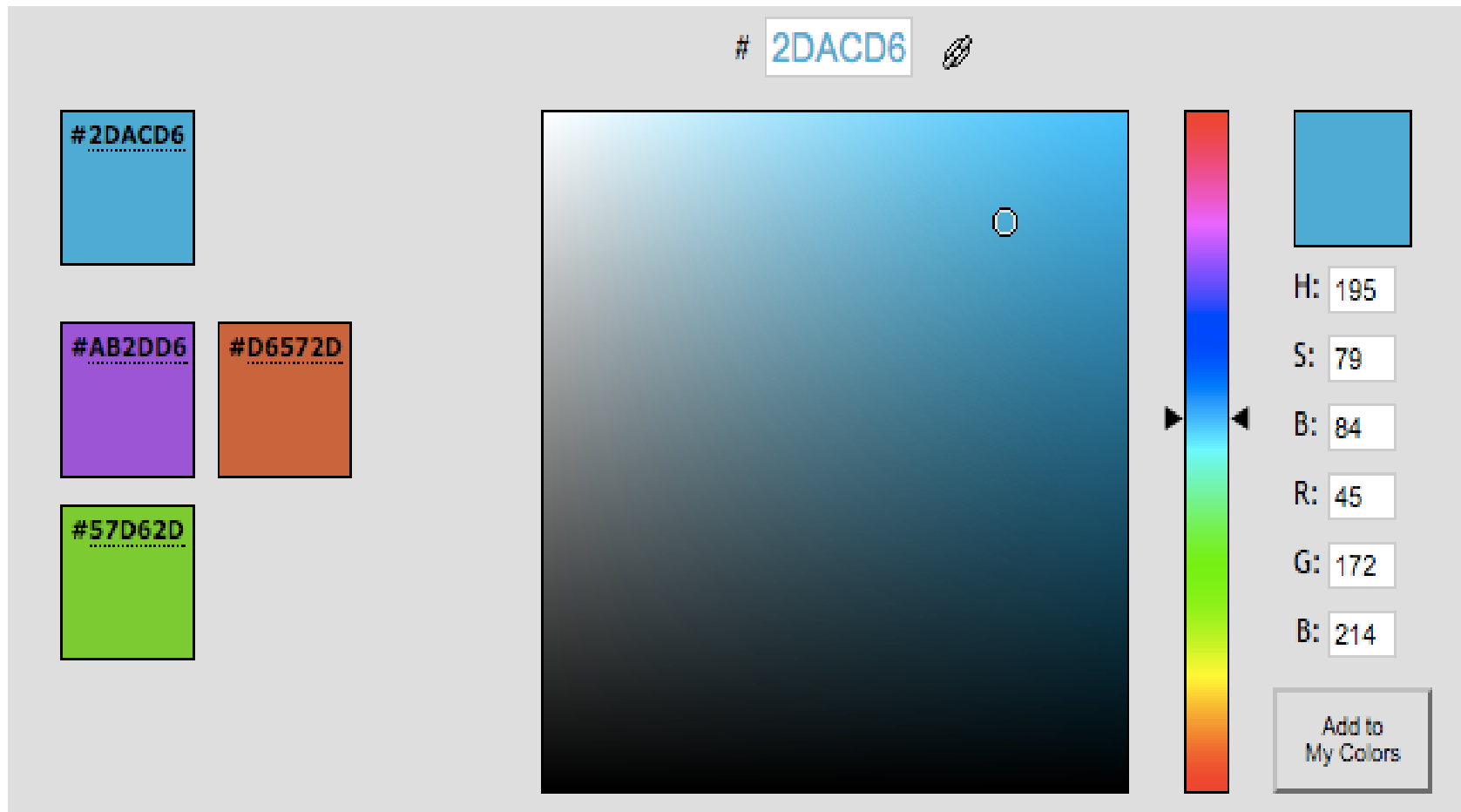
Exemple d'utilisation de paint



D'autres outils

- Comme vous avez pu le constater dans l'exemple, pour utiliser **la méthode RGB, il faut taper rgb(Rouge, Vert, Bleu)** en remplaçant « Rouge, Vert, Bleu » par les nombres correspondants.
- Pour information, ces quantités sont toujours **comprises entre 0 et 255**.
- **D'autres outils**
- taper "**Color Picker**" sur Google et vous trouverez plusieurs outils et sites qui vous aident à choisir une couleur.
- Par exemple, <http://www.colorpicker.com> permet de **trouver la valeur hexadécimale** d'une couleur très facilement :
- Vous pouvez aussi trouver **des extensions de navigateur** qui permettent de "récupérer" **n'importe quelle couleur qui vous plaît sur un site web** :
- [ColorPicker](#) pour **Firefox**
- [ColorZilla](#) pour **Chrome**

<https://www.webpagefx.com/web-design/color-picker/>



Couleur de fond

- Pour indiquer **une couleur de fond**, on utilise la propriété CSS **background-color**.
- Elle s'utilise de la même manière que la propriété **color**, c'est-à-dire que vous pouvez taper
- le **nom d'une couleur**,
- l'écrire **en notation hexadécimale**
- ou encore **utiliser la méthode RGB**.
- Pour indiquer **la couleur de fond de la page web**, il faut travailler **sur la balise <body>**, qui correspond à l'ensemble de la page web,
- */* On travaille sur la balise body, donc sur TOUTE la page */*

body

{

background-color: black; /* Le fond de la page sera noir */

color: white; /* Le texte de la page sera blanc */

}

Résultat de l'exemple



L'héritage

- tous les paragraphes<p>et titres<h1>ont pris cette couleur.
- **Ce phénomène s'appelle l'héritage.**
- **Le CSS et l'héritage**
- En CSS, **si vous appliquez un style à une balise, toutes les balises qui se trouvent à l'intérieur prendront le même style.**
- **La balise<body> contient entre autres les balises de paragraphe<p> et de titre<h1>.**
- Si j'applique une couleur de fond noire et une couleur de texte blanche à la balise<body>, tous mes titres et paragraphes auront eux aussi un arrière-plan de couleur noire et un texte de couleur blanche...
- C'est ce phénomène qu'on appelle l'héritage : **on dit que les balises qui se trouvent à l'intérieur d'une autre balise « héritent » de ses propriétés.**

L'héritage suite

- C'est d'ailleurs de là que vient le nom « CSS », qui signifie « *Cascading Style Sheets* », c'est-à-dire « **Feuilles de style en cascade** ».
- **Les propriétés CSS sont héritées en cascade : si vous donnez un style à un élément, tous les sous-éléments auront le même style.**
- Si vous dites **par la suite que vous voulez vos titres en rouge, ce style aura la priorité et vos titres seront donc en rouge.**
- En revanche, **si vous n'indiquez rien de particulier** (comme on l'a fait tout à l'heure), **alors vos titres hériteront de la couleur blanche.**

Cela ne fonctionne pas uniquement pour la couleur.

- **Toutes les propriétés CSS seront héritées** : vous pouvez par exemple demander **une mise en gras dans la balise <body> et tous vos titres et paragraphes seront en gras.**

Exemple d'héritage avec la balise<mark>

- On a tendance à croire qu'on ne peut modifier que la couleur de fond de la page.
- C'est faux : **vous pouvez changer le fond de n'importe quel élément** : vos titres, vos paragraphes, certains mots... Dans ce cas, **ils apparaîtront surlignés** (comme si on avait mis un coup de marqueur dessus).
- la balise<mark>qui permet de mettre en valeur certains mots

<h1>Qui a éteint la lumière ?</h1>

<p>Brr, il fait tout noir sur ce site, c'est un peu <mark>inquiétant</mark> comme ambiance non vous trouvez pas ?</p>

- Par défaut, le texte s'affiche sur un fond jaune. **Vous pouvez changer ce comportement en CSS :**

La balise la plus précise

body

```
{  
  background-color: black;  
  color: white;  
}
```

mark

```
{  
  /* La couleur de fond prend le pas sur celle de toute la page */  
  background-color: red;  
  color: black;  
}
```

Sur le texte de la balise<mark>, **c'est la couleur de fond rouge qui s'applique.**

En effet, même si le fond de la page est noir, c'est la propriété CSS de l'élément le plus précis qui a la priorité (figure suivante).

Résultat de l'exemple



Généralisation du même principe

- **Le même principe vaut pour toutes les balises HTML et toutes les propriétés CSS !** Si vous dites :
- mes **paragraphes ont une taille de 1.2 em ;**
- mes **textes importants () ont une taille de 1.4 em ;**
- ... on pourrait penser qu'**il y a un conflit.**
- **Le texte important fait partie d'un paragraphe,** quelle taille lui donner ? **1.2 em ou 1.4 em ?**
- Le CSS décide que **c'est la déclaration la plus précise qui l'emporte :**
- **commecorrespond à un élément plus précis que les paragraphes, le texte sera écrit en 1.4 em.**

Images de fond

- Dans les exemples qui suivent, **je vais modifier l'image de fond de la page.**
- Cependant, tout comme pour la couleur de fond, **n'oubliez pas que l'image de fond ne s'applique pas forcément à la page entière.**
- **On peut aussi mettre une image de fond derrière les titres, paragraphes, etc.**
- **Appliquer une image de fond**
- La propriété permettant d'indiquer une image de fond est
- ***background-image*.**
- Comme valeur, on doit renseigner `url("nom_de_l_image.png")`. Par exemple :

body

{

background-image: url("neige.png");

}

Ce qui nous donne la figure suivante.

Le résultat de l'exemple



Le format de l'image de fond

- Bien entendu, **votre fond n'est pas forcément en PNG**, il peut aussi être en **JPEG** ou en **GIF**.
L'adresse indiquant où se trouve l'image de fond peut être écrite en absolu (http://...) ou en relatif (fond.png).
- Attention lorsque vous écrivez **une adresse en relatif** dans le fichier CSS ! **L'adresse de l'image doit être indiquée *par rapport au fichier.css* et non pas par rapport au fichier.html.**
- Pour simplifier les choses, je vous conseille de placer **l'image de fond dans le même dossier que le fichier.css**(ou dans un sous-dossier).
- **Options disponibles pour l'image de fond**
- On peut **compléter la propriété background-image** que nous venons de voir par plusieurs autres propriétés qui permettent de changer le comportement de l'image de fond.

background-attachment: fixer le fond

- La propriété CSS **background-attachment** permet de « **fixer** » le fond.
- L'effet obtenu est intéressant car on voit alors le texte « glisser » par-dessus le fond.
- Deux valeurs sont disponibles :
- **fixed**: l'image de fond reste fixe ;
- **scroll**: l'image de fond défile avec le texte (par défaut).

body

{

background-image: url("neige.png");

background-attachment: fixed; /* Le fond restera fixe */

}

background-repeat: répétition du fond

- Par défaut, **l'image de fond est répétée en mosaïque.**
- Vous pouvez changer cela avec la propriété ***background-repeat***:
- ***no-repeat***: le fond ne sera pas répété. L'image sera donc unique sur la page.
- ***repeat-x***: le fond sera répété uniquement sur la première ligne, horizontalement.
- ***repeat-y***: le fond sera répété uniquement sur la première colonne, verticalement.
- ***repeat***: le fond sera répété en mosaïque (par défaut).
- Exemple d'utilisation :

```
body  
{  
background-image: url("soleil.png");  
background-repeat: no-repeat;  
}
```

background-position: position du fond

- On peut **indiquer où doit se trouver l'image de fond** avec ***background-position***.
- Cette propriété n'est **intéressante que si elle est combinée** avec ***background-repeat: no-repeat***; (un fond qui ne se répète pas).
- Vous devez donner à ***background-position*** deux valeurs en pixels pour **indiquer la position du fond par rapport au coin supérieur gauche** de la page (ou du paragraphe, si vous appliquez le fond à un paragraphe).
- Ainsi, si vous tapez :
 - ***background-position: 30px 50px***;
 - votre fond sera placé à 30 pixels de la gauche et à 50 pixels du haut. Il est aussi possible d'utiliser ces valeurs en anglais
 - ***top***: en haut ;
 - ***bottom***: en bas ;
 - ***left***: à gauche ;
 - ***center***: centré ;
 - ***right***: à droite.

background-position: position du fond (suite)

- Il est possible **de combiner ces mots**. Par exemple, pour aligner une image en haut à droite, vous taperez :

background-position: top right;

- Ainsi, si je **veux afficher un soleil en image de fond** (figure suivante),
- en **un unique exemplaire (no-repeat)**,
- **toujours visible (fixed)**
- **et positionné en haut à droite (top right)**, je vais écrire ceci :

body

{

background-image: url("soleil.png");

background-attachment: fixed; /* Le fond restera fixe */

background-repeat: no-repeat; /* Le fond ne sera pas répété */

background-position: top right; /* Le fond sera placé en haut à droite */

}

Résultat de l'exemple



Combiner les propriétés

- Si vous utilisez **beaucoup de propriétés en rapport avec le fond** (comme c'est le cas sur ce dernier exemple), vous pouvez utiliser une sorte de « **super-propriété** » appelée **background** dont la valeur peut **combiner plusieurs des propriétés vues précédemment**
- ***:background-image, background-repeat, background-attachment et background-position.***
- On peut donc tout simplement écrire :

body

{

background: url("soleil.png") fixed no-repeat top right;

}

- **L'ordre des valeurs n'a pas d'importance.** Vous pouvez combiner les valeurs dans n'importe quel ordre.
- Vous n'êtes pas obligés de mettre toutes les valeurs. Ainsi, si vous ne voulez pas écrire **fixed**, vous pouvez l'enlever sans problème.

Plusieurs images de fond

- Depuis CSS3, **il est possible de donner plusieurs images de fond à un élément.**
- Pour cela, il suffit de **séparer les déclarations par une virgule, comme ceci :**

body

{

***background: url("soleil.png") fixed no-repeat top right,
url("neige.png") fixed;***

}

- **La première image de cette liste sera placée par-dessus les autres** (figure suivante).
- Attention donc, **l'ordre de déclaration des images a son importance** : si vous inversez le soleil et la neige dans le code CSS précédent, vous ne verrez plus le soleil

Résultat de l'exemple



dans tous ces exemples, j'ai appliqué un fond à la page entière (body).

on peut appliquer un fond à n'importe quel élément (un titre, un paragraphe, certains mots d'un paragraphe, etc.).

La transparence

- Le CSS nous permet de jouer très facilement **avec les niveaux de transparence des éléments** ! Pour cela, nous allons utiliser des fonctionnalités de CSS3 : la propriété ***opacity*** et la notation ***RGBA***.
- **La propriété *opacity***
- La propriété ***opacity***, très simple, permet **d'indiquer le niveau d'opacité (c'est l'inverse de la transparence)**.
- Avec une **valeur de 1, l'élément sera totalement opaque** : c'est le comportement par défaut.
- Avec une **valeur de 0, l'élément sera totalement transparent**.
- Il faut donc choisir une **valeur comprise entre 0 et 1**. Ainsi, avec une valeur de **0.6, votre élément sera opaque à 60%**... et on verra donc à travers !

p

{

opacity: 0.6;

}

Exemple de transparence

Voici un exemple qui va nous permettre d'apprécier la transparence.

```
body  
{  
  background: url('neige.png');  
}
```

```
p  
{  
  background-color: black;  
  color: white;  
  opacity: 0.3;  
}
```

- Si vous appliquez la propriété **opacity** à un élément de la page, **tout le contenu de cet élément sera rendu transparent** (même les images, les autres blocs à l'intérieur, etc.).
- Si vous voulez **juste rendre la couleur de fond transparente, utilisez plutôt la notation RGBa** que nous allons découvrir.

Résultat de l'exemple



La notation RGBa

- Il s'agit en fait de **la notation RGB que nous avons vue précédemment**, mais avec **un quatrième paramètre : le niveau de transparence (appelé « canal alpha »)**.
- De la même façon que précédemment, avec une **valeur de 1, le fond est complètement opaque**.
- Avec **une valeur inférieure à 1, il est transparent**.

```
p  
{
```

```
background-color: rgba(255, 0, 0, 0.5); /* Fond rouge à moitié transparent */
```

```
}
```

- Vous pouvez obtenir exactement le même effet qu'avec **opacity**
- Pour **les navigateurs plus anciens, il est recommandé d'indiquer la notation RGB classique en plus de RGBa**.
Pour ces navigateurs, le fond ne sera alors **pas transparent mais, au moins, il y aura bien une couleur d'arrière-plan**.

```
p  
{
```

```
background-color: rgb(255,0,0); /* Pour les navigateurs anciens */
```

```
background-color: rgba(255,0,0,0.5); /* Pour les navigateurs plus récents */
```

```
}
```

Les bordures et les ombres

- **Bordures standard**
- Le CSS vous offre un **large choix de bordures pour décorer votre page**. De nombreuses propriétés CSS vous permettent de modifier l'apparence de vos bordures :
 - **border-width**
 - **border-color**
 - **border-style...**
- Pour aller à l'essentiel, je vous propose ici d'utiliser directement la **super-propriété border** qui regroupe l'ensemble de ces propriétés.
- Pour **border** on peut utiliser jusqu'à trois valeurs pour modifier l'apparence de la bordure :
 - **La largeur** : indiquez la largeur de votre bordure. Mettez une valeur en **pixels (comme 2px)**.
 - **La couleur** : c'est la couleur de votre bordure. Utilisez, comme on l'a appris, soit un **nom de couleur** (black, red, ...), soit une **valeur hexadécimale (#FF0000)**, soit une valeur **RGB (rgb(198, 212, 37))**.

Le type de bordure

- Voici les différentes valeurs disponibles
 - **none**: pas de bordure (par défaut) ;
 - **solid**: un trait simple ;
 - **dotted**: pointillés ;
 - **dashed**: tirets ;
 - **double**: bordure double ;
 - **groove**: en relief ;
 - **ridge**: autre effet relief ;
 - **inset**: effet 3D global enfoncé ;
 - **outset**: effet 3D global surélevé.
- Ainsi, pour avoir une **bordure bleue, en tirets, épaisse de 3 pixels** autour de mes titres, je vais écrire :

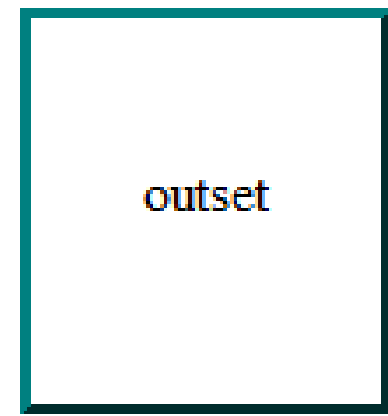
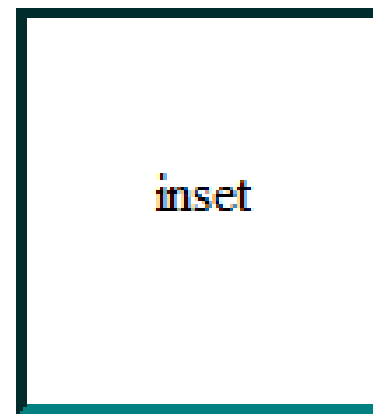
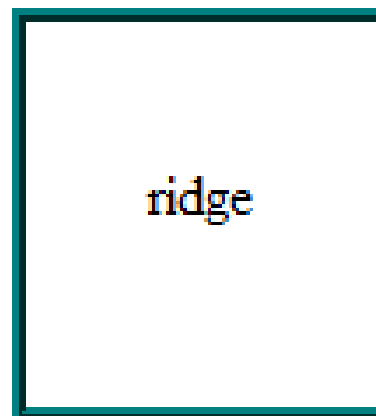
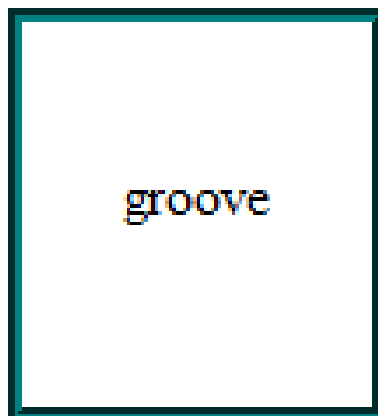
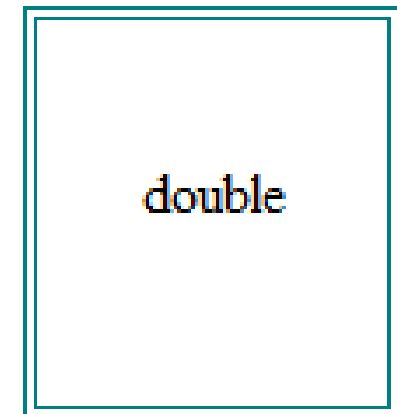
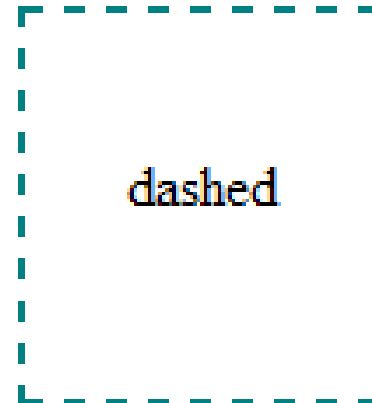
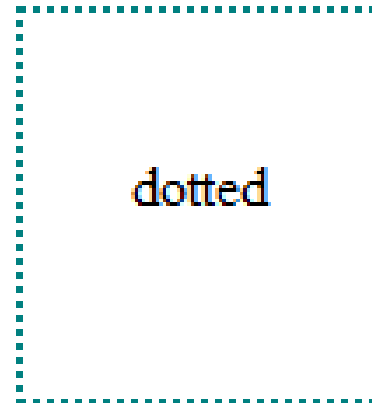
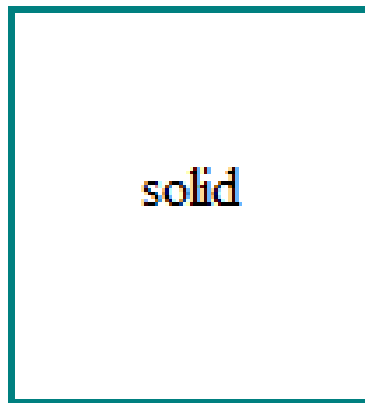
h1

{

border: 3px blue dashed;

}

les différents styles de bordures



La position

- **En haut, à droite, à gauche, en bas...**
- si vous voulez mettre **des bordures différentes en fonction du côté** (haut, bas, gauche ou droite) vous devrez utiliser ces quatre propriétés :
- ***border-top***: bordure du haut ;
- ***border-bottom***: bordure du bas ;
- ***border-left***: bordure de gauche ;
- ***border-right***: bordure de droite.
- Il existe aussi des équivalents pour paramétrer chaque détail de la bordure si vous le désirez :
- ***border-top-width*** pour modifier l'épaisseur de la bordure du haut,
- ***border-top-color*** pour la couleur du haut, etc.

Border

- Ces super-propriétés fonctionnent comme ***border*** mais **ne s'appliquent donc qu'à un seul côté.**
- Pour ajouter **une bordure uniquement à gauche** et à **droite des paragraphes**, on écrira donc :

```
p  
{  
border-left: 2px solid black;  
border-right: 2px solid black;  
}
```

- **On peut modifier les bordures de n'importe quel type d'élément sur la page.**
- Nous l'avons fait ici sur les paragraphes mais on peut aussi modifier
- la bordure des images,
- des textes importants **comme**, etc.

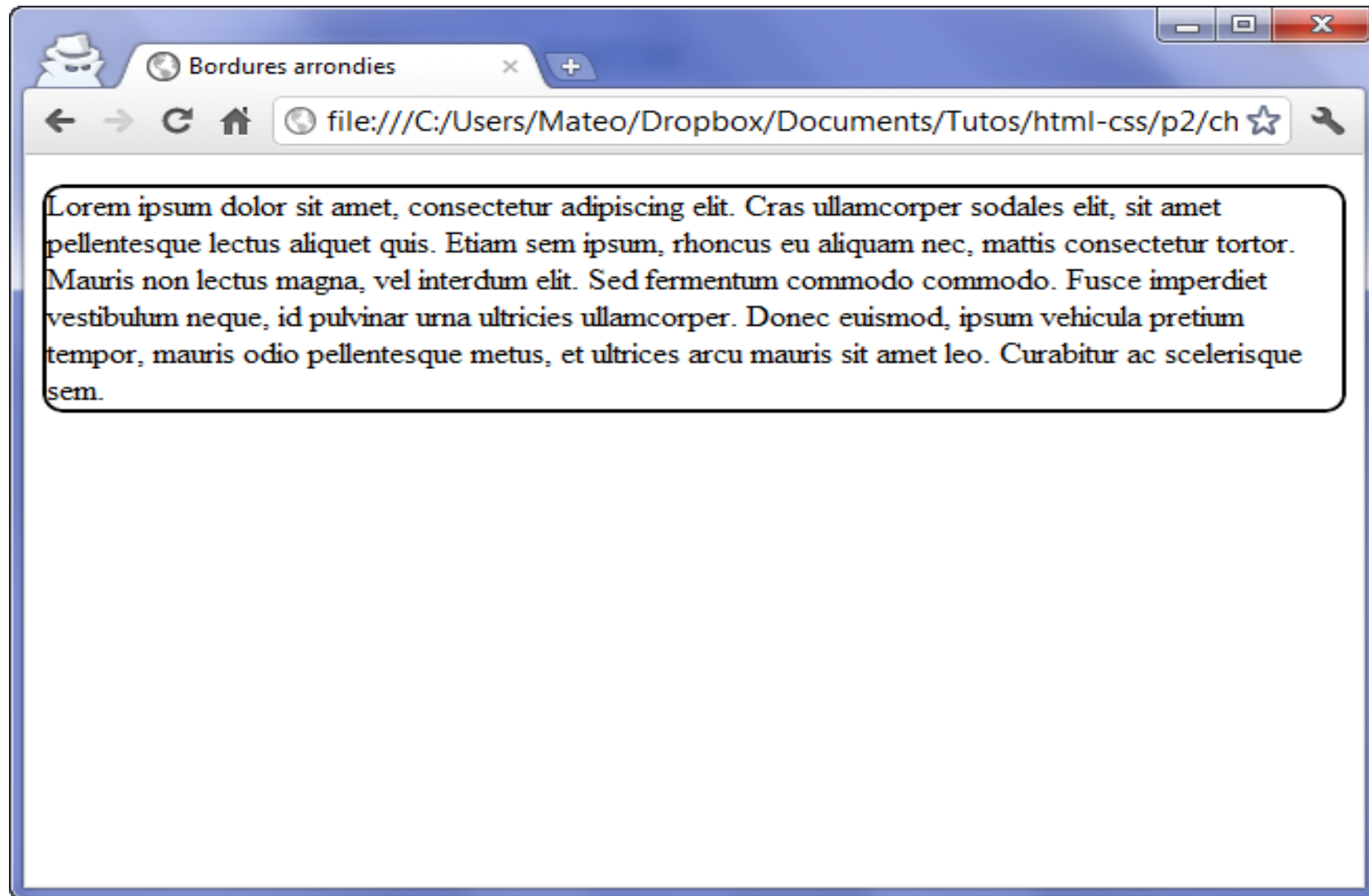
Bordures arrondies

- Depuis que CSS3 est arrivé, il est enfin possible d'en créer facilement des **bordures arrondies** !
- La propriété ***border-radius*** va nous permettre d'arrondir facilement les angles de n'importe quel élément.
- Il suffit **d'indiquer la taille** (« l'importance ») de **l'arrondi** en pixels :

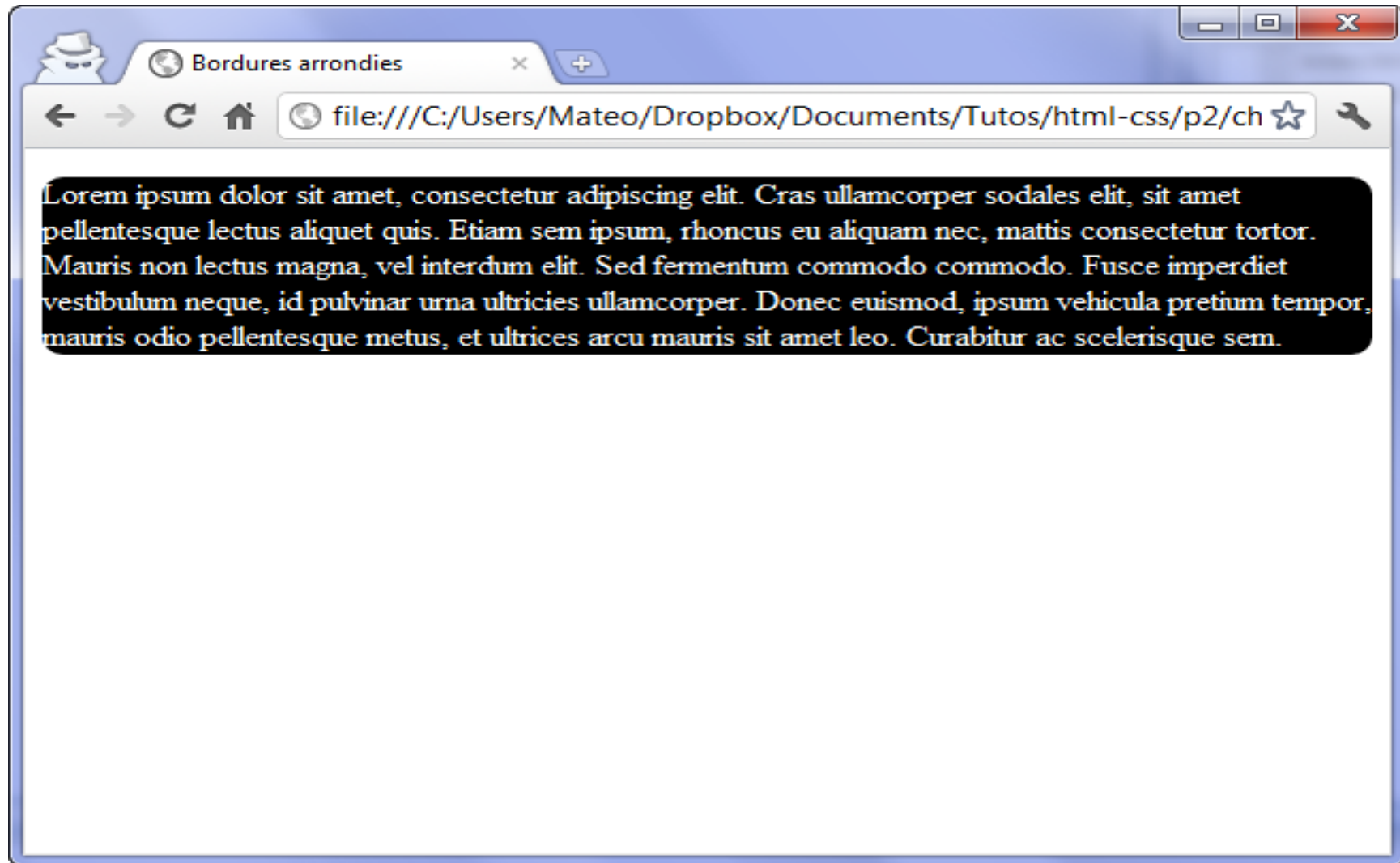
```
p  
{  
border-radius: 10px;  
}
```

- L'arrondi se voit notamment **si l'élément a des bordures**, comme sur la figure suivante.

Exemple de bordure arrondie



Bordure Arrondi avec couleur de fond



La forme de l'arrondie

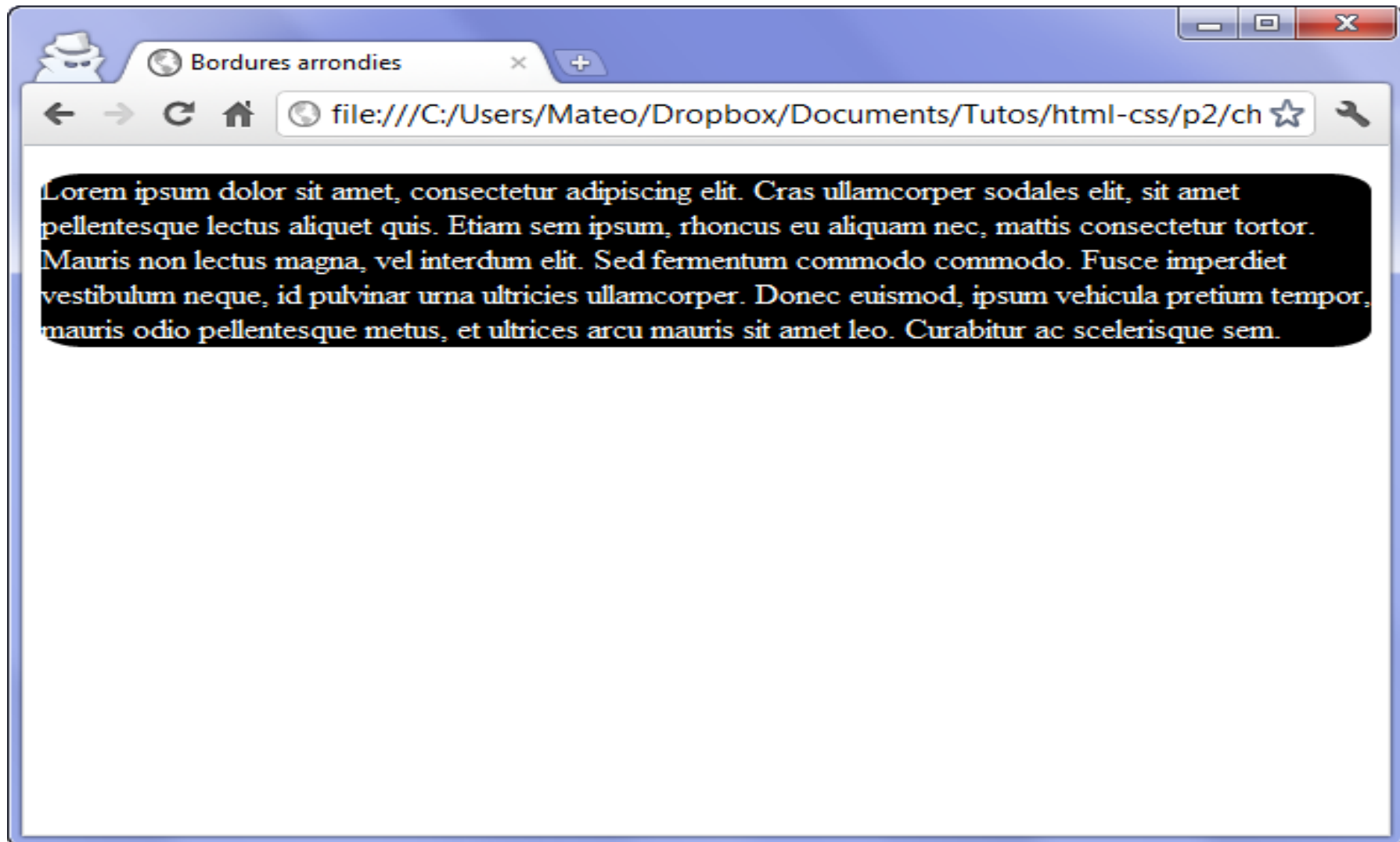
- On peut aussi préciser la forme de **l'arrondi pour chaque coin**. Dans ce cas, indiquez **quatre valeurs** :

```
p  
{  
border-radius: 10px 5px 10px 5px;  
}
```

- Les valeurs correspondent aux angles suivants dans cet ordre :
 - **en haut à gauche ;**
 - **en haut à droite ;**
 - **en bas à droite ;**
 - **en bas à gauche.**
- Enfin, il est possible **d'affiner l'arrondi** de nos angles en créant des **courbes elliptiques** (figure suivante).
- Dans ce cas, il faut indiquer **deux valeurs séparées par une barre oblique** (*slash*, caractère **/**).

```
p  
{  
border-radius: 20px / 10px;  
}
```

Résultat de l'exemple



Les ombres

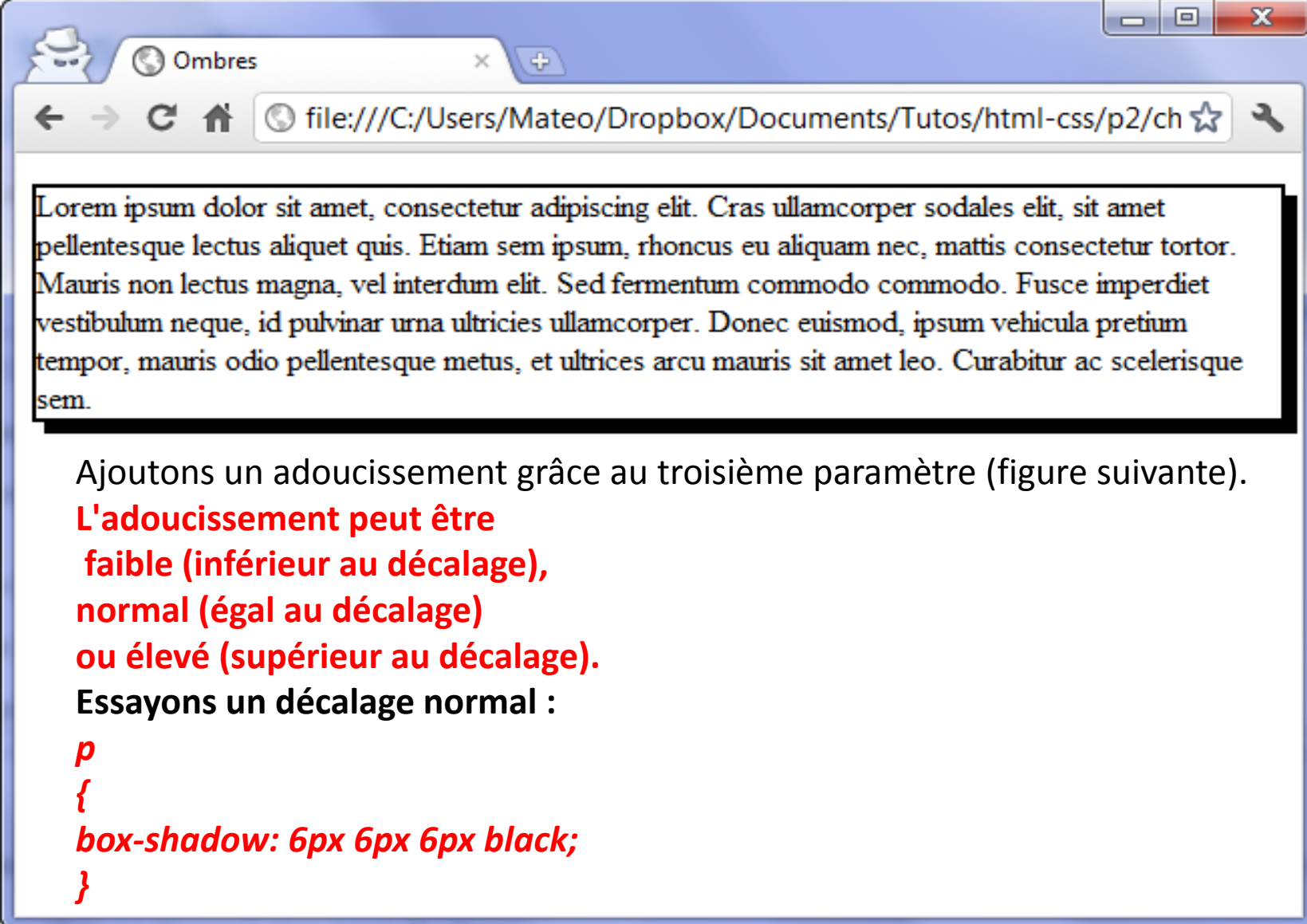
- Les ombres font partie des nouveautés récentes proposées par CSS3.
- Nous allons ici découvrir deux types d'ombres :
- **les ombres des boîtes ;**
- **les ombres du texte.**
- ***box-shadow*: les ombres des boîtes**
- La propriété ***box-shadow*** s'applique à **tout le bloc** et prend **quatre valeurs** dans l'ordre suivant :
- **le décalage horizontal de l'ombre ;**
- **le décalage vertical de l'ombre ;**
- **l'adoucissement du dégradé ;**
- **la couleur de l'ombre.**
- Par exemple, pour une ombre noire de 6 pixels, sans adoucissement, on écrira :

```
p  
{
```

```
  box-shadow: 6px 6px 0px black;
```

```
}
```


Résultat de l'exemple

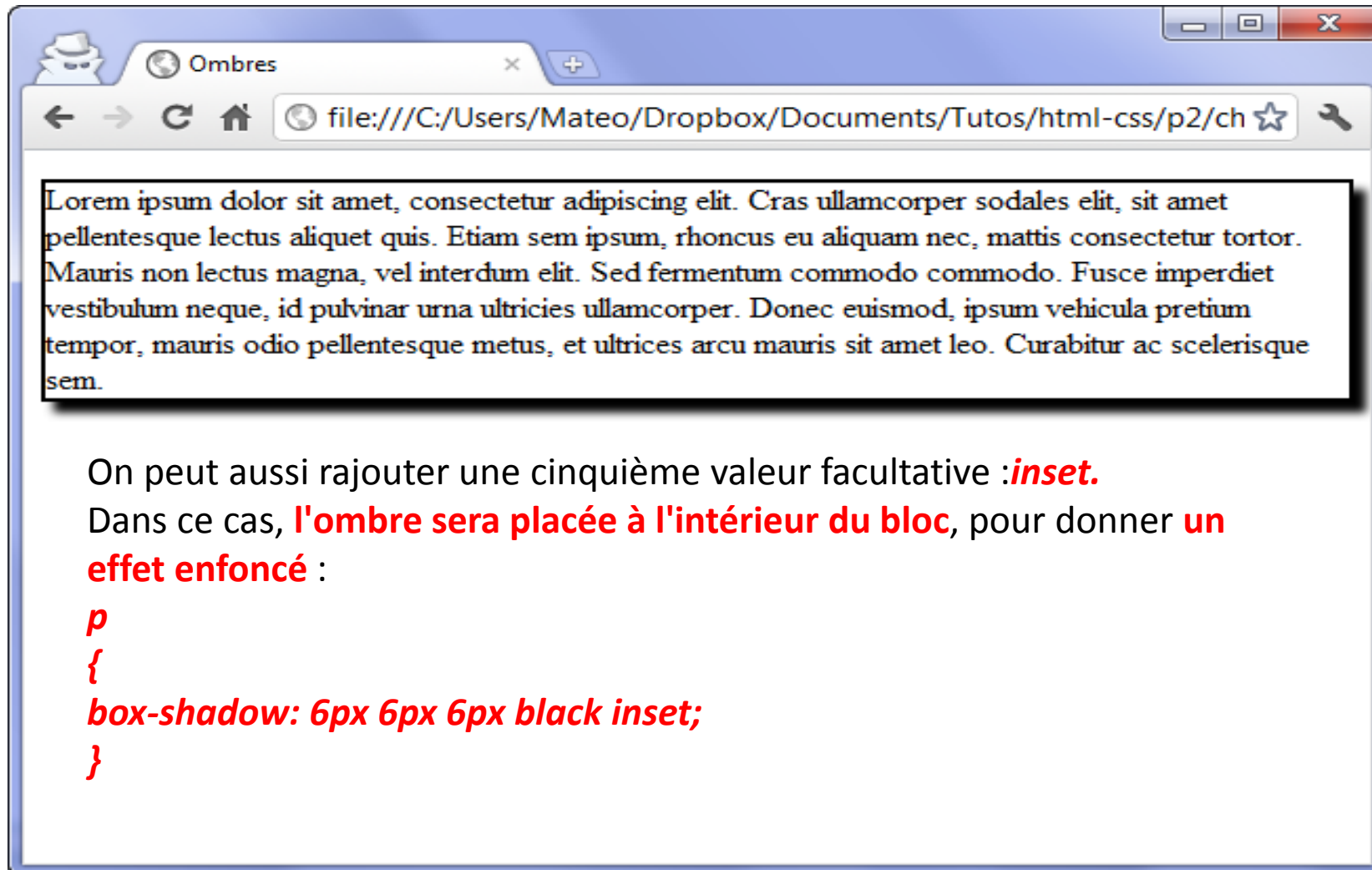
A screenshot of a web browser window. The browser has a single tab titled "Ombres". The address bar shows a file path: "file:///C:/Users/Mateo/Dropbox/Documents/Tutos/html-css/p2/ch". The main content area displays a paragraph of Lorem Ipsum text enclosed in a black-bordered box with a drop shadow. Below the screenshot, there is a text block explaining the shadow effect and providing CSS code.

>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ullamcorper sodales elit, sit amet pellentesque lectus aliquet quis. Etiam sem ipsum, rhoncus eu aliquam nec, mattis consectetur tortor. Mauris non lectus magna, vel interdum elit. Sed fermentum commodo commodo. Fusce imperdiet vestibulum neque, id pulvinar urna ultricies ullamcorper. Donec euismod, ipsum vehicula pretium tempor, mauris odio pellentesque metus, et ultrices arcu mauris sit amet leo. Curabitur ac scelerisque sem.

Ajoutons un adoucissement grâce au troisième paramètre (figure suivante).
L'adoucissement peut être faible (inférieur au décalage), normal (égal au décalage) ou élevé (supérieur au décalage).
Essayons un décalage normal :

```
p  
{  
  box-shadow: 6px 6px 6px black;  
}
```

La même ombre avec adoucissement



The screenshot shows a web browser window with a single tab titled "Ombres". The address bar contains the file path: `file:///C:/Users/Mateo/Dropbox/Documents/Tutos/html-css/p2/ch`. The main content area displays a paragraph of Lorem Ipsum text enclosed in a black-bordered box with a soft, blurred shadow effect. Below the browser window, there is a text block explaining the `inset` property of `box-shadow`.

On peut aussi rajouter une cinquième valeur facultative : ***inset***.
Dans ce cas, **l'ombre sera placée à l'intérieur du bloc**, pour donner **un effet enfoncé** :

```
p  
{  
box-shadow: 6px 6px 6px black inset;  
}
```

l'ombre du texte

- ***text-shadow***: l'ombre du texte
- Avec ***text-shadow***, vous pouvez ajouter une ombre directement sur les lettres de votre texte !
- Les valeurs fonctionnent exactement de la même façon que ***box-shadow***: décalage, adoucissement et couleur.

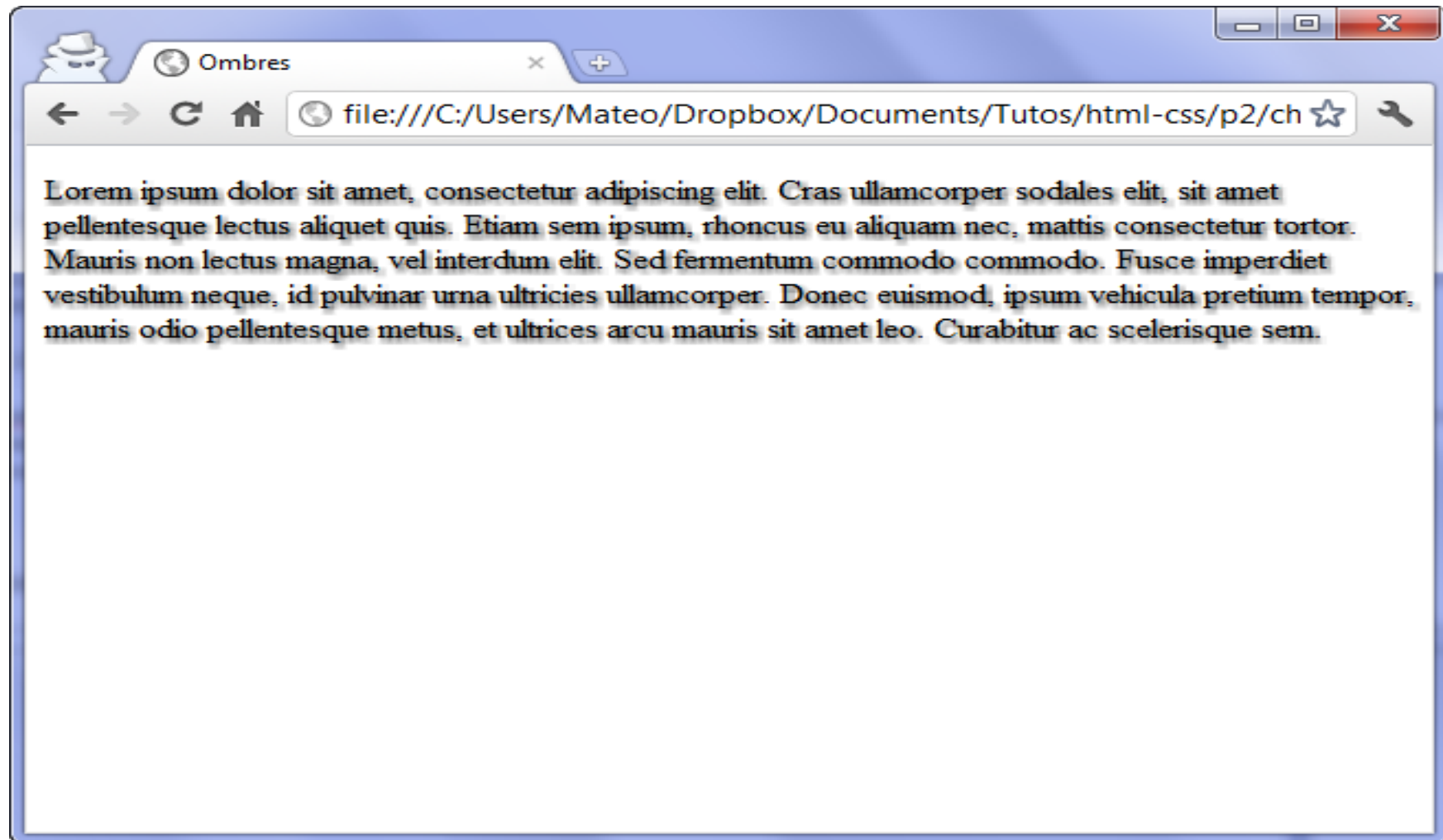
p

{

text-shadow: 2px 2px 4px black;

}

Résultat de l'exemple



Création d'apparences dynamiques

- C'est une de ses forces : le CSS nous permet aussi de modifier **l'apparence des éléments de façon dynamique**,
- c'est-à-dire que des **éléments peuvent changer de forme une fois que la page a été chargée**.
- Nous allons faire appel à une **fonctionnalité puissante du CSS : les pseudo-formats**.
- Nous verrons dans cette section comment changer l'apparence :
 - **au survol ;**
 - **lors du clic ;**
 - **lors du focus** (élément sélectionné) ;
 - **lorsqu'un lien a été consulté.**

Au survol

- Nous allons découvrir dans ce chapitre plusieurs **pseudo-formats CSS**. Le premier est ***:hover***.
- Comme tous les autres pseudo-formats que nous allons voir, c'est une information que **l'on rajoute après le nom de la balise** (ou de la classe) dans le CSS, comme ceci :

a:hover

```
{  
}
```

- ***:hover*** signifie « ***survoler*** ». ***a:hover*** peut donc se traduire par : « ***Quand la souris est sur le lien*** » (quand on pointe dessus).
- À partir de là, c'est à vous de définir l'apparence que doivent avoir les liens lorsqu'on pointe dessus
- Voici un exemple de présentation des liens:

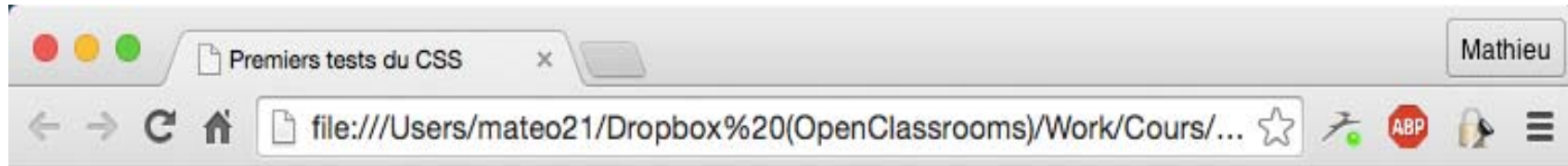
Exemple

```
a /* Liens par défaut (non survolés) */  
{  
  text-decoration: none;  
  color: red;  
  font-style: italic;  
}
```

```
a:hover /* Apparence au survol des liens */  
{  
  text-decoration: underline;  
  color: green;  
}
```

- On a défini ici deux versions des styles pour les liens :
- **pour les liens par défaut (non survolés) ;**
- **pour les liens au survol.**
- Le résultat se trouve à la figure suivante.

Résultat de l'exemple



Quelques bonnes adresses

Vous connaissez *Google* ? C'est le moteur de recherche le plus utilisé au monde !

Vous connaissez *le W3C* ? Ce sont les personnes qui définissent HTML et CSS.

Vous connaissez *OpenClassrooms* ? Ah ben oui, quelle question stupide...



Effet de survol sur les autres éléments

- Même si on l'utilise souvent sur les liens, vous pouvez modifier l'apparence de n'importe quel élément.
- Par exemple, **vous pouvez modifier l'apparence des paragraphes lorsqu'on pointe dessus :**

p:hover /* Quand on pointe sur un paragraphe */

{
}

Au clic et lors de la sélection

- nous pouvons **changer l'apparence des éléments lorsque l'on clique dessus et lorsqu'ils sont sélectionnés !**
- ***:active***: au moment du clic
- Le pseudo-format ***:active*** permet d'**appliquer un style particulier au moment du clic**.
- En pratique, il n'est utilisé que sur les liens.
- Le lien gardera cette apparence très peu de temps : en fait, le changement **intervient lorsque le bouton de la souris est enfoncé**.
- En clair, ce n'est pas forcément toujours bien visible.
- On peut par exemple changer la couleur de fond du lien lorsque l'on clique dessus :

a:active /* Quand le visiteur clique sur le lien */

{

background-color: #FFCC66;

}

:focus: lorsque l'élément est sélectionné

- Là, c'est un peu différent. Le pseudo-format **:focus** applique un **style lorsque l'élément est sélectionné**.
- C'est-à-dire
- Une fois que vous avez cliqué, le lien reste « sélectionné » (il y a une petite bordure en pointillés autour). C'est cela, la sélection.
- Ce pseudo-format pourra être appliqué à d'autres balises HTML que nous n'avons pas encore vues, comme les éléments de formulaires.
- Essayons pour l'instant sur les liens :

```
a:focus /* Quand le visiteur sélectionne le lien */  
{  
background-color: #FFCC66;  
}
```

- Sous Google Chrome et Safari, l'effet ne se voit que si l'on appuie sur la touche **Tab**.

Lorsque le lien a déjà été consulté

- Il est possible d'appliquer **un style à un lien vers une page qui a déjà été vue**.
- Par défaut, le navigateur **colore le lien en un violet**.
- Vous pouvez changer cette apparence avec ***:visited***(qui signifie « visité »).
- En pratique, sur les liens consultés, on ne peut pas changer beaucoup de choses à part la couleur (figure suivante).

a:visited /* Quand le visiteur a déjà vu la page concernée */

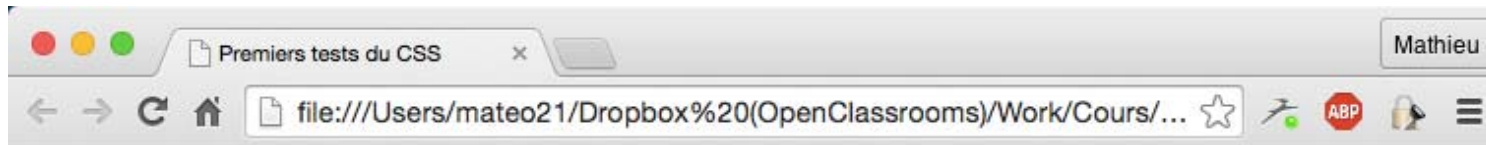
{

color: #AAA; /* Appliquer une couleur grise */

}

- Si vous ne souhaitez pas que les liens déjà visités soient colorés d'une façon différente, il vous faudra leur appliquer la même couleur qu'aux liens normaux.

Résultat de l'exemple



Quelques bonnes adresses

Vous connaissez *Google* ? C'est le moteur de recherche le plus utilisé au monde !

Vous connaissez *le W3C* ? Ce sont les personnes qui définissent HTML et CSS.

Vous connaissez *OpenClassrooms* ? Ah ben oui, quelle question stupide...