

HTML

Table des matières

1. Le fonctionnement des sites web.....	2
2. HTML et CSS : deux langages pour créer un site web.....	3
2.1. Les rôles de HTML et CSS.....	3
2.2. Les différentes versions de HTML et CSS.....	3
3. Les navigateurs.....	4
4. Les balises et leurs attributs.....	5
5. Structure de base d'une page HTML5.....	6
5.1. Le doctype.....	7
5.2. La balise <html>.....	7
5.3. L'en-tête <head> et le corps <body>.....	7
5.4. L'encodage (charset).....	7
5.5. Le titre principal de la page.....	7
5.6. Les commentaires.....	8
6. Organiser son document.....	8
6.1. Les paragraphes.....	8
6.2. Sauter une ligne.....	9
6.3. Les titres.....	9
6.4. La mise en valeur.....	10
7. Les listes.....	10
7.1. Liste non ordonnée.....	10
7.2. Liste ordonnée.....	11
8. Créer des liens.....	11
8.1. Un lien vers un site.....	11
8.2. Un lien vers une page locale.....	12
8.3. Un lien vers une ancre.....	12
8.4. Lien vers une ancre située dans une autre page.....	13
8.5. Un lien pour envoyer un e-mail.....	13
8.6. Un lien pour télécharger un fichier.....	13
9. Les images.....	13
9.1. Les différents formats d'images.....	13
9.2. Insérer une image.....	13
9.3. Les figures.....	14



1. Le fonctionnement des sites web

HTML, de CSS sont des **langages informatiques** qui permettent de créer des sites web. Tous les sites web sont basés sur ces langages. Ils sont à la base même du Web. Le langage HTML a été inventé par Tim Berners-Lee et Robert Caillau en 1991 au CERN¹.

Langages
HTML et CSS

Traduction par
l'ordinateur



Résultat visible
à l'écran



Tim Berners-Lee a créé le *World Wide Web Consortium* (W3C) qui définit les nouvelles versions des langages liés au Web.

De nombreuses personnes confondent Internet et le Web. Le Web *fait partie* d'Internet.

Internet est un réseau mondial d'ordinateurs interconnectés entre eux.

Tim Berners-Lee n'est donc pas l'inventeur d'Internet, c'est « seulement » l'inventeur du Web.

¹ Centre Euporéen de la Recherche Nucléaire

2. 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), il faut aussi indiquer où placer ce texte, insérer des images, faire des liens entre les pages, etc.

2.1. Les rôles de HTML et CSS

Pour expliquer à l'ordinateur ce que vous voulez faire, il va falloir utiliser un langage qu'il comprend.

Il y a deux langages qui *se complètent* car ils ont des rôles différents :

- **HTML** (*HyperText Markup Language*) : il a fait son apparition dès 1991 lors du lancement du Web. Son rôle est de gérer et organiser le contenu. C'est donc en HTML que vous écrirez ce qui doit être affiché sur la page : du texte, des liens, des images...
- **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.

HTML



HTML + CSS



- Le HTML définit le contenu.
- Le CSS permet de définir la présentation : couleurs, image de fond, marges, taille du texte...

2.2. Les différentes versions de HTML et CSS

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

Voici un très bref historique de ces langages.



- **HTML 1** : c'est la toute première version créée par Tim Berners-Lee en 1991.
- **HTML 2** : la deuxième version du HTML apparaît en 1994 et prend fin en 1996 avec

l'apparition du HTML 3.0. C'est cette version qui posera en fait les bases des versions suivantes du HTML. Les règles et le fonctionnement de cette version sont donnés par le W3C².

- **HTML 3** : apparue en 1996, cette nouvelle version du HTML rajoute de nombreuses possibilités au langage comme les tableaux, les applets, les scripts, le positionnement du texte autour des images, etc.
- **HTML 4** : il s'agit de la version la plus répandue du HTML. Elle apparaît pour la première fois en 1998 et propose l'utilisation de frames (qui découpent une page web en plusieurs parties), des tableaux plus complexes, des améliorations sur les formulaires, etc. Mais surtout, cette version permet pour la première fois d'exploiter des feuilles de style CSS !
- **HTML 5** : c'est LA dernière version. Encore assez peu répandue, elle fait beaucoup parler d'elle car elle apporte de nombreuses améliorations comme la possibilité d'inclure facilement des vidéos, un meilleur agencement du contenu, de nouvelles fonctionnalités pour les formulaires, etc.




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




3. Les navigateurs

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

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

Navigateur	OS	Téléchargement	Commentaires
Google Chrome 	Windows Mac Linux	Téléchargement	Le navigateur de Google, simple d'emploi et très rapide.
Mozilla Firefox 	Windows Mac Linux	Téléchargement	Le navigateur de la fondation Mozilla, célèbre et réputé.

² World Wide Web Consortium

Navigateur	OS	Téléchargement	Commentaires
Internet Explorer 	Windows	Téléchargement (Déjà installé sur Windows)	Le navigateur de Microsoft, qui équipe tous les PC Windows.
Safari 	Windows Mac	Téléchargement (Déjà installé sur Mac OS X)	Le navigateur d'Apple, qui équipe tous les Mac.
Opera 	Windows Mac Linux	Téléchargement	Il est moins utilisé mais propose de nombreuses fonctionnalités.

La plupart des navigateurs sur smartphones sont les mêmes que sur ordinateur, dans une version plus légère adaptée aux mobiles. Tout dépend du type de téléphone.

- **iPhone** : sur l'iPhone d'Apple, le navigateur utilisé est Safari Mobile. Il s'agit d'une version *light* et néanmoins très complète de Safari pour ordinateur.
- **Android** : les portables sous Android bénéficient du navigateur Chrome Mobile. Là encore, il s'agit d'une version adaptée aux mobiles.
- **Windows Phone** : sous Windows Phone, on retrouve... Internet Explorer Mobile ! Le principe est le même que pour les précédents navigateurs : il s'agit d'une version dédiée aux mobiles.

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

4. Les balises et leurs attributs

Les pages HTML sont constituées de ce qu'on appelle des balises. Celles-ci sont invisibles à l'écran pour les internautes, 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>

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 :

1. Les balises en paires

Elles s'ouvrent, contiennent du texte, et se ferment plus loin. Voici à quoi elles ressemblent :
<title>Ceci est un titre</title>

On distingue une balise ouvrante (<title>) et une balise fermante (</title>) 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.

2. Les balises orphelines

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

Une balise orpheline s'écrit comme ceci : ``

Les attributs viennent compléter les balises 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 `` que nous venons de voir. Seule, elle ne sert pas à grand chose. On pourrait rajouter un attribut qui indique le nom de l'image à afficher : ``

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

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

```
<cite author="Neil Armstrong" date="21/07/1969">  
    C'est un petit pas pour l'homme, mais un bond de géant pour l'humanité.  
</cite>
```

5. Structure de base d'une page HTML5

Le code ci-dessous 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>
```

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.*

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.

Toutes ces balises ont un rôle précis.

5.1. Le doctype

```
<!DOCTYPE html>
```

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

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

5.2. La balise <html>

C'est la balise principale du code. Elle englobe tout le contenu de votre page et la balise fermante </html> se trouve tout à la fin du code !

5.3. 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.

5.4. L'encodage (charset)

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

Cette balise indique l'encodage utilisé dans le fichier .html qui détermine comment les caractères spéciaux vont s'afficher (accents, idéogrammes chinois et japonais, caractères arabes, etc.)³.

Il y a plusieurs techniques d'encodage portant des noms bizarres et utilisées en fonction des langues : ISO-8859-1, OEM 775, Windows-1253... Une seule cependant devrait être utilisée aujourd'hui autant que possible : UTF-8. Cette méthode d'encodage permet d'afficher sans aucun problème pratiquement tous les symboles de toutes les langues.

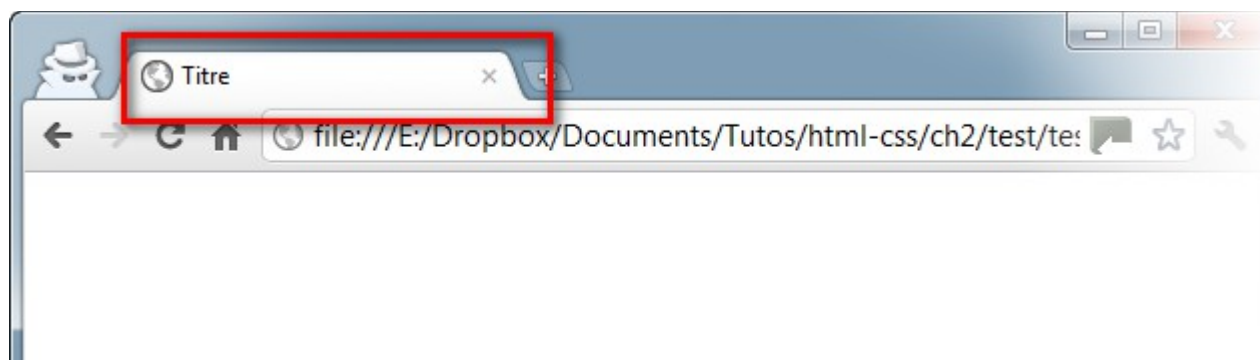
5.5. Le titre principal de la page

```
<title>
```

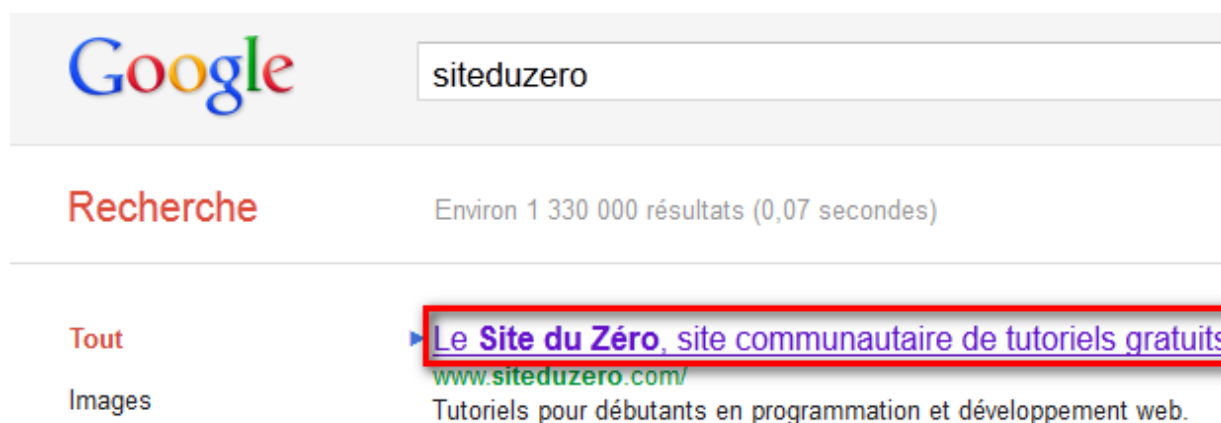
C'est le titre de votre page, probablement l'élément le plus important ! Toute page doit avoir un titre qui décrit ce qu'elle contient. Il est conseillé de garder le titre assez court (moins de 100 caractères en général).

3 Voir cours numération - codage

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



5.6. 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.

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

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

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

6. Organiser son document

6.1. Les paragraphes

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

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

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

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

Le contenu du site web entre les balises `<body></body>`. Il nous suffit donc de mettre notre paragraphe entre ces deux balises.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Paragraphe</title>
  </head>
  <body>
    <p>Bonjour et bienvenue sur mon site !</p>
  </body>
</html>
```

6.2. Sauter une ligne

Pour écrire un deuxième paragraphe, il suffit d'utiliser une deuxième balise `<p>`.

Exemple :

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Paragraphe</title>
  </head>
  <body>
    <p>Bonjour et bienvenue sur mon site !</p>
    <p>Voici une deuxième ligne de texte.</p>
  </body>
</html>
```

Pour aller à la ligne dans un paragraphe sans sauter une ligne il existe une balise « Aller à la ligne ». C'est une balise **orpheline** qui sert juste à indiquer qu'on doit aller à la ligne : `
`. Vous devez obligatoirement la mettre à l'intérieur d'un paragraphe.

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

6.3. Les titres

En HTML, il est possible d'utiliser six niveaux de titres différents :

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

- `<h5>` `</h5>` : titre pas important.
- `<h6>` `</h6>` : titre vraiment, mais alors là vraiment pas important du tout.

Attention : ne pas confondre 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 servent à créer des titres qui seront affichés *dans* la page web.

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

Pour modifier la taille ou la couleur du texte il faudra utiliser le CSS.

6.4. La mise en valeur

- Pour mettre *un peu* en valeur un texte : balise `` ``
- Pour mettre un texte bien en valeur : balise `` `` qui signifie « important ».
- Pour marquer un texte : balise `<mark>` `</mark>` permet de faire ressortir visuellement une portion de texte. L'extrait n'est pas forcément considéré comme important mais on veut qu'il se distingue bien du reste du texte. Cela peut être utile pour faire ressortir un texte pertinent après une recherche sur votre site par exemple.

7. Les listes

Les listes permettent d'ordonner nos informations. Il existe deux types de listes :

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

7.1. Liste non ordonnée

Une liste non ordonnée ressemble à ceci :

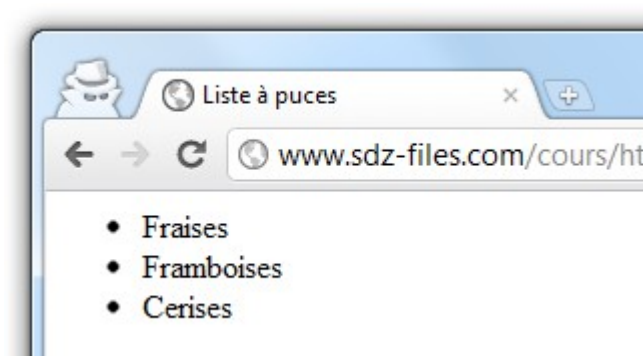
- Fraises
- Framboises
- Cerises

Pour créer une liste non ordonnée, il suffit d'utiliser deux balises :

- ```` délimite toute la liste ;
- ```` délimite un élément de la liste (une puce).

Exemple :

```
<ul>
  <li>Fraises</li>
  <li>Framboises</li>
  <li>Cerises</li>
</ul>
```



7.2. Liste ordonnée

Une liste ordonnée fonctionne de la même façon, seule une balise change : il faut remplacer `` par ``.

Exemple :

```
<h1>Ma journée</h1>
<ol>
  <li>Je me lève.</li>
  <li>Je mange et je bois.</li>
  <li>Je retourne me coucher.</li>
</ol>
```



8. Créer des liens

8.1. Un lien vers un 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.

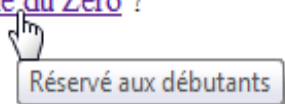
Pour faire un lien, il faut utiliser la balise `<a>` et lui ajouter un attribut, `href`, pour indiquer vers quelle page le lien doit conduire.

Exemple : `Site du Zéro`

Les liens que nous venons de voir sont appelés **liens absolus** car on indique l'adresse complète.

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.

Souhaitez-vous visiter le [Site du Zéro](http://www.siteduzero.com) ?

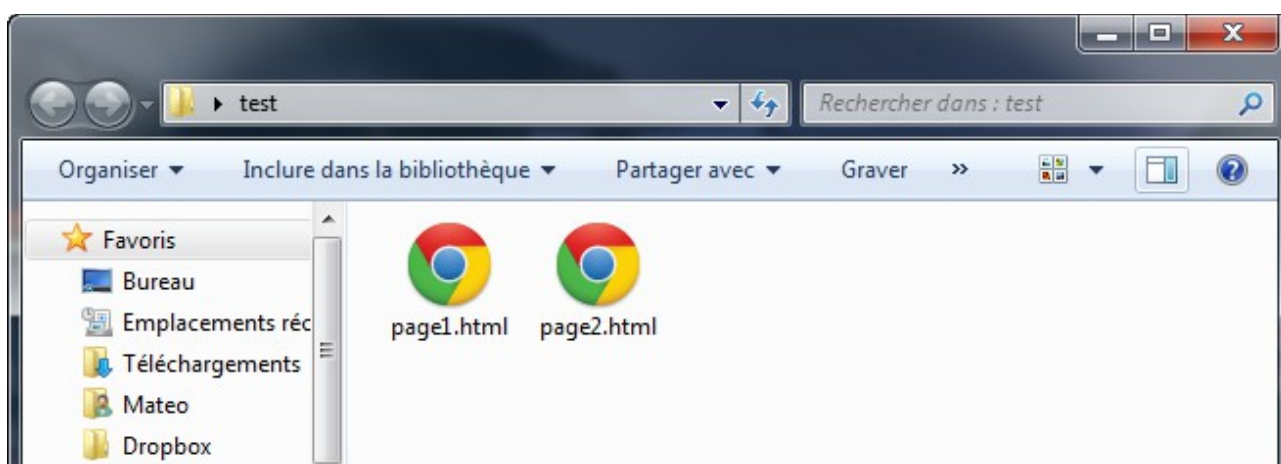


`<p>Bonjour. Souhaitez-vous visiter le Site du Zéro ?</p>`

8.2. Un lien vers une page locale

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 : ``

C'est un **lien relatif**.



Si les pages sont situées dans des dossiers différents, le lien doit être rédigé en indiquant le chemin relatif : ``

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

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

8.3. 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`

8.4. 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. Il suffit de taper le nom de la page, suivi d'un dièse (#), suivi du nom de l'ancre.

Par exemple : ``

8.5. 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` en modifiant la valeur de l'attribut `href` comme ceci :

`<p>Envoyez-moi un e-mail !</p>`

8.6. Un lien pour télécharger un fichier

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.

Exemple : `<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.

9. Les images

9.1. Les différents formats d'images

Certains formats sont plus adaptés que d'autres selon l'image (photo, dessin, image animée...).

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.

Voici quel format à adopter en fonction de l'image :

- **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.

9.2. Insérer une image

La balise qui va nous permettre d'insérer une image est ``

Cette balise de type **orpheline** 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) qui ne peuvent malheureusement pas « voir » l'image. Cela aide aussi les robots des moteurs de recherche pour les recherches d'images. Pour la fleur, on mettrait par exemple : `alt="Une fleur"`.
- **Title** : permet d'afficher une bulle d'aide. Cet attribut est facultatif.

Les images doivent se trouver obligatoirement à l'intérieur d'un paragraphe (`<p></p>`) ou d'une figure (`<figure></figure>`).

9.3. Les figures

Ce sont des éléments qui viennent enrichir le texte pour compléter les informations de la page. Les figures ne sont pas *forcément* des images : un code source aussi illustre le texte.

Les figures peuvent être de différents types :

- images ;
- citations ;
- etc.

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

```
<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>Logos des différents navigateurs</figcaption>
</figure>
```

- Si l'image n'apporte aucune information : placez l'image dans un paragraphe.
- Si elle apporte une information : placez l'image dans une figure.

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

QUIZZ

1. Quel est le rôle du navigateur web ?

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

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

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

3. Lequel de ces langages permet de dire *"Mon menu de navigation est placé en haut à droite"* ?

- HTML
- CSS
- JavaScript
- PHP

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

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

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

- ``
- `<p>`
- ``
- ``

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

- PNG
- JPEG
- GIF
- BMP

7. Pourquoi l'attribut `alt` est-il obligatoire pour les images ?

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

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

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

9. Quelle balise délimite une liste à puce numérotée ?

- `<dl>`
- ``
- ``
- ``

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

- ``
- ``
- ``
- ``