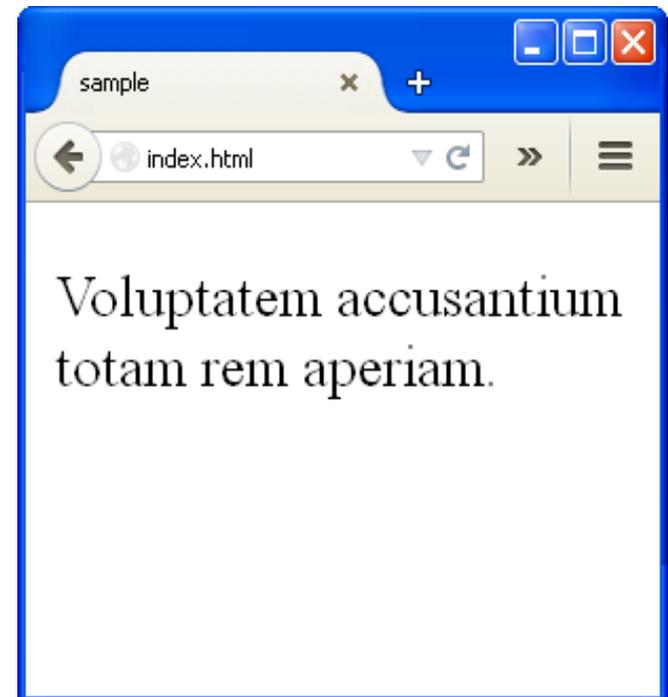


UN PREMIER PAS DANS LES LANGAGES HTML ET CSS

```
<!DOCTYPE html>
<html>
<!-- created 2010-01-01 -->
<head>
  <title>sample</title>
</head>
<body>
  <p>Voluptatem accusantium
  totam rem aperiam.</p>
</body>
</html>
```

HTML



OBJECTIFS

- ➔ Comprendre les principes de base et la structure d'un document HTML
- ➔ Savoir mettre en forme du texte (gras, italique, etc...)
- ➔ Savoir afficher des images
- ➔ Savoir construire et mettre des objets (texte, images, etc...) dans un tableau
- ➔ Savoir créer des liens vers d'autres documents

SITES DE RÉFÉRENCE

➔ <https://www.w3schools.com/>

➔ <https://developer.mozilla.org/fr/docs/Web/HTML>

➔ [Openclassroom - Première page web](https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/1604361-creez-votre-premiere-page-web-en-html)

<https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/1604361-creez-votre-premiere-page-web-en-html>

➔ [Livre-HTML-CSS.pdf](http://lewebpedagogique.com/langemelanie/files/2014/05/Livre-HTML-CSS.pdf)

<http://lewebpedagogique.com/langemelanie/files/2014/05/Livre-HTML-CSS.pdf>

I.

Quelques termes
importants

1.1 Internet

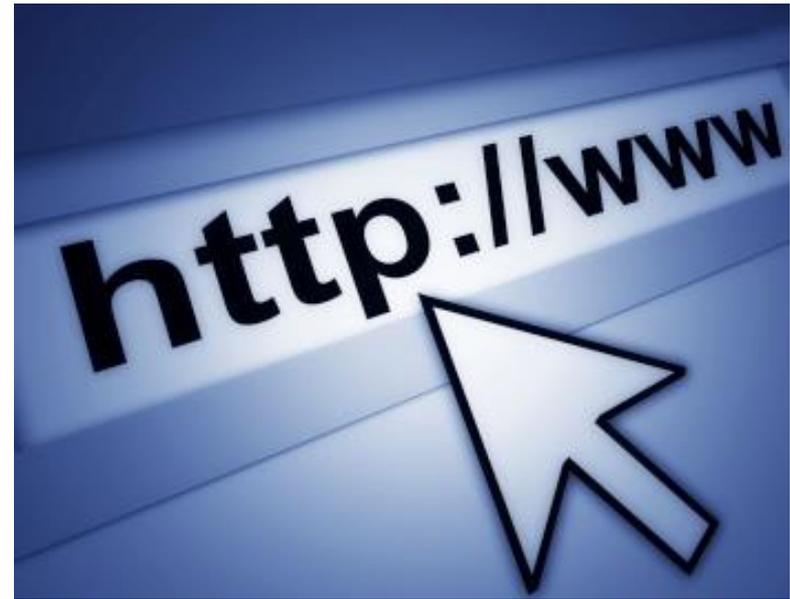
➔ Réseau informatique mondial physique accessible au public



➔ Composé des réseaux publics, privés, universitaires, commerciaux, ...

1.2 www - world wide web

- WWW :
- "le web"
 - " la toile d'araignée mondiale"



- ↳ Ensemble des informations contenues sur les machines du réseau et la manière de communiquer entre elles.
- ↳ Le web est une application du réseau internet parmi d'autres (mail, messagerie instantanée, partage de fichiers P2P, streaming vidéo, etc.).

1.3 HTML- HyperText Markup Language

```
<!DOCTYPE html>
<html>
<!-- created 2010-01-01 -->
<head>
  <title>sample</title>
</head>
<body>
  <p>Voluptatem accusantium
  totam rem aperiam.</p>
</body>
</html>
```

HTML



- ➔ Langage de balisage de "fichiers hypertextes"
 - ↳ (pages web)
- ➔ Utilisé pour décrire des pages web
 - ↳ Langage de description ≠ langage de programmation !!

1.4 HTML- HyperText Markup Language

➔ On visualise les fichiers HTML avec des **navigateurs web** :



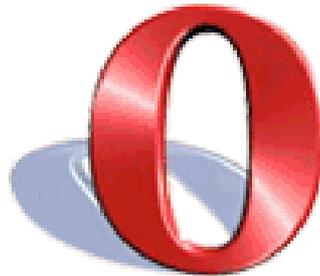
Internet Explorer



Firefox



Chrome



Opera

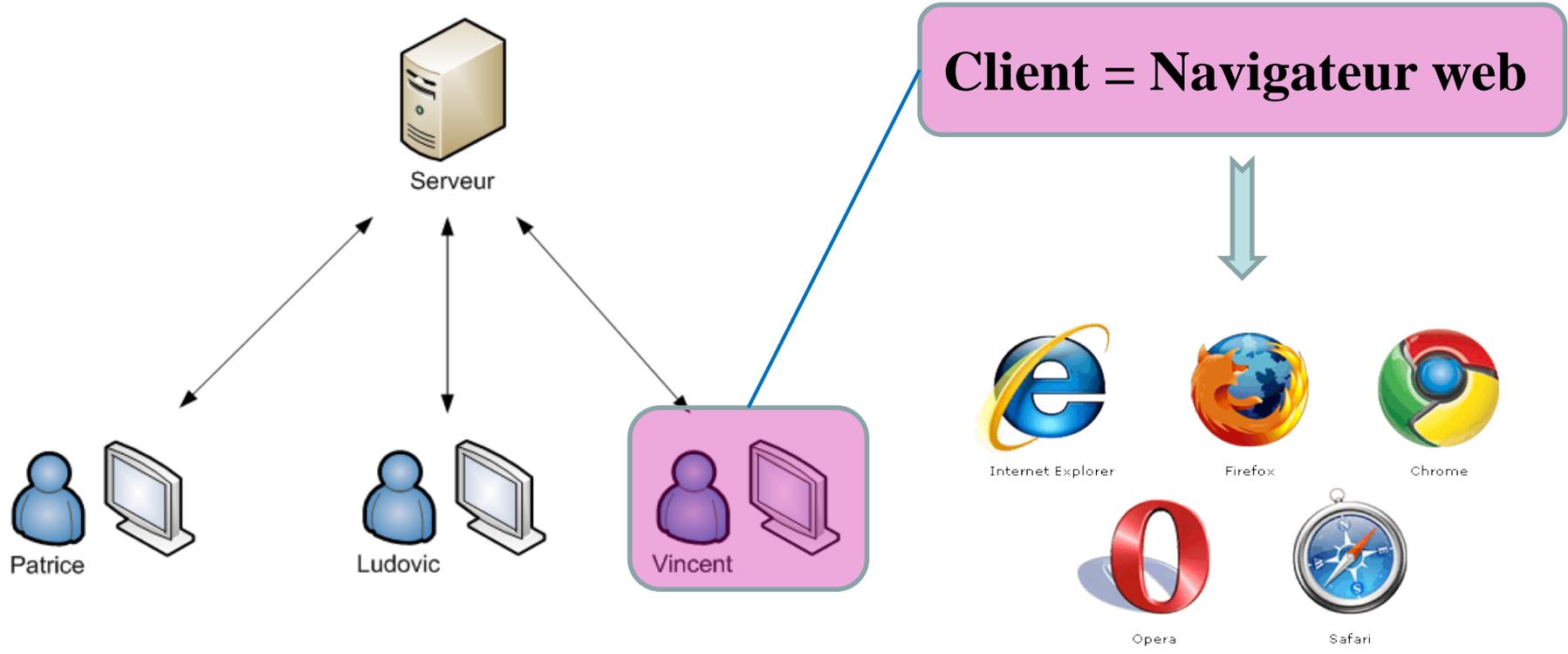


Safari

1.5 Principe de fonctionnement du Web : Client/Serveur

➔ Un site web classique fonctionne sur un système

Client-Serveur (voir Wikipedia: <https://fr.wikipedia.org/wiki/Client%E2%80%93serveur>)



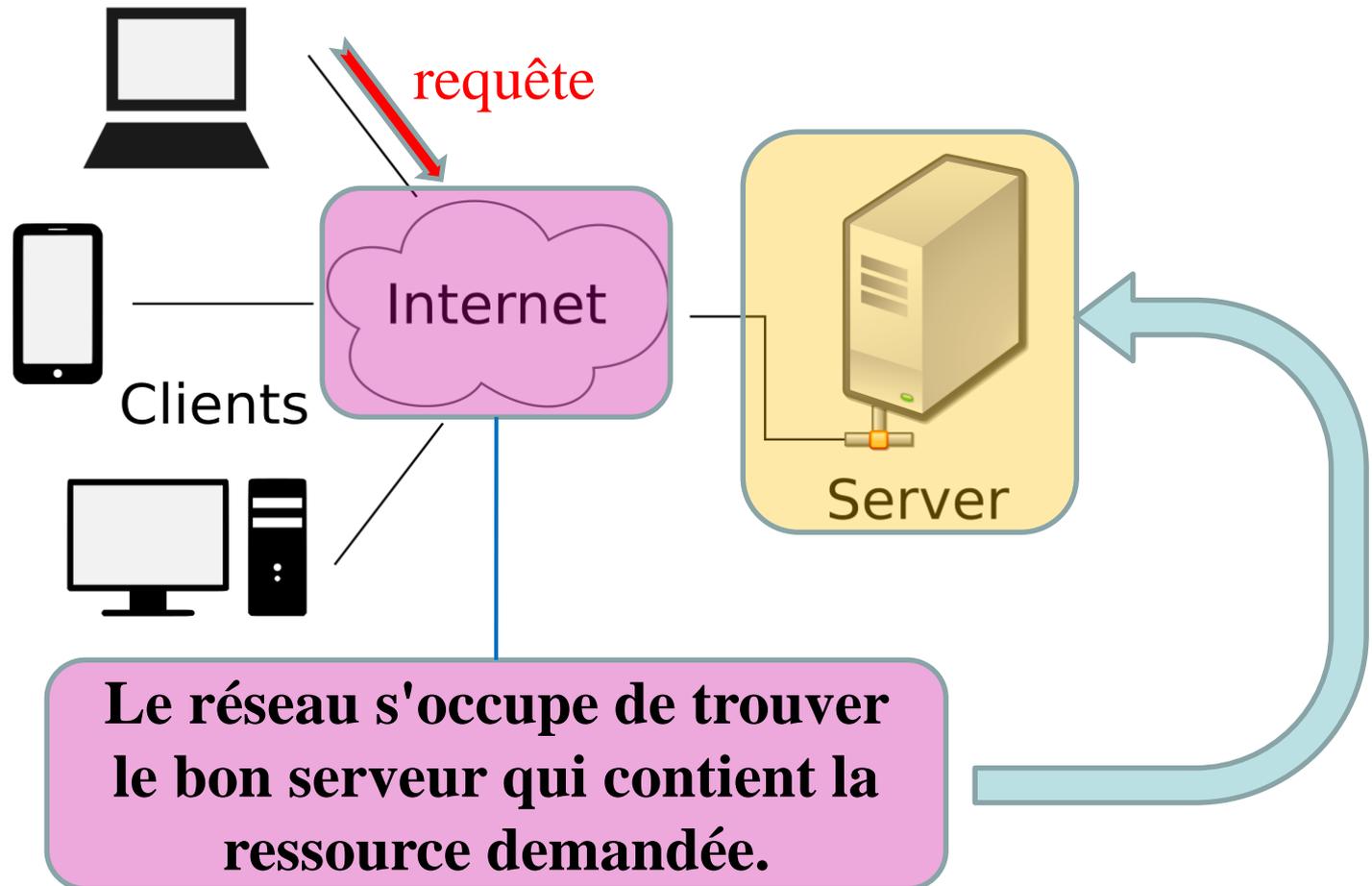
Client = Navigateur web

Un lien identifie de façon unique une ressource distante sur le réseau web (une application ou page web hébergée sur un serveur web).

Programme qui permet d'interroger un serveur par un **lien ou URL.**

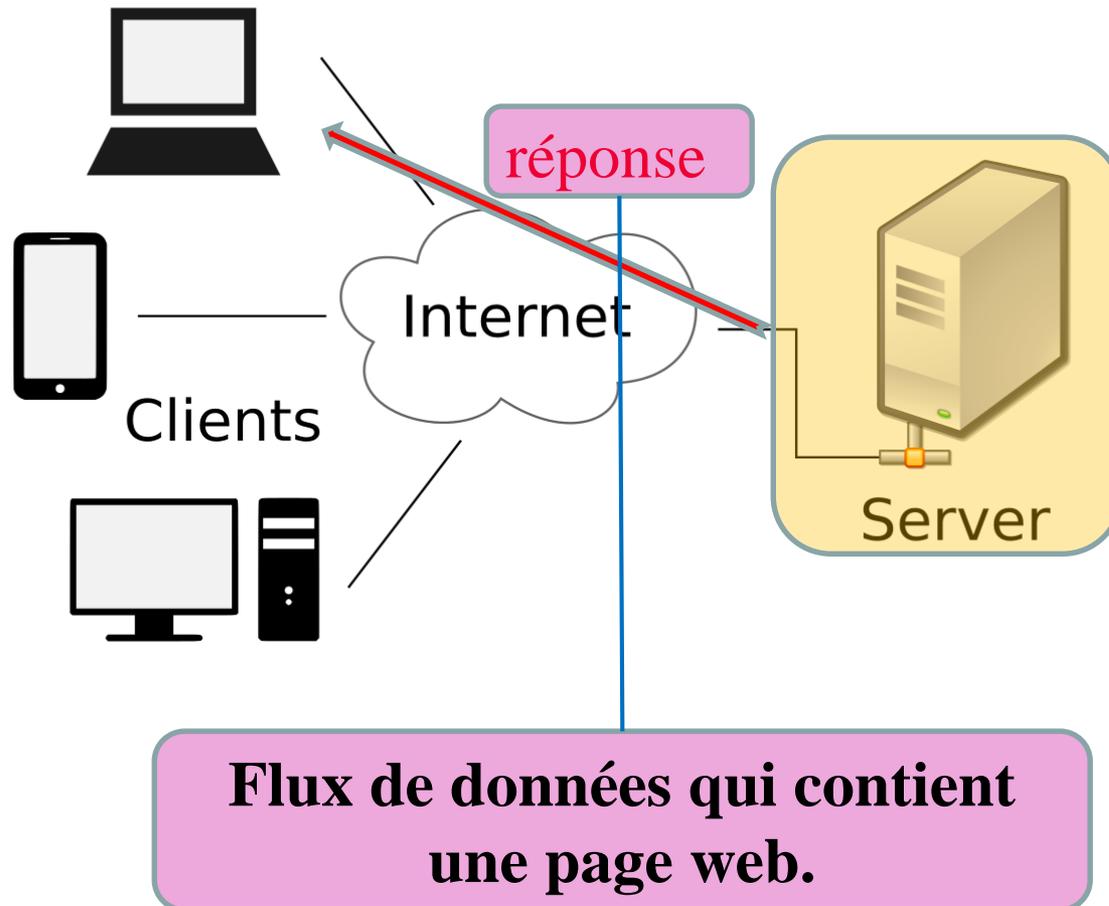
Requête

Lorsque qu'un **lien ou URL** est saisi dans le navigateur ou lorsque qu'un lien est cliqué à partir d'une page, le **navigateur interroge le réseau** par une **requête**.



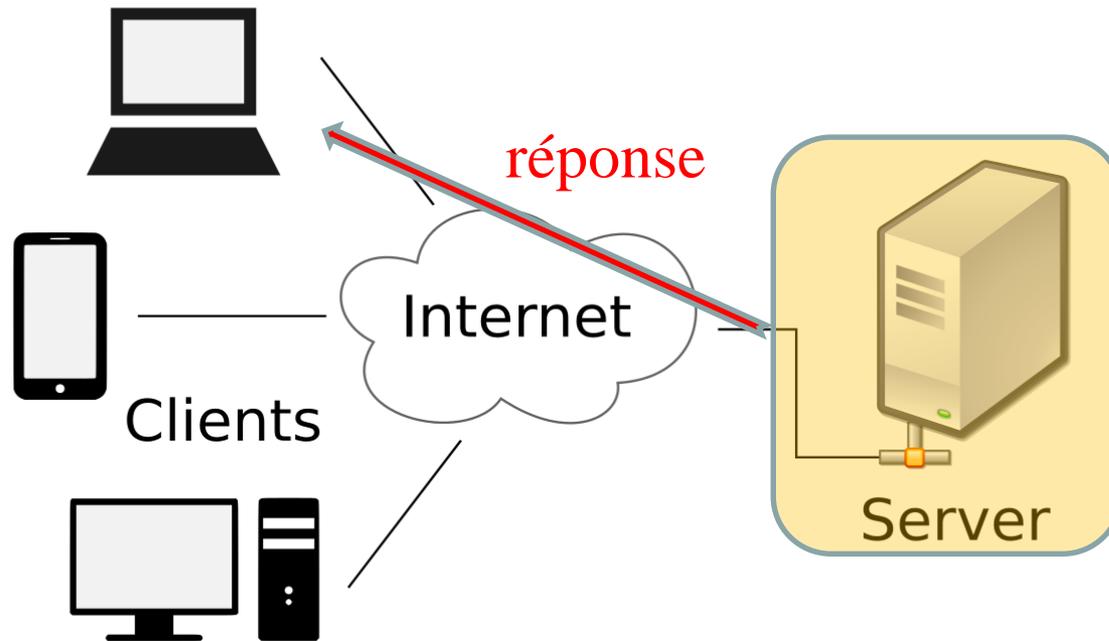
Réponse

Le serveur interrogé construit une réponse et l'envoie au client (vers le navigateur de votre ordinateur).



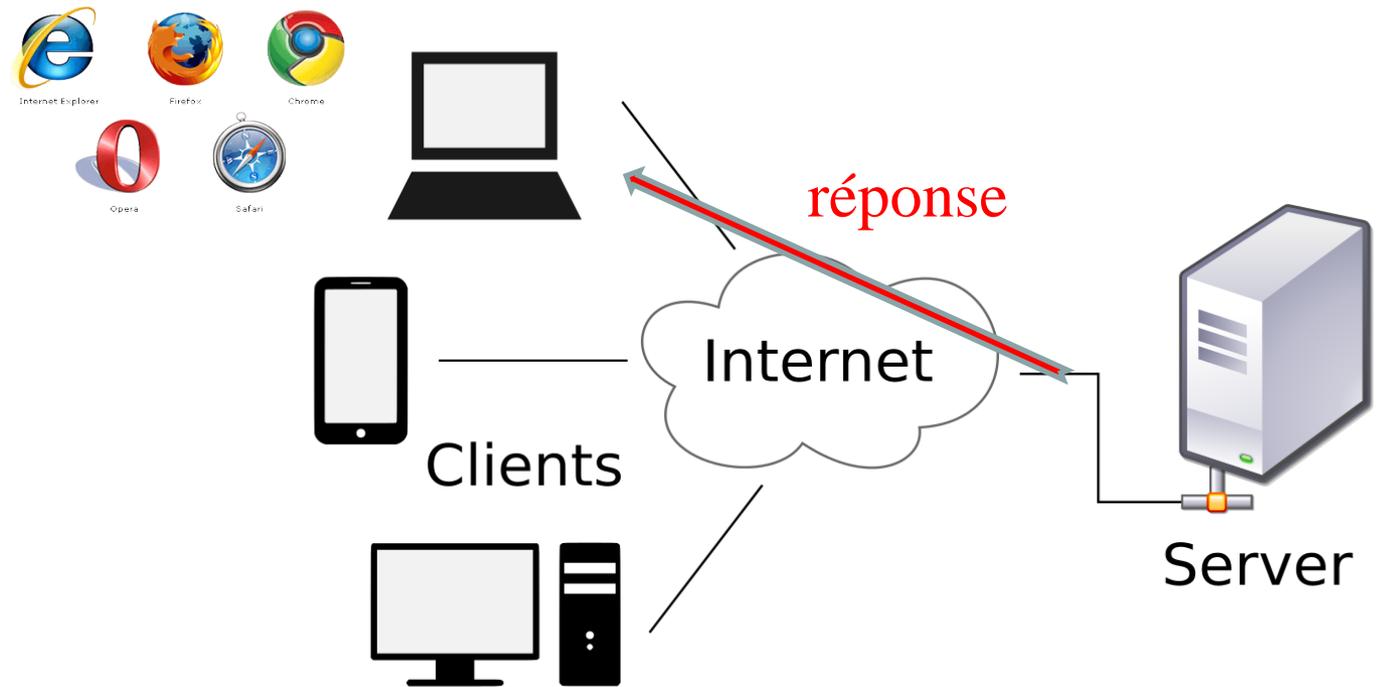
Réponse

Le message échangé entre le client et le serveur transite sur le réseau en utilisant un **protocole HTTP** ou **HTTPS** ('s' pour sécurisé).

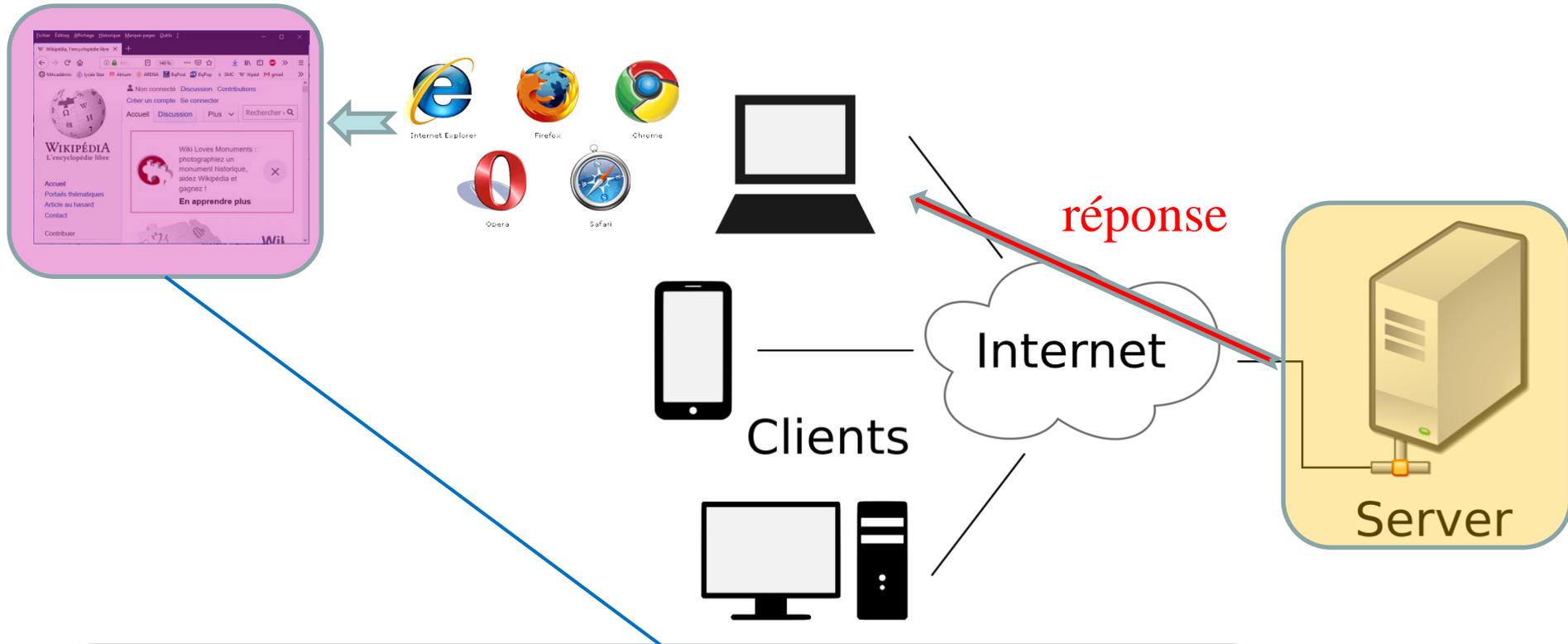


Réponse

La page web renvoyée peut être **interprété par le navigateur** qui reçoit la réponse.



Réponse Le navigateur affiche la page



Contient :

- des balises HTML pour structurer l'information
- du CSS pour la mise en page et le style
- du JavaScript pour exécuter des petits traitements "coté client" (à l'opposé des traitements exécutés "coté serveur" lors de la préparation de la page).

II.

Présentation du

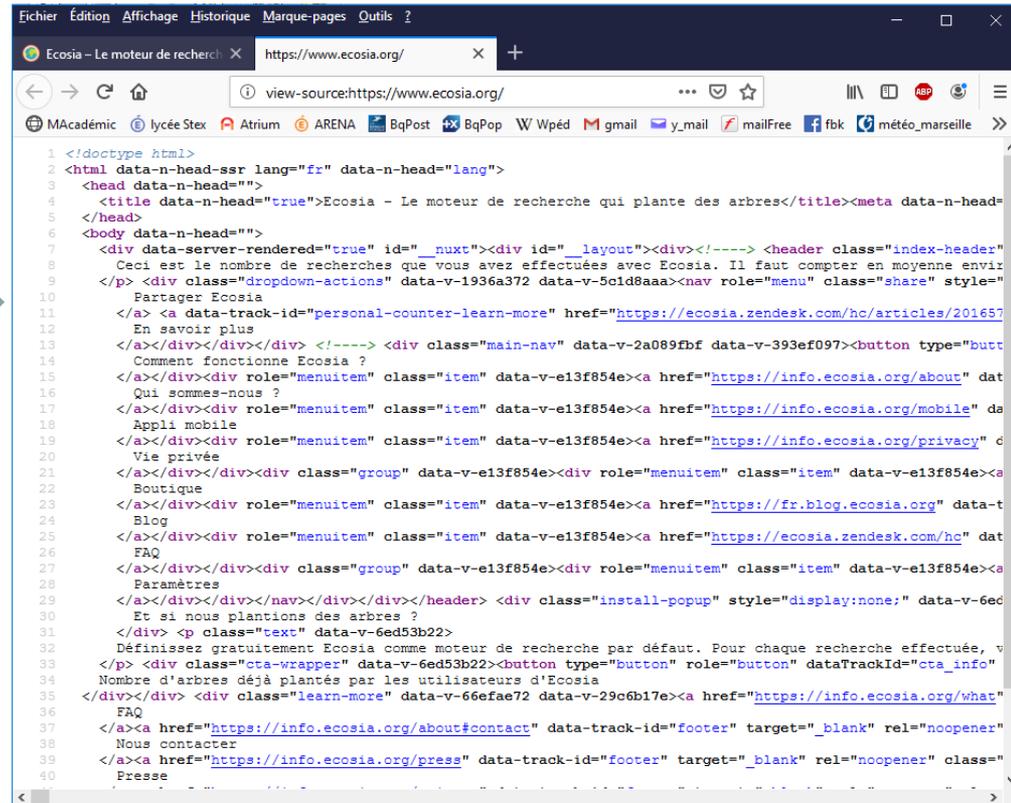
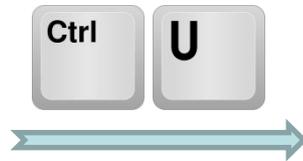
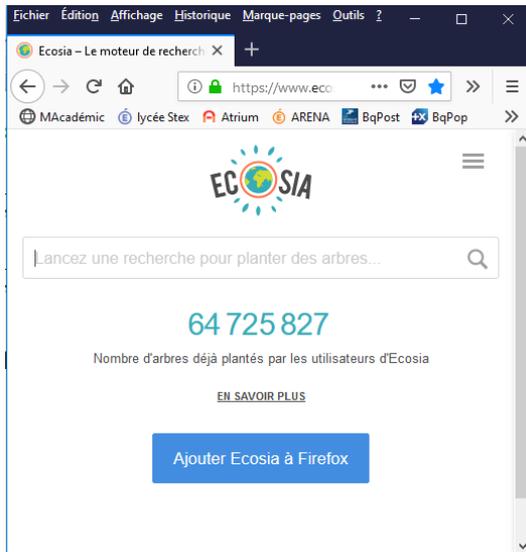
HTML

2.1 Pourquoi apprendre le HTML ?

➔ Langage de représentation universel et unique.

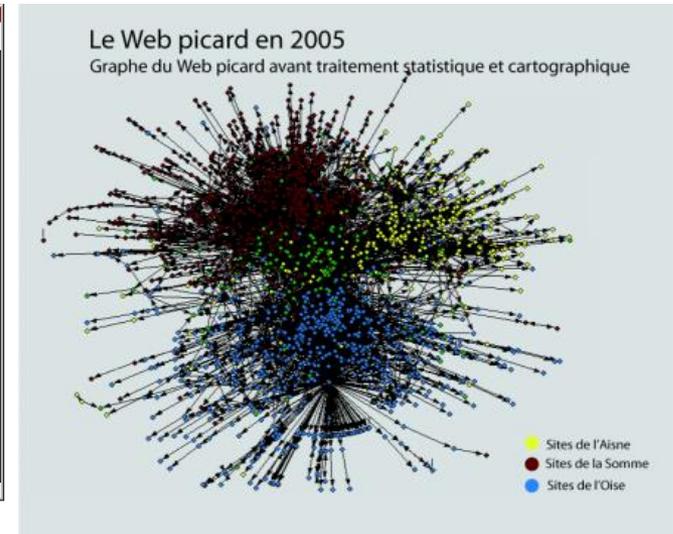
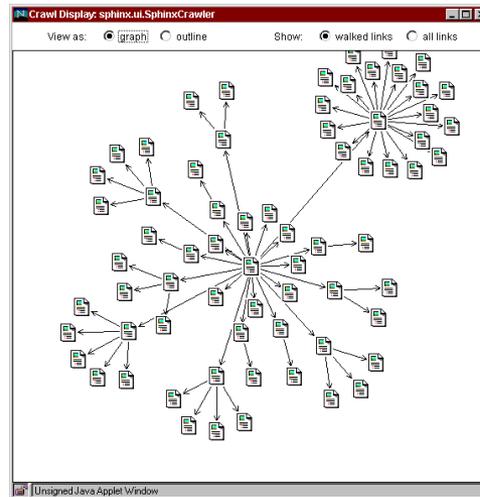
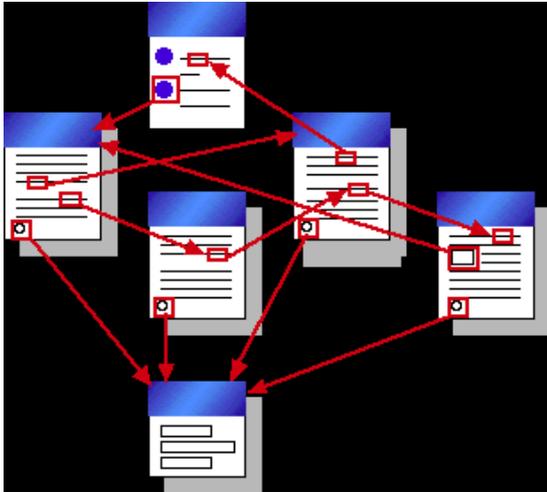
➔ Langage de représentation Transparent.

➔ Exemple : <https://www.ecosia.org/>



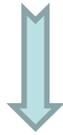
2.2 Qu'est-ce qu'un document HyperText ?

- ➔ Les pages HTML sont du texte "enrichi".
- ➔ Les liens hypertexte permettent de passer d'une page à une autre.



2.3 Un exemple de page HTML

```
<html>
  <head>
    <title>Le titre de la page</title>
  </head>
  <body>
    <h1>Mon premier titre</h1>
    <p>Mon premier <b>paragraphe</b></p>
  </body>
</html>
```



Page vue par un codeur :

Avec un éditeur de texte
(Notepad++, Vim, Emacs,
Eclipse, etc.)

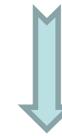
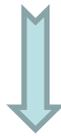
Page vue par un utilisateur:

Avec un navigateur (Firefox,
Chrome, etc.)

2.3 Un exemple de page HTML

```
<html>
  <head>
    <title>Le titre de la page</title>
  </head>
  <body>
    <h1>Mon premier titre</h1>
    <p>Mon premier <b>paragraphe</b></p>
  </body>
</html>
```

Certains éléments, comme `<html>`, `<body>`, `` etc... ne sont pas affichés sur la page web.



Ces éléments donnent des indications au navigateur sur la façon d'afficher le texte.

III.

Programmer en

HTML

3.1 Le HTML : un langage à balises

- ➔ Pour composer une page web, on écrit donc du **texte**, et on y insère des **balises** pour mettre en forme ce texte.
- ➔ Un **document HTML** est donc composé de **texte** et de **balises**.

Exemple de contenu de balise:

```
<html>
  <head>
    <title>Le titre de la page</title>
  </head>
  <body>
    <h1>Mon premier titre</h1>
    <p>Mon premier <b>paragraphe</b></p>
  </body>
</html>
```

Les balises sont :

<html>, <head>, <title>, </title>, </head>, <body>, <h1>, </h1>, <p>, , , </p>, </body>, </html>

BILAN:

→ Une balise HTML est un élément de texte (un nom) encadré par le caractère inférieur (<) et le caractère supérieur (>).

↳ Exemple : <h1> (titre 1)

→ Une balise ouverte doit (presque) toujours être refermée.

↳ Exemple : <h1> « du texte » </h1>

→ Il existe de nombreuses balises pour mettre en forme et enrichir le texte.

↳ Exemple : <h1>, <h2>, <i><u><video> , etc.

→ Le texte écrit entre la balise ouvrante et la balise fermante est le contenu de la balise.

Exemple: Mon premier paragraphe,</p></div>

 Le terme « paragraphe » est le contenu de la balise .</p></div>

Exercice: Trouver le contenu de chaque balises</p></div>

```
<html>
  <head>
    <title>Le titre de la page</title>
  </head>
  <body>
    <h1>Mon premier titre</h1>
    <p>Mon premier <b>paragraphe</b></p>
  </body>
</html>
```

Contenu de la balise <head> :
<title>Le titre de la page</title></p></div>

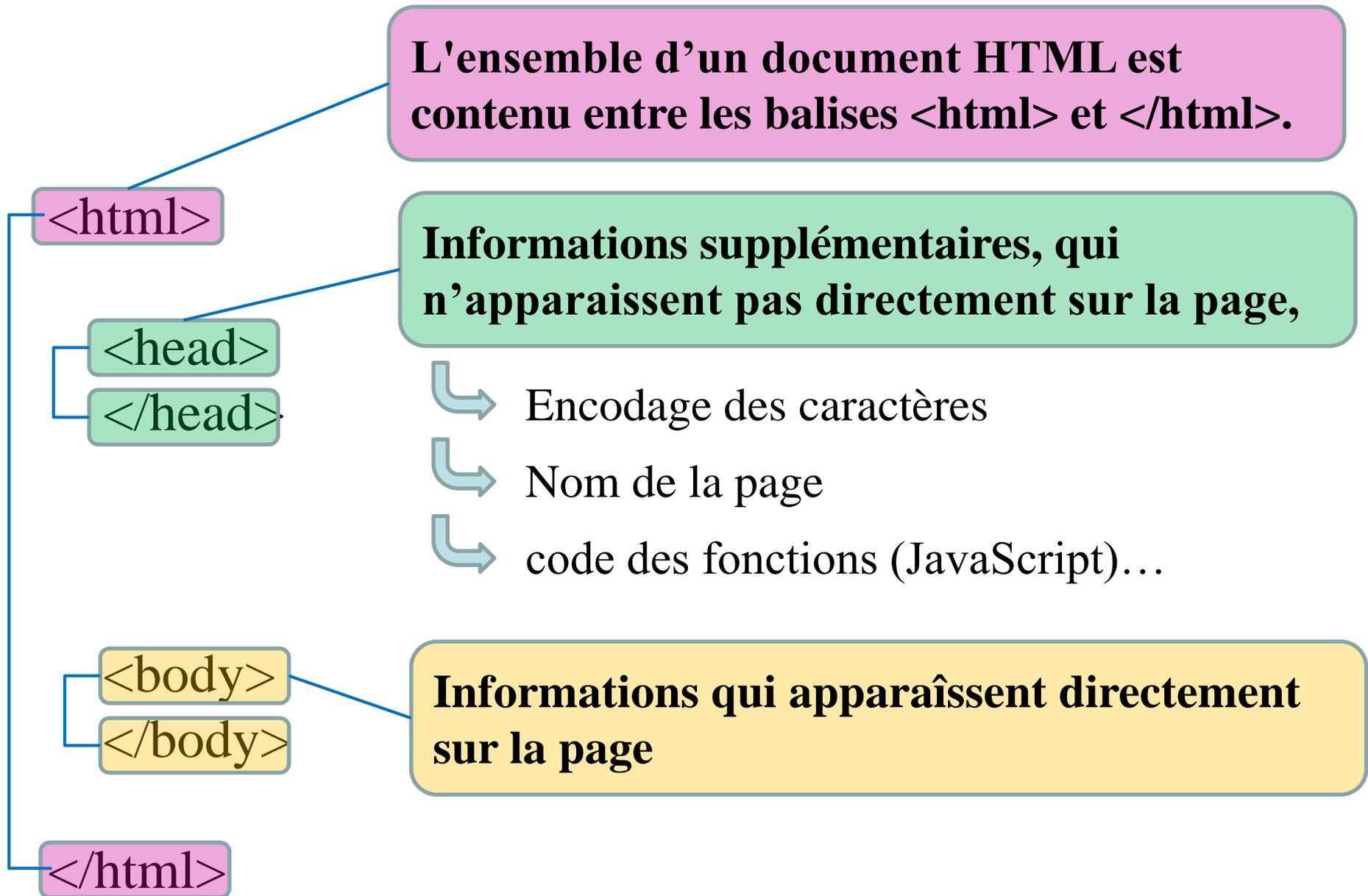
Contenu de la balise <h1> :
Mon premier titre</p></div>

Contenu de la balise <body> :

<h1>Mon premier titre</h1>,</p></div>

<p>Mon premier paragraphe</p></div>

3.2 Structure de base d'un document HTML



Les principales balises :

Correspond au texte qui sera affiché comme titre de l'onglet.

Le contenu de cette balise ne correspond pas à un titre qui apparaîtrait directement sur la page.

```
<html>
  <head>
    <title>Le titre de la page</title>
  </head>
  <body>
    <h1>Mon premier titre</h1>
    <p>Mon premier <b>paragraphe</b></p>
  </body>
</html>
```

heading 1 : titres principal , à l'intérieur de la page.

Taille de police plus grande

<h2> pour les sous-titres,
<h3> pour les titres de section,
etc...

Définit un paragraphe

- (bold) permet de mettre du texte en gras.
- <i> (pour italic) .
- (pour emphasis)

3.3 Structuration des pages

➔ Le HTML permet de structurer les pages pour mieux faire ressortir les titres, les sous-titres, les paragraphes, les listes, les citations, etc

```
<h1> Mon titre </h1>
<h2> Un sous-titre </h2>
<p> Mon premier paragraphe est
très important </p>
<p> Le deuxième paragraphe est
plus long et il contient une
liste :
  <ul>
    <li> point 1</li>
    <li> point 2</li>
  </ul>
</p>
```

Code HTML



Mon titre

Un sous-titre

Mon premier paragraphe est très important

Le deuxième paragraphe est plus long et il contient une liste :

- point 1
- point 2

Résultat sur le navigateur

3.4 Mise en forme du texte – Imbriquer les balises

- ➔ Des balises de mise en forme du texte telles que `<i>`, ``, `<u>` permettent de mettre une partie du texte en **gras**, *italique*, souligné,...

Exemple:

Une partie en `` gras `` et une en `<i>` italique `</i>`

↳ Résultat: Une partie en **gras** et une en *italique*

- ➔ Il est possible d’imbriquer les balises de mise en forme du texte.

Exemple:

Une partie en `<u><i>` italique `</i>` et tout est souligné `</u>`

↳ Résultat: Une partie en *italique* et tout est souligné

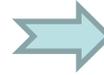
3.5 Attributs d'une balise

Exemple:

Magnifique hôtel

```
<hr width='150px' align='left' />
```

Vue sur le lac



L'attribut align='left' permet de placer le début de la ligne à gauche de la page.

L'attribut width='150px' permet de définir la largeur de la ligne à 150 pixels.

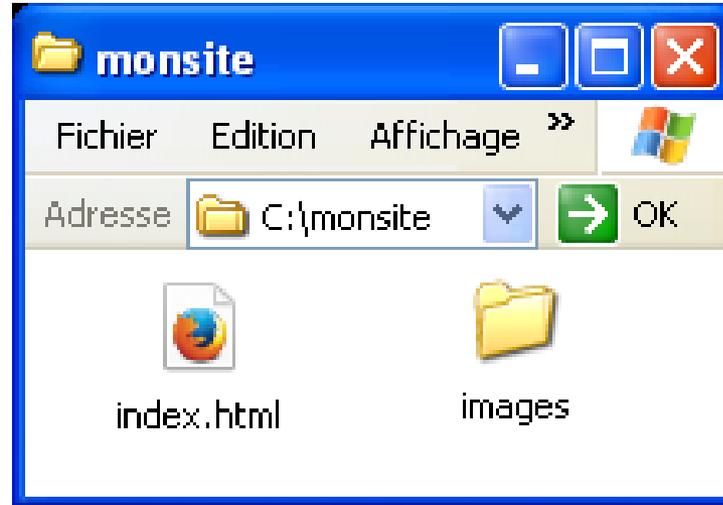
La balise <hr/> permet d'afficher des lignes horizontales

↳ Par défaut, la ligne sera affichée sur toute la largeur de la page.

Caractéristiques d'un attribut

- ➔ Un attribut a un **nom** (*width*) **suivi du signe** « = » et d'une **valeur** ('150px').
- ➔ Les attributs d'une balise s'insèrent **après le nom** de la balise et **avant le chevron de fermeture** (>).
- ➔ Les attributs ne se placent que dans la **balise d'ouverture**, jamais dans celle de fermeture.
- ➔ Quand il y a plusieurs attributs, ils sont placés **les uns à la suite des autres**, dans un **ordre quelconque** et **séparés par des espaces**.
- ➔ La **valeur** de l'attribut est entourée par des **guillemets** (width="150px") ou des **apostrophes** (width='150px').

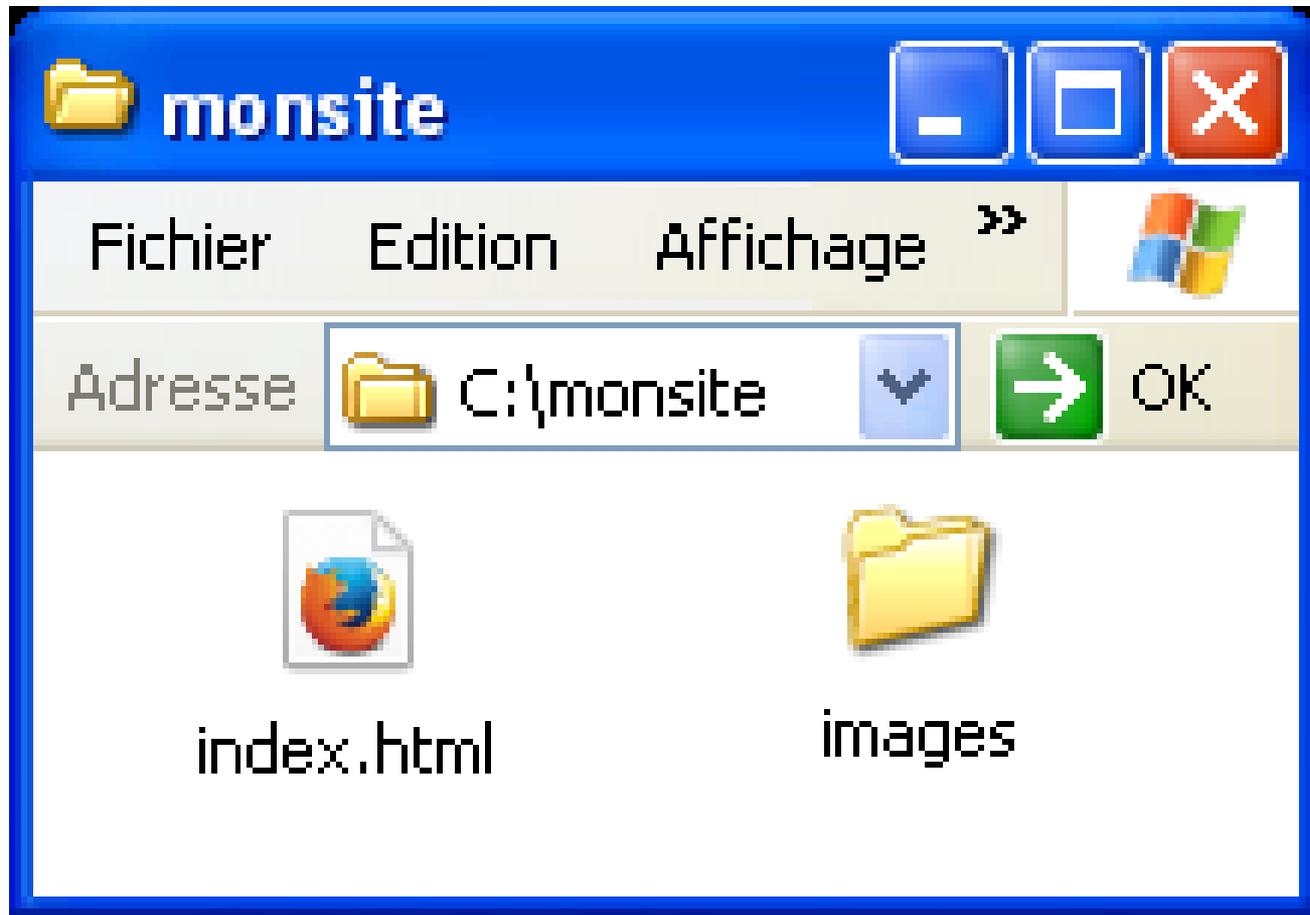
3.6 Structuration des fichiers



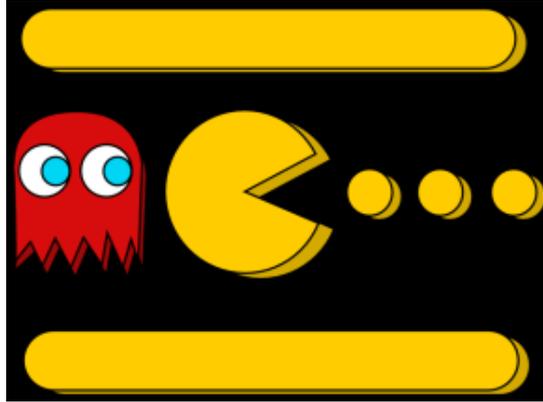
- ➔ Il est important de structurer le stockage des fichiers
- ➔ Le fichier de lancement du site s'intitule *index.html*
- ➔ Il convient de le placer dans un dossier portant le nom de votre site (par exemple *C:\monsite*).
- ➔ Pour lancer le site dans le navigateur, il suffit de double-cliquer sur le fichier *index.html*.

3.7 Insertion d'images

→ Toutes les images associées aux pages du site doivent être placées dans un dossier nommé par exemple « *images* ».



➔ Différentes manières d'insérer de l'image dans le texte :



↳ En spécifiant un chemin réduit qui, par défaut, partira du dossier courant (C:\monsite) :

```
<img src='images/pacman.jpg' alt='pacman' />
```

↳ En spécifiant l'adresse internet (URL) de l'image :

```
<img src='images/pacman.jpg' alt='pacman' />
```

↳ En spécifiant le chemin complet à partir de la racine (plus sûr !!):

```
<img src='C:/monsite/images/pacman.jpg' alt='pacman' />
```

Remarque:

``

Chemin d'accès du fichier image

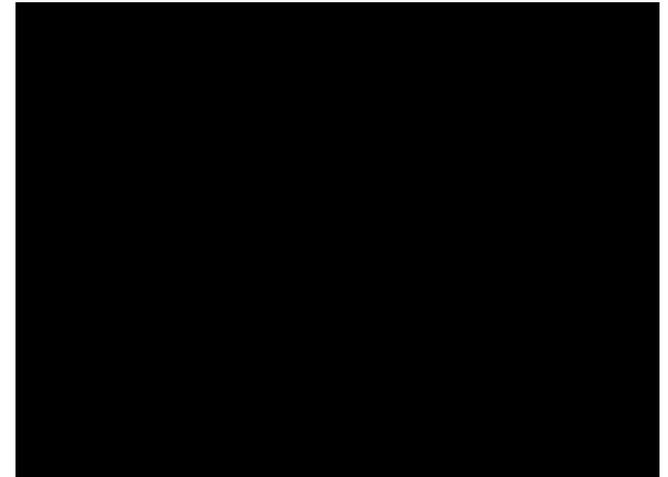
**Attribut src :
soucre**

**Balise :
insérer une image**

3.8 Insertion de vidéos (HTML5)

Lancement de plusieurs vidéos les unes à la suite des autres :

```
<video width="200" height="300" controls="controls" autoplay="true">  
  <source src="minestorm.mp4" type="video/mp4" />  
  <source src="minestorm.webm" type="video/webm" />  
  <source src="minestorm.ogv" type="video/ogg" />  
  Ecrire une alternative à la vidéo  
</video>
```

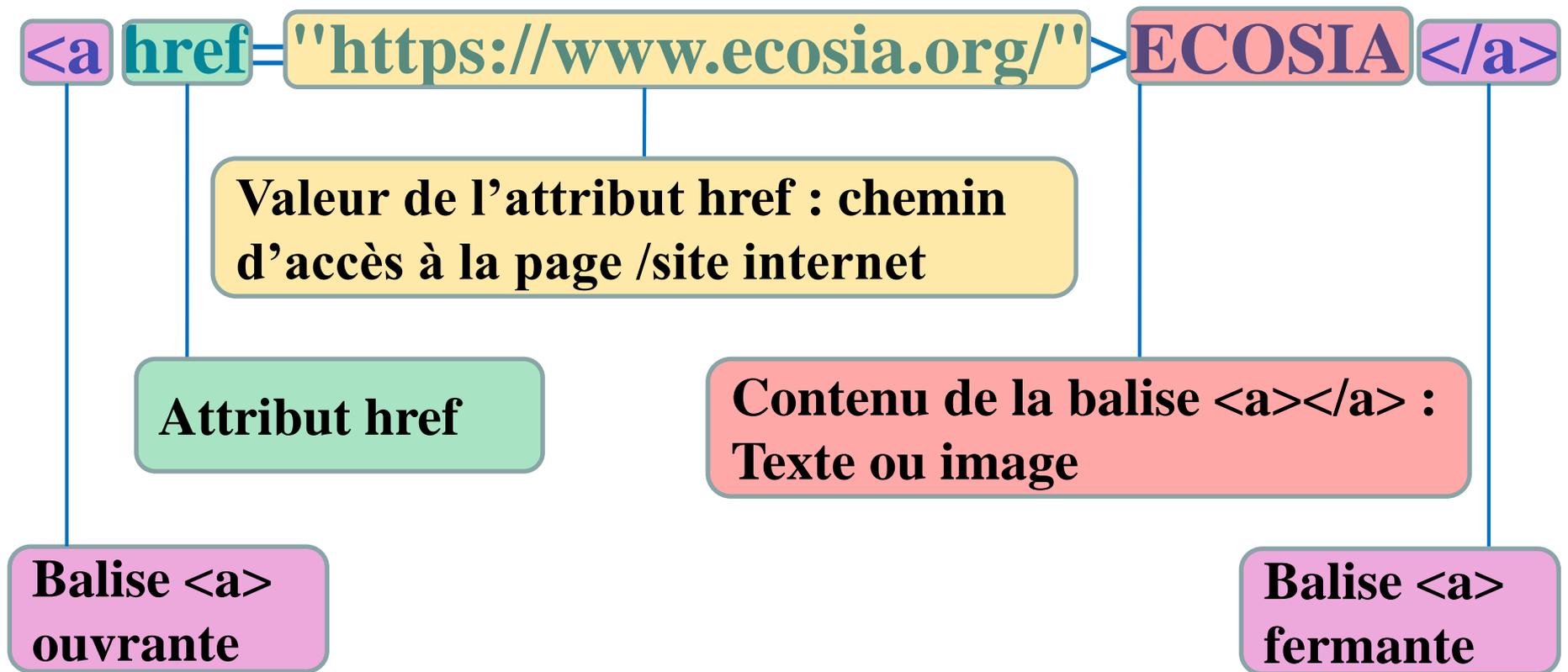


Exercice: trouver les différentes balises, leur contenu et les attributs correspondants

3.9 Insertion de liens : sans liens, pas de web !!

→ La balise `<a>`, avec son attribut **href** permet de créer un hyperlien vers une autre page (locale ou internet).

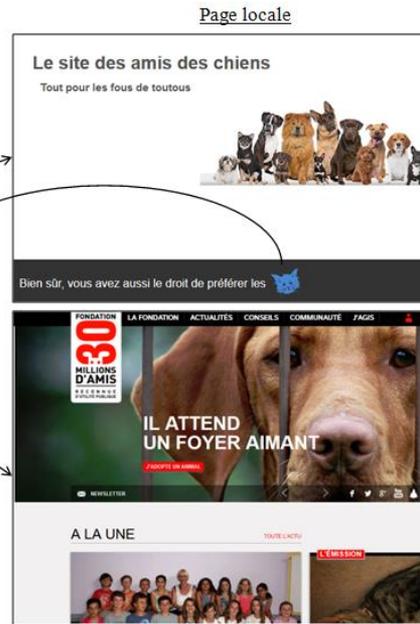
Exemple 1:



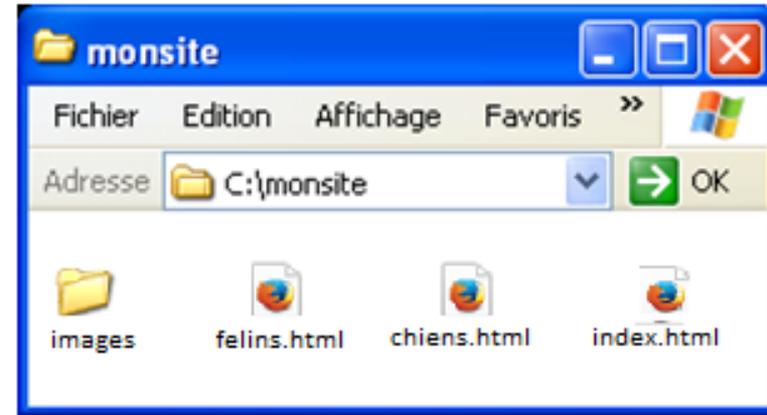
Exemple 2:



Page locale



Page web



felins.html :

... visitez `` notre site sur les canidés `` ...

... `` 30millionsdamis.fr

`` ...

chiens.html :

texte

... préférer les ``

``

image

`` ...

3.10 Balises sans contenus

→ La plus part des balises ont un contenu :

↳ Ex : `belle`

→ D'autres balises n'ont pas de contenu :

↳ Ex : `<hr/>`, ou encore ``.

↳ Plutôt que d'écrire la balise ouvrante et la balise fermante

`<hr><hr/>`



on fusionne ces deux balises : `<hr/>`

→ **On n'écrit qu'une seule balise dans laquelle le slash se situe à la fin de la balise :**

`<hr/>`

``

3.11 Balises

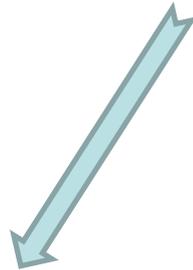
→ La balise n'a aucun effet :

Un peu de texte

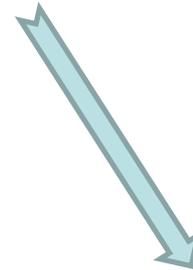


Un peu de texte

↳ Utile pour **délimiter** une portion du texte entre et



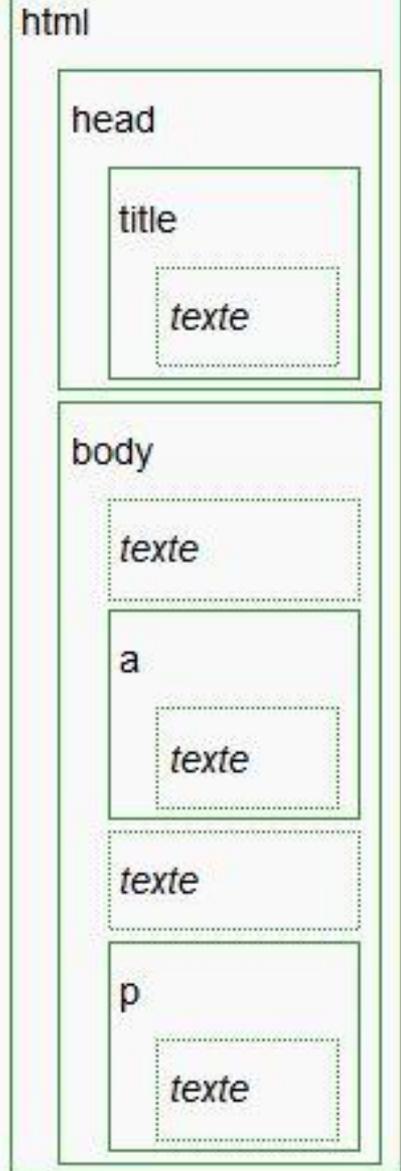
Mettre en forme la
portion de texte
(CSS)



Modifier la portion de
texte
(JavaScript)

3.12 Architecture générale d'une page web

```
<!DOCTYPE html>
<html>  <!-- Marque le début du document HTML -->
  <head> <!-- Marque le début de la zone d'en-tête -->
    <title> <!-- Titre du document -->
      Exemple de HTML
    </title>
  </head> <!-- Marque la fin de la zone d'en-tête -->
  <body> <!-- Marque le début du corps de la page -->
    Ceci est une phrase avec un <a href="cible.html">hyperlien</a>.
    <p>
      Ceci est un paragraphe où il n'y a pas d'hyperlien.
    </p>
  </body> <!-- Marque la fin du corps de la page -->
</html> <!-- Marque la fin du document HTML -->
```



REMARQUE :

<!DOCTYPE html>

➔ <! DOCTYPE> est la 1^{ère} déclaration écrite dans un document HTML, avant la balise <html>

➔ <! DOCTYPE> n'est pas une balise <html>

↳ Instruction qui permet au navigateur web de connaître la version de HTML dans laquelle est écrite la page.

➔ <! DOCTYPE> fait référence à une DTD (Document Type Definition) – Définition de la structure d'un document.

↳ Spécifie les règles du HTML, de sorte que les navigateurs affichent correctement le contenu

Voir : https://www.w3schools.com/tags/tag_doctype.asp
https://fr.wikipedia.org/wiki/Standard_Generalized_Markup_Language

➔ Le HTML5 ne repose pas sur le SGML (Standard Generalized Markup Language). Il ne nécessite pas de référence à une DTD

↳ On valide le HTML5 avec un schéma XSD (XML Schema Definition)

Voir : https://www.w3schools.com/xml/schema_intro.asp



**Toujours ajouter la déclaration
<!DOCTYPE html> à un document HTML**

IV.

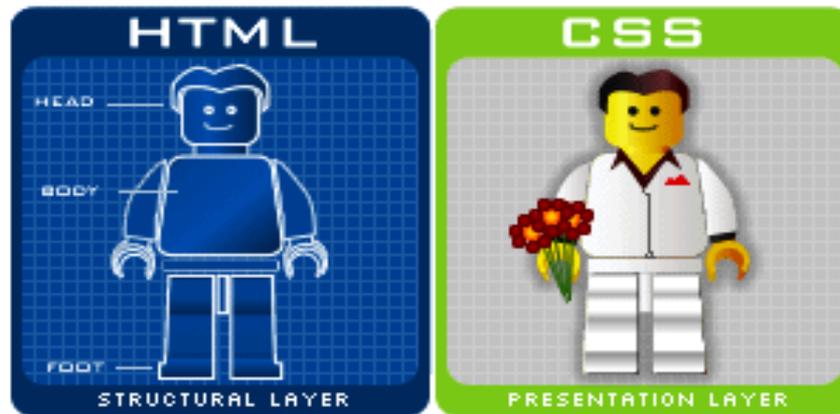
La CSS

Cascading Style Sheets

habille le HTML

4.1 A quoi sert la CSS

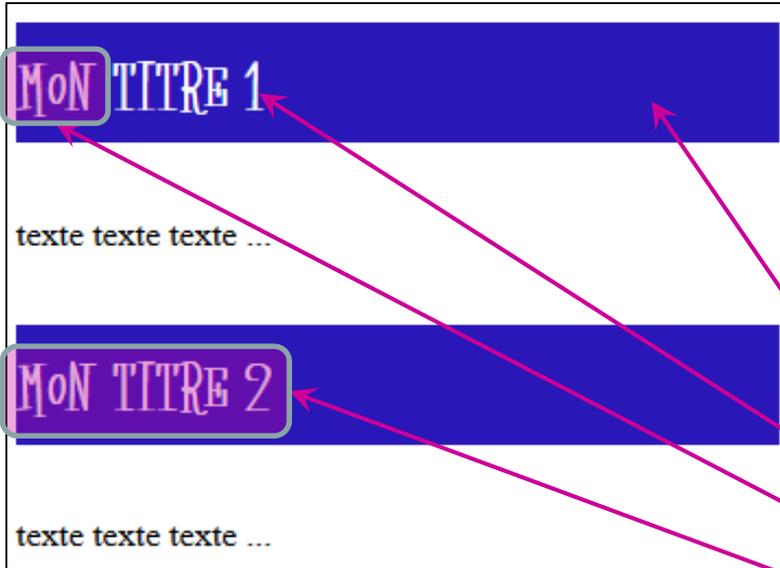
Les **CSS** (**Cascading Style Sheets - feuilles de style en cascade**, forment un langage informatique qui décrit la présentation des documents HTML.



Idée principale : Séparer le fond de la forme

- ↳ Mettre en forme le texte (police, taille du texte, style, couleur ...)
- ↳ Positionner, mettre en page des différents éléments de la page

4.2 La syntaxe de la CSS



Code HTML

```
<h1> Mon titre 1 </h1>  
<p> texte texte texte ... </p>  
<h1> Mon titre 2 </h1>  
<p> texte texte texte ...</p>
```

Code CSS

```
h1 {  
  background-color : #2A17B7;  
  color : white;  
  font-size : 20px;  
  font-family : darkmoon;  
}
```

Écriture générale:

La syntaxe de base de CSS est composée de 3 parties :

- un sélecteur
 - une propriété
 - une valeur
- sélecteur { propriété : valeur }

Exemple:

h1 { font-size : 20px },

sélecteur

propriété

valeur

↳ Tous les titres principaux du document auront une taille de 20 pixels

Propriétés:

- ➔ Si la valeur d'un attribut contient un espace, alors la valeur de l'attribut s'écrit en guillemets :

```
h2 { font-family : "sans serif" }
```

- ➔ Il est possible de définir plusieurs attributs pour un même sélecteur. Chaque propriété sera séparée par un point-virgule :

```
p {font-family:"sans serif"; font-size: 90%; color: blue}
```



En écrivant une propriété par ligne,
la lisibilité est meilleure :

```
P
```

```
{
```

```
font-family:"sans serif";
```

```
font-size: 90%;
```

```
color: blue
```

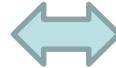
```
}
```

4.3 Grouper les sélecteurs

Si certaines propriétés s'appliquent à plusieurs sélecteurs, il est possible de les grouper :

```
h1{
  color : blue;
  font-family : "sans serif";
}

h2{
  color : blue;
  font-family : "sans serif";
}
```



```
h1,h2{
  color : blue;
  font-family : "sans serif";
}
```

4.4 Les différentes manières de placer le code CSS

➔ Dans les balises HTML avec l'attribut *style* : **une méthode lourde à éviter !!**

```
<h1 style = "background-color : #2A17B7;
            color : white;font-size : 20px;
            font-family : darkmoon;">
    Mon titre 1
</h1>
<p> texte texte texte ...</p>
<h1> Mon titre 2 </h1>
<p> texte textetexte ...</p>
```



↳ Dans ce cas, seule la première balise est concernée.

➔ Dans le « HEAD » en précisant les balises concernées:
une méthode à éviter aussi !!

```
<html>
  <head>
    <style>
      h1{
        background-color : #2A17B7;
        color : white;
        font-size : 3em;
        font-family : darkmoon;
      }
    </style>
  </head>
  <body>
    <h1> Mon titre 1 </h1>
    <p> texte texte texte ... </p>
    <h1> Mon titre 2 </h1>
    <p> texte texte texte ... </p>
  </body>
</html>
```



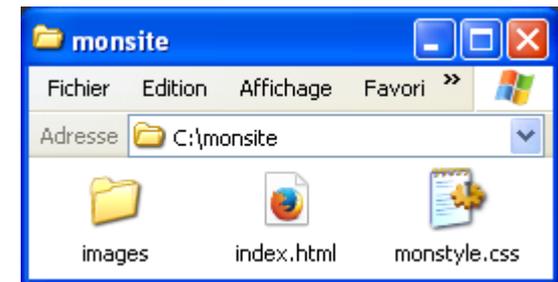
↳ Dans ce cas, toutes les balises <h1> du document sont concernées.

➔ Dans un fichier (par exemple : *monstyle.CSS*) séparé :
LA méthode à utiliser autant que possible !!

```
<html>
<head>
  <linkrel="stylesheet"
    ref="monstyle.css">
</head>
<body>
  <h1> Mon titre 1 </h1>
  <p> texte texte texte ... </p>
  <h1> Mon titre 2 </h1>
  <p> texte texte texte ... </p>
</body>
</html>
```

```
body{
  background : white;
}

h1{
  background-color : #2A17B7;
  color : white;
  font-size : 3em;
  font-family : darkmoon;
}
```



Code CSS à enregistrer dans le
fichier *monstyle.CSS* dans le
dossier courant...

4.5 Formater certains éléments avec l'attribut « class »

➔ Il est possible de formater plusieurs éléments de même type avec la même mise en forme en utilisant l'attribut « *class* ».

↳ **Exemple** : sur un site de mathématique, une mise en forme différente pour les **définitions** et les **théorèmes**

HTML

```
<p class='definition'>
```

```
La cocotte minute est un oiseau  
rapide.
```

```
</p>
```

```
<p class='theorem'>
```

```
Le carré de l'hypothénuse est  
égal à quelque chose...
```

```
</p>
```

```
<p class='theorem'>
```

```
Si le A est faux, et B est  
vrai, alors A ou B est vrai.
```

```
</p>
```

CSS

```
p.theorem{  
background-color : #EE4444;  
font-style : italic;  
border-style : dashed;  
border-width : 1px;  
margin-left : 10px;  
}  
p.definition{  
background-color : #4444EE;  
font-weight : bold;  
padding : 10px;  
}
```

La cocotte minute est un oiseau rapide.

Le carré de l'hypothénuse est égal à quelque chose...

Si le A est faux, et B est vrai, alors A ou B est vrai.

4.6 Formater un élément précis avec l'attribut « id »

➔ Il est possible de formater un élément précis avec une mise singulière en utilisant l'attribut « *id* ».

↳ **Exemple** : sur le site de mathématique précédent, vous souhaitez qu'un des deux **théorèmes** soit mis en valeur par un **encadrement rouge**.

HTML

```
<p class='definition'>
```

```
La cocotte minute est un oiseau rapide.
```

```
</p>
```

```
<p class='theorem'>
```

```
Le carré de l'hypothénuse est égal à  
quelque chose...
```

```
</p>
```

```
<p id='th_logique_1' class='theorem'>
```

```
Si le A est faux, et B est vrai, alors A  
ou B est vrai.
```

```
</p>
```

CSS

```
p.theorem{  
background-color : #EE4444;  
font-style : italic;  
border-style : dashed;  
border-width : 1px;  
margin-left : 10px;  
}
```

```
p#th_logique_1{  
border-style : solid;  
border-width : 4px;  
border-color : red;  
}
```

La cocotte minute est un oiseau rapide.

*Le carré de l'hypothénuse est égal à quelque
chose...*

Si le A est faux, et B est vrai, alors A ou B est vrai.

4.7 Feuille de style et visualisation en fonction du type d'appareil utilisé

➔ Les fichiers de style doivent être inclus dans le fichier html entre les balises <head></head> :

```
<link href="css/style.css" rel="stylesheet" type="text/css" media="screen" />
```

↳ L'attribut media permet de définir différentes feuilles de style suivant le type d'appareil utilisé :

- **screen** : pour un écran
- **print** : pour une imprimante
- **handheld** : pour un téléphone portable
- etc

4.8 Balise div, marges et espacement

➔ Une balise `<div>`, qui signifie « **division du document** », est un **conteneur** dans lequel des éléments sont placés pour les maintenir ensemble.

Permet de:

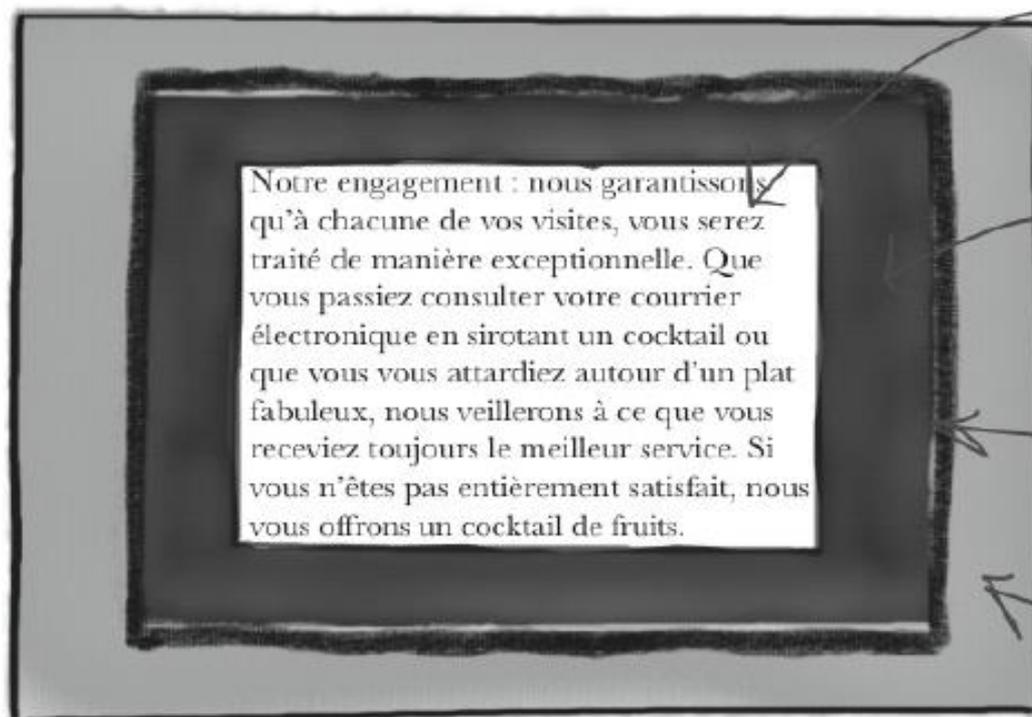
↳ Appliquer un même style à un groupe d'éléments

↳ Placer différents éléments ensembles sur une page en utilisant les attributs **class** ou **id**.

Exemple:

Du point de vue des CSS, chaque élément est une boîte.

Une boîte est constituée d'une aire de contenu avec ses espacements, ses bordures et ses marges.



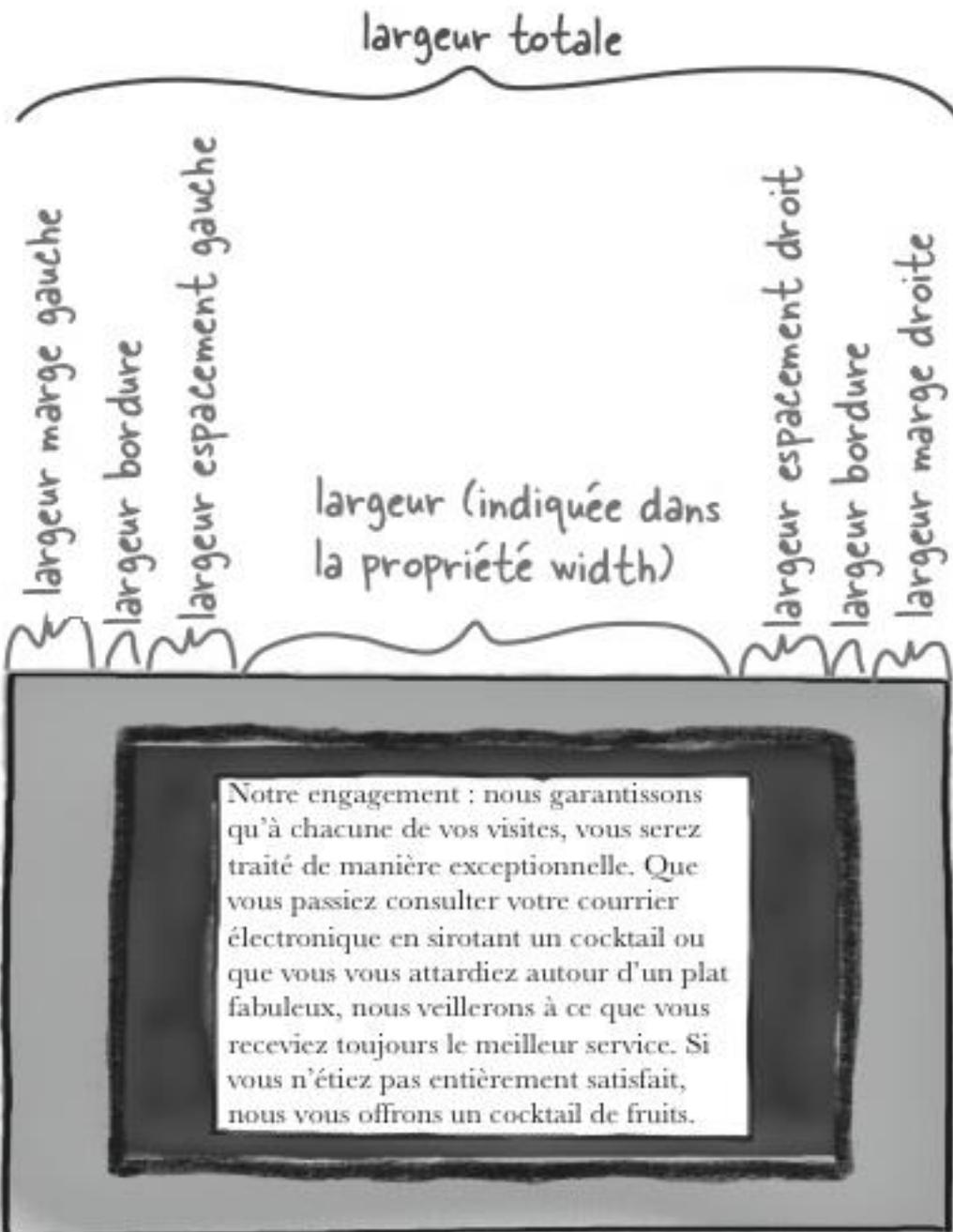
L'aire de contenu accueille... le contenu (texte ou image, par exemple).

Cette aire est entourée d'un espacement transparent (optionnel).

Une bordure peut être placée autour de l'espacement.

Et enfin, une marge transparente (également optionnelle) encadre le tout.

Tous les éléments sont traités comme des boîtes : paragraphes, titres, citations, listes, items de liste, etc. Même les éléments en-ligne comme `` et les liens le sont.



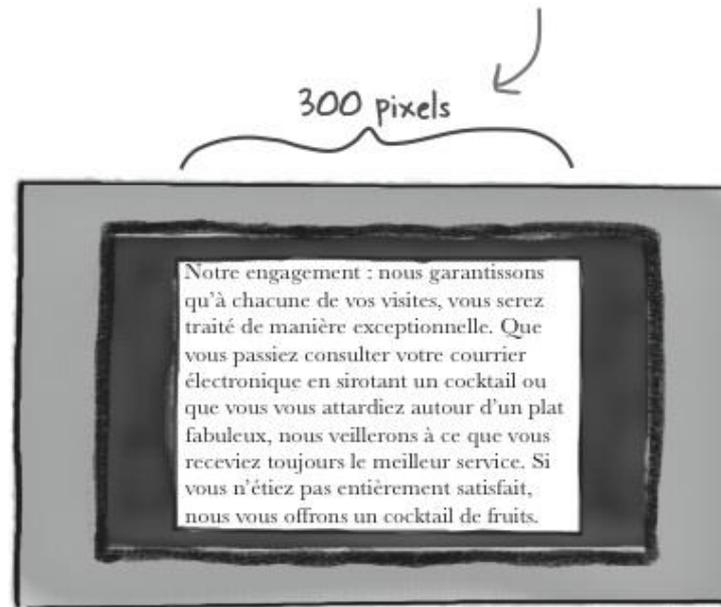
- ➔ Pour ajouter de l'**espacement**, on utilise la propriété **padding**.
- ➔ Pour ajouter une **marge**, on utilise la propriété **margin**.



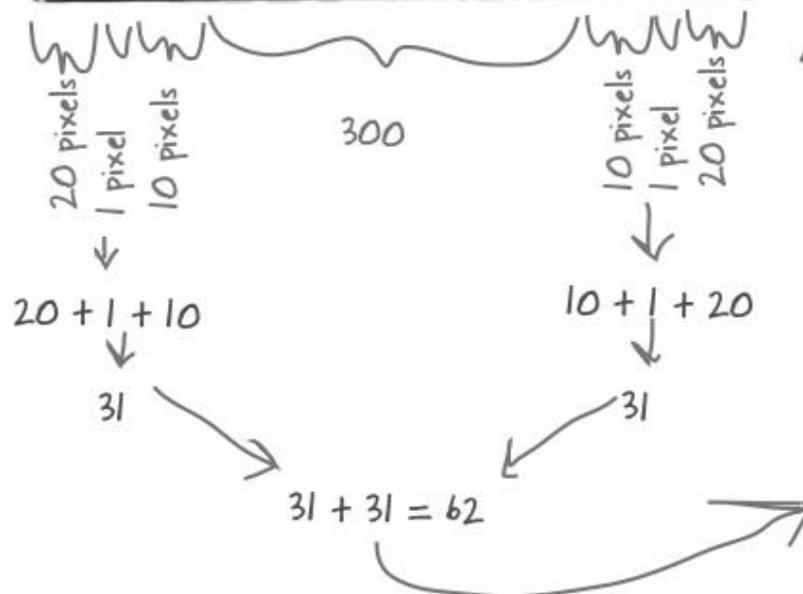
Largeur totale : additionner la largeur de l'aire de contenu à celles des marges de gauche et de droite plus celle de la bordure qu'il faudra compter deux fois puisqu'il y en a une à gauche et une autre à droite

Exemple:

(1) L'aire de contenu est de 300 pixels de large.



(2) Calculez la taille des marges, des espacements et des bordures.



(3) La taille totale est de 62 pixels que l'on additionne à celle de l'aire de contenu de 300 pixels, ce qui donne $300 + 62 = 362$ pixels de large pour la boîte entière.

4.9 Interligne

➔ La propriété **line-height** permet de déterminer la taille de l'espace vertical situé entre deux lignes de texte.

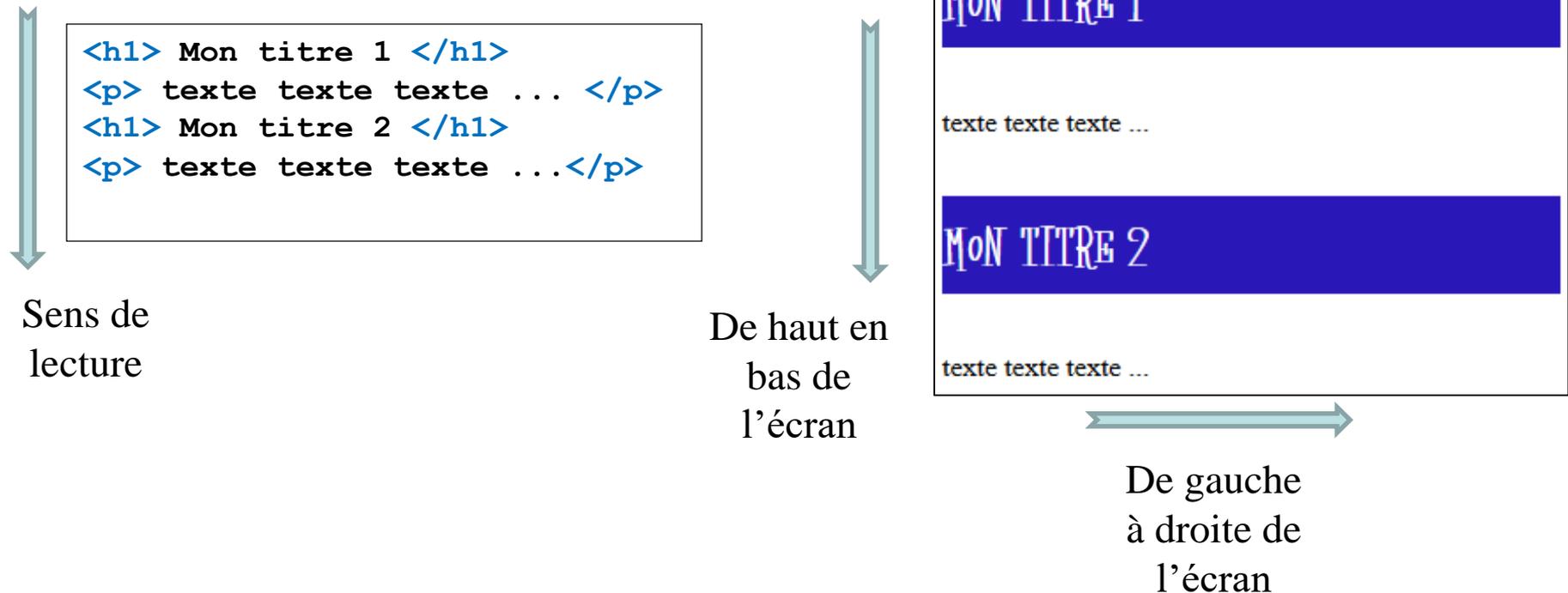
↳ il est possible de l'exprimer :

- en **pixels (px)**,
- en **em** (taille de police de caractère)
- en **pourcentage (%)**

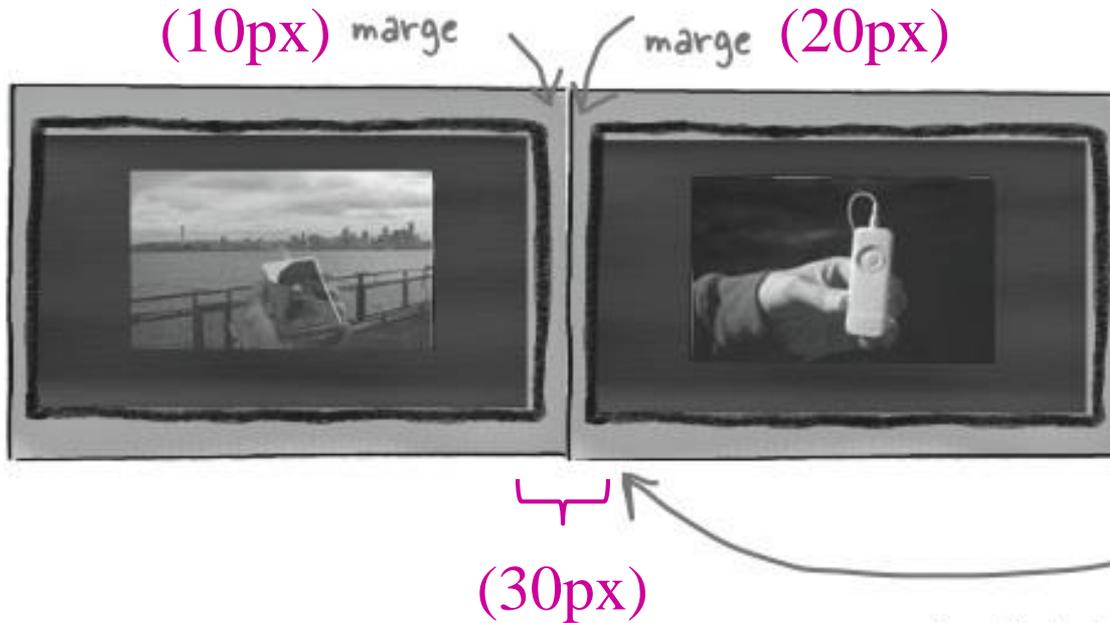
ces deux derniers étant relatifs à la taille de la police.

4.10 Le flux : placement des éléments sur une page

- ➔ Le **navigateur** utilise le **flux** pour effectuer la **mise en pages** des éléments HTML
- ➔ Affichage des éléments (blocs) dans l'ordre de lecture:



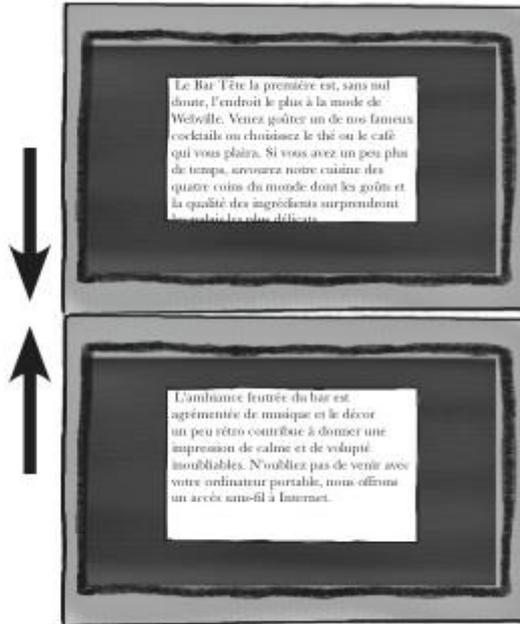
➔ Les éléments *en-ligne* se placent les uns à côté des autres depuis le coin en haut à gauche jusqu'à celui en bas à droite.



Ces deux images côte à côte sont des éléments en-ligne. Le navigateur se sert de chacune de leur marge pour calculer la taille de l'espace qui les sépare.

➔ Éléments placés verticalement

Les marges de ces blocs superposés sont fusionnées par le navigateur.



La hauteur de la marge commune est celle de la marge la plus grande. Si la marge du bas de l'élément du dessus est de 10 pixels et celle du haut de l'élément placé en-dessous est de 20 pixels, la marge résultant de la fusion sera de 20 pixels.



4.11 Éléments flottants dans la page

➔ Il est possible d'insérer des **blocs flottants** pour lesquels l'ordre d'empilement est légèrement différent de celui du flux normal.

Ordre de positionnement des blocs

1. L'arrière-plan et les bordures de l'élément racine du document ;
2. Les blocs qui descendent les uns à la suite des autres et qui sont situés dans le flux normal, dans l'ordre dans lequel ils apparaissent dans le code HTML ;
3. Les blocs flottants ;
4. Les éléments enfants positionnés, dans leur ordre d'apparition dans le code HTML ;

Procédure:

1. Donner un identifiant unique (id="id_flot") à l'élément que vous voulez faire flotter
2. Donner une largeur à cet élément, par exemple 200px
3. Ajouter la propriété float.

```
#id_flot
{
width: 200px;
float: right;
}
```

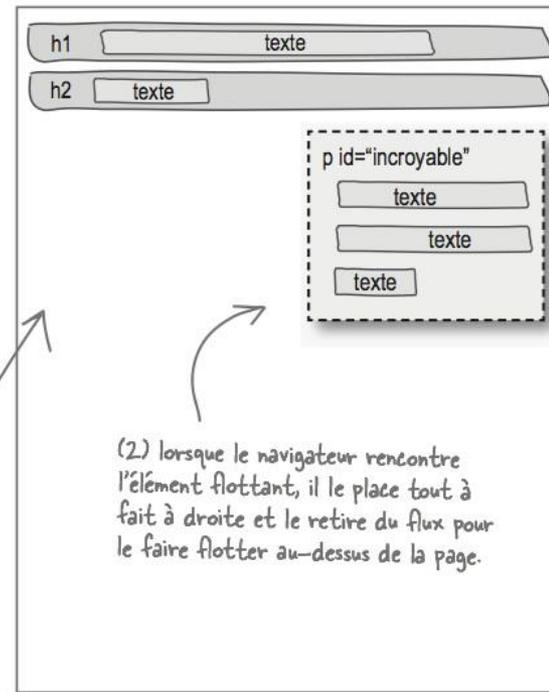
Faites-le flotter

Ajoutons la propriété **float**. Celle-ci peut être mise à gauche ou bien à droite, nous retenons ce choix :

```
#incroyable {  
  width: 200px;  
  float: right;  
}
```

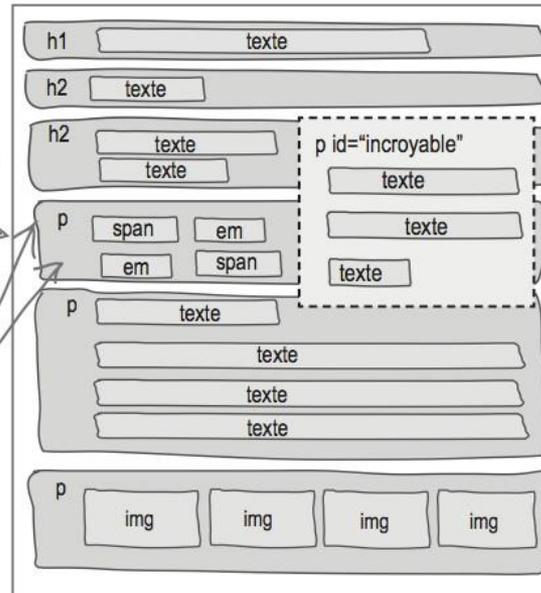
Voyons maintenant comment ce paragraphe et ses voisins sont gérés par le navigateur.

(1) Le navigateur place le flux des éléments de haut en bas, comme d'habitude.



(2) lorsque le navigateur rencontre l'élément flottant, il le place tout à fait à droite et le retire du flux pour le faire flotter au-dessus de la page.

(3) Comme ce paragraphe flottant n'est plus géré par le flux normal, les éléments de bloc se comportent comme si le paragraphe en question n'existait pas.



Les éléments de bloc sont placés sous l'élément flottant puisque celui-ci n'est plus géré par le flux.

(4) Mais quand les éléments en-ligne sont positionnés, ils respectent les limites de l'élément flottant en se positionnant autour de lui.

Les éléments en-ligne inclus dans les éléments de bloc se positionnent autour des bordures de l'élément flottant.