

Module 1106
Initiation au développement web

Nathalie Friburger

September 17, 2021

Table des Matières

1 Module R109 : Organisation	1
1.1 Organisation du module R109	1
1.1.1 Module R109 : Introduction aux technologies Web	1
I Introduction	3
2 Les sites web	5
2.1 Structure d'un site web	5
2.1.1 Agencement de la page	5
2.1.2 Eléments de navigation	5
2.1.3 Adresse d'un site web	8
2.1.4 Navigation dans les dossiers	8
2.2 Fonctionnement d'un site web	9
2.2.1 Création d'un site web	9
2.2.2 L'hébergement	10
2.2.3 Architecture client / serveur	10
2.2.4 Les langages du web	10
2.2.5 Sites statiques ou dynamiques	10
II HTML5	13
3 HTML5 ou comment structurer une page web.	15
3.1 Les principes	15
3.1.1 Les balises	15
3.1.2 Modèle d'une page html	16
3.2 Structurer avec des balises	18
3.2.1 Mettre en valeur	18
3.2.2 Sauts de lignes et paragraphes	19
3.2.3 Les listes	19
3.2.4 Les liens	20
3.3 Les différentes zones d'une page web	25
3.3.1 En-tête et pied de page	25
3.3.2 La navigation dans le site	25
3.3.3 Les différents types de contenus	27
3.3.4 Le comportement des balises HTML	27
3.4 Tableaux	28
3.4.1 Les tableaux	28
3.4.2 Organisation des éléments du HTML5	30
III CSS3	31
4 Les feuilles de style CSS	33
4.1 Introduction	33
4.1.1 Historique	33
4.1.2 Pourquoi ?	33

4.1.3	Définition de propriétés des éléments HTML	37
4.1.4	Où spécifier les styles ?	38
4.1.5	Compatibilité	38
4.2	Mettre un style	38
4.2.1	Les couleurs et les fonds	38
4.2.2	Les fontes (polices de caractères)	40
4.2.3	Les bordures et les marges	42
4.3	Classes etc.	44
4.3.1	Héritage	44
4.3.2	Les éléments	45
4.3.3	Les classes de style	46
4.3.4	Les identifiants	47
4.3.5	Les pseudo-classes	48
4.4	Positionnement	49
4.4.1	Comportements	49
4.4.2	Les grilles	53
4.5	RWD	61
4.5.1	Qu'est-ce que le Responsive Web Design ?	61
4.5.2	Les media queries	61
IV	Les réseaux, Internet, le web ... ?	65
5	Internet et le Web	67
5.1	D'où vient l'Internet ?	67
5.1.1	Les premiers réseaux	67
5.1.2	Les années 80 en France	67
5.1.3	Internet, un réseau décentralisé	69
5.2	Le web	69
5.2.1	Les navigateurs	69
5.2.2	Traffic / Nombre de sites web	69
5.2.3	Qu'est-ce que le web ?	72
5.3	Les sites web	72
5.3.1	Les acteurs du web	72
5.3.2	Qu'est ce qu'un site web ?	72
5.3.3	Nom de domaine	73

Chapitre 1

Module R109 : Organisation

1.1 Organisation du module R109

1.1.1 Module R109 : Introduction aux technologies Web

Module R109 : Introduction aux technologies Web

- **Volume horaire prévu**
 - 2 CM = 3h
 - 2 TD = 3h
 - 2 TP = 6h
- Objectif du module :
 - Bases conceptuelles et pratiques pour écrire et modifier des pages Web dans un langage normalisé de description de contenus et de sa présentation, utiles pour le site web et intranet d'une entreprise
 - Création de contenus Web (thème abordé par PIX, <https://pix.fr/competences>)
 - Technologies mises en oeuvre pour délivrer ses contenus aux utilisateurs par le biais d'un navigateur Web.

Module R109 : Introduction aux technologies Web

- Contenu :
 - Utilisation avancée d'un navigateur Web
 - Structure d'un site Web : client-serveur, arborescence, URL
 - Structure d'une page : langage à balise, mise en forme et feuilles de styles (notions élémentaires de CSS), notions de responsive design
 - Contenu d'une page : éléments multimédia, encodage des caractères
 - Sensibilisation aux mentions obligatoires d'un site Web (mentions légales, copyright, ...)
- SAÉ (situation d'apprentissage et d'évaluation)
 - SAÉ14 Se présenter sur Internet
 - SAÉ15 Traiter des données

Liens et prolongements en BUT R&T

- RT3-Programmer Niveau 1 : AC0314 Connaître l'architecture et les technologies d'un site Web

Evaluation du module

- Travail de TP :
 - En TP, travail personnel (pas de binôme)
 - Un QCM noté à chaque fin de séance
 - Le dernier TP sera noté
- Devoir surveillé en fin de module
 - seule une page recto-verso de notes MANUSCRITES (= écrite à la main) sera autorisée

Accès à Celene

Celene = plateforme de cours en ligne pour l'université de Tours

- soit par l'ENT : ent.univ-tours.fr Onglet études / Cliquez sur Mes Cours en Ligne...
- soit directement par : celene.univ-tours.fr

Rappels

- Pendant les enseignements : pas de téléphone portable/PC
- Avoir son poly de cours
- Écart de comportement ⇒ exclusion de la séance ET absence non justifiée
- **En TP :**
 - Les sujets sont sur Celene ⇒ au travail dès qu'on rentre dans la salle !
 - On ne peut sortir que lorsqu'on a fait la preuve qu'on a tout fini correctement !

Première partie

Introduction

Chapitre 2

Les sites web

2.1 Structure d'un site web

Adresse : <https://www.amboise-valdeloire.com/destination-amboise-chateaux-loire/leonard-de-vinci/>

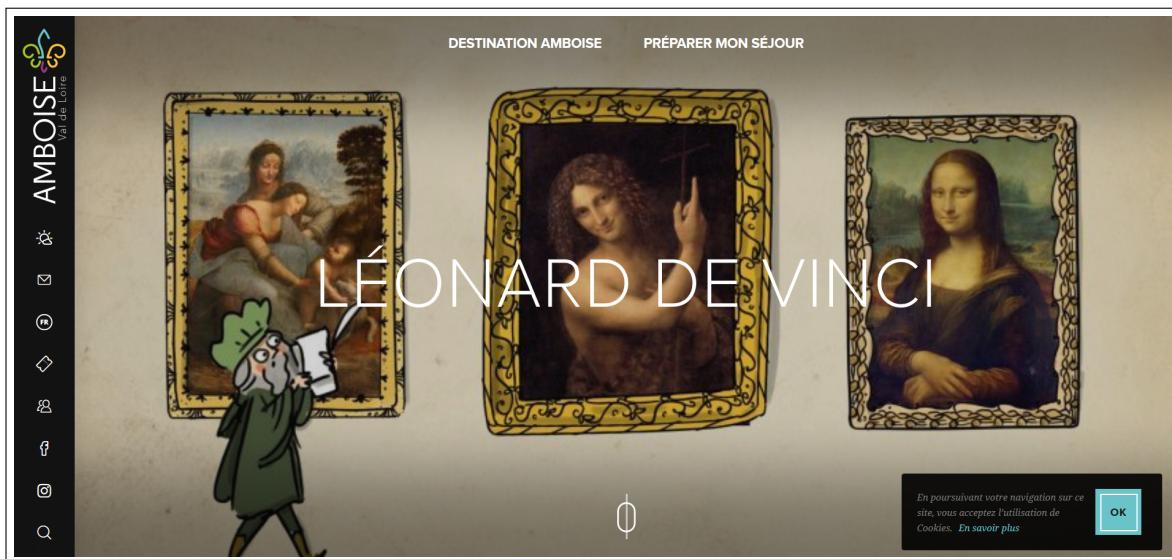


FIGURE 2.1 – site d'Amboise

Adresse : <http://iut-blois.univ-tours.fr/>

Adresse : <https://fr.wikipedia.org/wiki/Amboise/>

2.1.1 Agencement de la page

Agencement de la page

- des éléments invariables, identiques sur toutes les pages
- et des éléments variables d'une page à l'autre.

2.1.2 Eléments de navigation

Navigation par onglets

Flèches de navigation

retour à la page d'accueil précédent suivant

Fil d'ariane (en anglais breadcrumbs trail)

Suite de liens hiérarchisée pour se repérer et se déplacer dans le site

- *breadcrumb* (= miette de pain) dû au conte de Grimm "Hansel et Gretel"
- *fil d'Ariane* dû à Ariane et le labyrinthe du Minotaure (mythologie grecque).

Plan du site (en anglais, site map)

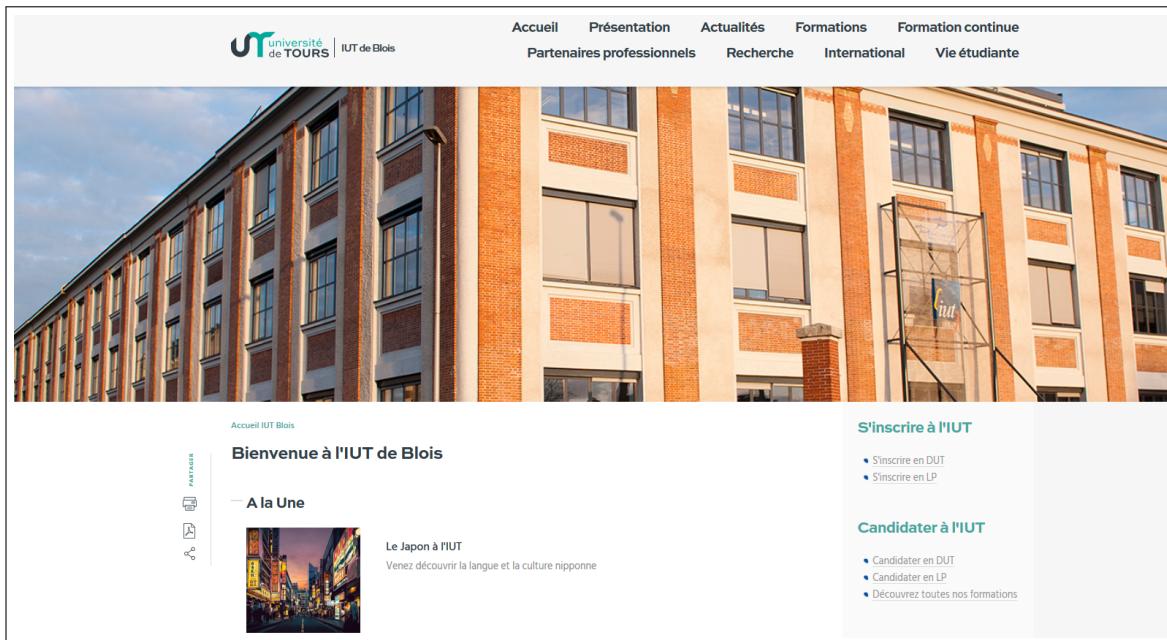


FIGURE 2.2 – site de l'IUT de Blois

FIGURE 2.3 – site de l'IUT de Blois

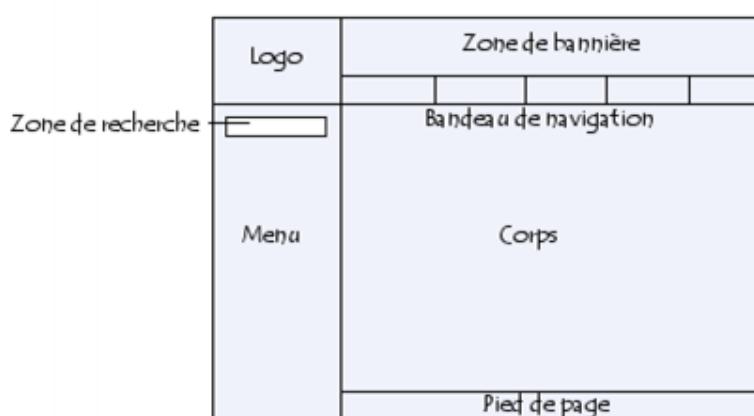


FIGURE 2.4 – Agencement de la page



FIGURE 2.5 – onglets



FIGURE 2.6 – home



FIGURE 2.7 – flèche gauche



FIGURE 2.8 – flèche droite

FIGURE 2.9 – fil d'ariane

FIGURE 2.10 – autre fil d'ariane

Plan du site
<ul style="list-style-type: none"> • Accueil • Présentation <ul style="list-style-type: none"> • Organisation • Informations pratiques <ul style="list-style-type: none"> • Annuaire • Historique <ul style="list-style-type: none"> • Lieux de formation • Actualités <ul style="list-style-type: none"> • Archives • Agenda • Formations <ul style="list-style-type: none"> • DUT Mesures Physiques <ul style="list-style-type: none"> • Programme • Enseignants • Foire aux questions (FAQ) • Salles de travaux pratiques • Témoignages d'étudiants

FIGURE 2.11 – plan du site web de l'IUT

2.1.3 Adresse d'un site web

Exemples d'adresses :

- adresse google :
`http://www.google.com`
- adresse RT :
`http://iut-blois.univ-tours.fr/formations/reseaux-et-telecommunications-92463.kjsp?RH=IUTBLOIS_FRRF=1227524927941`
- adresse Venir à Amboise :
`http://www.amboise-valdeloire.com/infos-pratiques/venir-a-amboise`

Utilisation du système d'adressage URL (Uniform Ressource Locator)

Ex : http://www.monadresse.amoi.fr/pages/infos/moninfo.html

<code>http://</code>	protocole (= manière de communiquer)
<code>www.monadresse.amoi.fr</code>	nom du serveur
<code>pages/info</code>	chemin sur le serveur
<code>moninfo.html</code>	page HTML

Nom du serveur

En 3 parties

<code>www</code>	serveur sur le world wide web
<code>monadresse.amoi</code>	nom du serveur
<code>fr</code>	type de serveur

2.1.4 Navigation dans les dossiers

Exemple :

Arborescence de dossiers de `www.moninfo.amoi.fr`

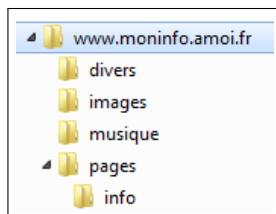


FIGURE 2.12 – arborescence de `www.moninfo.amoi.fr`

- Le dossier `images` contient 2 fichiers :
`image1.jpg` et `image2.jpg`
- Le dossier `pages` contient 2 fichiers :
`page1.php` et `page2.php`
- Le dossier `infos` contient 3 fichiers :
`moninfo1.html`, `moninfo2.html` et `moninfo3.html`

Exemple :

Arborescence de dossiers de `www.moninfo.amoi.fr`

Pour accéder/afficher un fichier, on donne le chemin absolu de ce fichier ou le chemin relatif

Chemin absolu

- ex : afficher le fichier `moninfo2.html`
`http://www.moninfo.amoi.fr/pages/infos/moninfo2.html`
- ex : afficher le fichier `page1.php`
`http://www.moninfo.amoi.fr/pages/page1.php`

Exemple :

Arborescence de dossiers de `www.moninfo.amoi.fr`

Chemin relatif

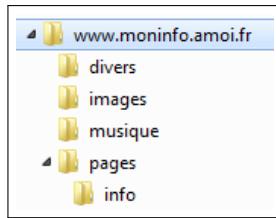


FIGURE 2.13 – arborescence de www.moninfo.amoi.fr

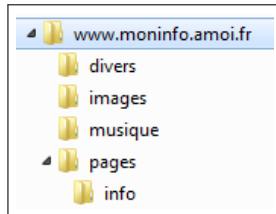


FIGURE 2.14 – arborescence de www.moninfo.amoi.fr

- ex : afficher moninfo1.html si on est dans le dossier racine (www.moninfo.amoi.fr)

./pages/infos/moninfo1.html
- ex : afficher moninfo1.html si on est dans le dossier contenant ce fichier (www.moninfo.amoi.fr/pages/infos)

moninfo1.html

Fichier par défaut demandé pour un dossier : index.html

Exemple :

Arborescence de dossiers de www.moninfo.amoi.fr

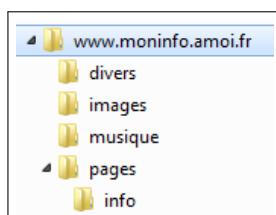


FIGURE 2.15 – arborescence de www.moninfo.fr

Chemin relatif

- ex : afficher moninfo1.html si on est dans le dossier pages (www.moninfo.amoi.fr/pages)

.//infos/moninfo1.html
- ex : afficher moninfo1.html si on est dans le dossier images (www.moninfo.amoi.fr/images)

.//pages/infos/moninfo1.html

2.2 Fonctionnement d'un site web

2.2.1 Crédation d'un site web

- Crédation
 - Conception
 - Réalisation
 - En éditant les fichiers HTML à la main
 - En utilisant un éditeur HTML WYSIWYG (What You See Is What You Get).
- Exploitation du site
 - Veille
 - Promotion et référencement
 - Maintenance et mise à jour

2.2.2 L'hébergement

- Hébergement
 - mise à disposition d'un serveur web connecté en permanence à internet
 - les hébergeurs gratuits (souvent en échange de pub)
 - les hébergeurs professionnels payants
- Mise en ligne
 - transférer (copier/upload) les fichiers de votre ordinateur sur le serveur.
 - à l'aide d'un client FTP (logiciel de transfert de fichiers).

Les engagements de l'hébergeur :

- Trafic : une bande passante minimale
- Disponibilité : le site est "en ligne" 24/24
- Fiabilité : peu de failles de sécurité (virus, etc.)

2.2.3 Architecture client / serveur

Principe général :

- L'utilisateur est sur un poste *client*
- Les données à accéder sont sur le *serveur*
- Le client effectue une *requête* et le serveur lui renvoie en réponse des données (fichier, flux de données, etc.)

2.2.4 Les langages du web

- eXtensible HyperText Markup Language XHTML
 - Versions actuelles xHTML 1.1 et HTML5
 - Recommandation W3C en 2000
 - Langage de balisage servant à l'écriture de pages WWW
- XHTML puis HTML5, évolutions de HTML
 - fondé sur la syntaxe définie par XML
 - Changement de syntaxe, aucune fonctionnalité n'ayant été ajoutée ou retirée
 - Extension des pages web : .htm ou .html

2 langages essentiels qui se complètent :

- HTML5 (HyperText Markup Language)
rôle : gérer et organiser le contenu
- CSS3 (Cascading Style Sheets = Feuilles de style en cascade)
rôle : gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte...)

Logiciels de création web : NVU, Kompozer, Matizha Sublime, Namo WebEditor etc.

- Environnement de développement
 - Un simple éditeur de texte suffit !
Ex : notepad ++
 - Dreamweaver ou autres ...

2.2.5 Sites statiques ou dynamiques

Sites web statiques

- HTML + css = site web statique
- Des fichiers texte contenant du code HTML
 - avec éventuellement des images et des liens vers d'autres documents.
- Les pages web sont envoyées "telles quelles"
- Rapide, mais rigide

Inconvénients :

- Maintenance difficile (modification manuelle des pages)
- Pas de page personnalisée selon le visiteur
- Pas de création de page en fonction de données d'une BD
- etc.

Solution => Sites web dynamiques

Automatiser la génération de pages web du côté du serveur

- HTML + css + d'autres langages ...
- Les pages sont " calculées " selon
 - Une base de données
 - L'architecture du site (inclusions)
 - L'utilisateur (authentification)
- Nécessite d'exécuter un programme (plus lent)
- Facilite la modification du contenu du site
- Grande majorité des sites
 - Servent à traiter la demande du client et préparer la réponse à envoyer
 - Code exécuté par le serveur => ne pourra jamais être vu par le client.

Langages serveurs

- **PHP** très important, utilisé par les plus grands CMS (wordpress, joomla, drupal, magento).
- **Java/JSP** pour sites et applications web plus rapides et sécurisés que PHP, mais lourd à développer, hébergeur qui acceptent le Java rares.
- **ASP.net** langage serveur de Microsoft, moins utilisé que PHP.
- **SQL** pour interroger les BDs

Langages clients

- **HTML** : langage client pour les sites web (gère la mise en forme)
- **CSS** : définit des règles d'affichage
- **javascript** : moyen de rendre dynamique des pages
 - peut modifier les propriétés HTML avec des transitions (ex : animations)
 - effectue des requêtes vers le serveur
 - discute avec des éléments embarqués (ex : flash, et autres)

Langages clients

- **Flash AS3** : langage orienté objet,
- **Java** : machine virtuelle directement sur le navigateur
- **XML, JSON** : pour la transmission d'informations et stockage.
- **Framework** (tel Symphony, cakePHP) : assemblage d'outils pour programmer plus proprement, plus rapidement, et plus efficacement.
- **AJAX** : manière de construire des applications Web avec Javascript/PHP => applications web très puissantes.

Deuxième partie

HTML5

Chapitre 3

HTML5 ou comment structurer une page web.

3.1 Les principes

- eXtensible HyperText Markup Language XHTML
 - Versions actuelles XHTML 1.1 et HTML5
 - Recommandation W3C en 2000
 - Langage de balisage servant à l'écriture de pages www
 - fondé sur la syntaxe définie par XML
 - Extension des pages web : .htm ou .html

3.1.1 Les balises

HTML est un langage de **structuration de document**

Le document est structuré avec des **balises**

2 types de balises

- <balise> </balise> : balise en paires, indique la *nature* du texte qu'elles encadrent
 - 1 | ` Ce texte est important. `
Affichage : **Ce texte est important.**
- <balise/> : balise orpheline, insère un élément à un endroit précis (ex : un saut de ligne, une image)
 - 1 | `Je veux sauter
 à la ligne suivante`
Affichage :

Les attributs

= options des balises

`<balise attribut="valeur">`

Exemple

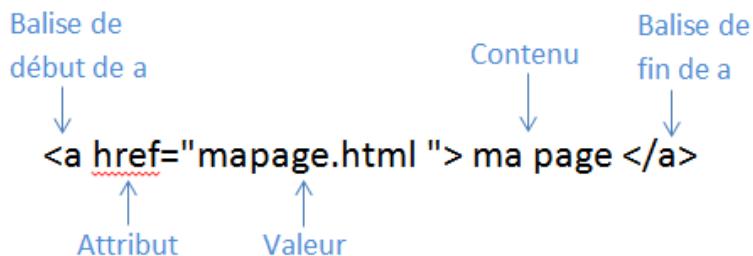


FIGURE 3.1 – html sans css

Commentaires dans les fichiers HTML

1 | `<!-- Voici mon commentaire blabla -->`

Les commentaires ne s'affichent pas !

3.1.2 Modèle d'une page html

page html minimal

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title> titre de la page </title>
6   ...
7   </head>
8   <body>
9   ...
10  </body>
11 </html>
```

(1) indique qu'il s'agit d'une page web HTML.

```

1 ...
2 <meta charset="utf-8" />
3 ...
```

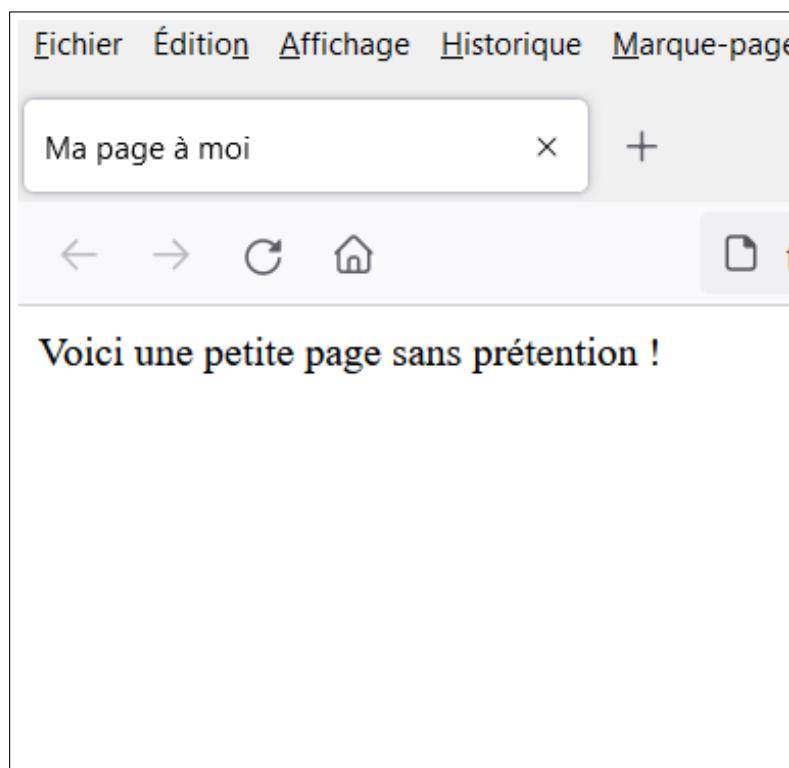
encodage (charset) du fichier html

Encodeage

- détermine l'affichage des caractères spéciaux (accents, idéogrammes chinois et japonais, caractères arabes, etc.).
- en fonction des langues :
 - ISO-8859-1, OEM 775, Windows-1253 ...
 - UTF-8 : permet d'afficher tous les symboles de toutes les langues !

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Ma page à moi</title>
6   </head>
7   <body>
8     Voici une petite page sans prétention !
9   </body>
10 </html>
```



<title> : titre qui s'affiche en haut de la page mais aussi dans les résultats des moteurs de recherche

Attention

Tout le monde peut voir le code HTML de vos pages présentes sur le Web.

grâce au navigateur : "Afficher le code source de la page"

-> pas d'infos importantes (ex : mots de passe)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="description" content="...">
<link rel="stylesheet" href="main.css">
</head>
<body>
<h1>Le monde du HTML</h1>
<h2>Suivez le guide !</h2>
<ul>
<li>titre 1</li>
<li>titre 2</li>
<li>titre 3</li>
</ul>
<!-- commentaire -->
<p>
Bienvenue ...
</p>
```

— Sans une bonne présentation du code ...

— difficile d'en voir les constituants

— d'en voir la structure

— risque d'erreurs de saisie

```

1  <!DOCTYPE html>
2  ▼ <html>
3  ▼ <head>
4      <meta charset="utf-8">
5      <title>Première page</title>
6      <link rel="stylesheet" href="main.css">
7  </head>
8  ▼ <body>
9      <h1>Le monde du HTML</h1>
10     <h2>Suivez le guide !</h2>
11     <!-- commentaires -->
12    <p>
13        Bienvenue ...
14    </p>
15

```

- Techniques :
 - Indentation
 - Coloration syntaxique d'un langage
 - Masquage et démasquage de blocs,
 - Mise en relief d'un délimiteur quand le curseur passe sur l'autre qui lui correspond, etc.

3.2 Structurer avec des balises

3.2.1 Mettre en valeur

 : emphase, mettre un peu en valeur
 : mettre bien en valeur
 <mark> : Marquer le texte pour faire ressortir une portion de texte.
 <dfn> : indiquer une définition
 <code> : pour un morceau de code
 Exemple :

```

1 Voici un <em> texte un peu en valeur </em> <br/>
2 Puis un <strong> texte plus important </strong> <br/>
3 ou un <mark> texte marqué </mark> <br/>
4 et une <dfn> définition </dfn> par exemple <br/>
5 Enfin un peu de code : <code>printf("%d ", cmpt);</code>

```

Affichage :

Voici un *texte un peu en valeur*
 Puis un **texte plus important**
 ou un **texte marqué**
 et une *définition* par exemple
 Enfin un peu de code : `printf("%d ", cmpt);`

FIGURE 3.2 – Mettre en valeur le texte

— Les balises peuvent s'imbriquer les unes dans les autres

```
1 | <p>Voici un <strong>exemple</strong></p>
```

— Attention : les balises doivent s'imbriquer correctement !

```
1 | <p>Voici un mauvais <strong>exemple</p> </strong>
```

3.2.2 Sauts de lignes et paragraphes

Sauts de lignes

— Si on écrit le texte html suivant :

```
1 | Bienvenue
2 | Voici mon premier saut de ligne
3 | Et en voici un deuxième !
```

— MAIS le résultat affiché dans le navigateur sera :

```
| Bienvenue Voici mon premier saut de ligne Et en voici un deuxième !
```

— Solution : la balise


```
1 | Bienvenue <br />
2 | Voici mon premier saut de ligne <br />
3 | Et en voici un deuxième !
```

Affichage :

```
| Bienvenue
| Voici mon premier saut de ligne
| Et en voici un deuxième !
```

Sauts de paragraphe

```
1 | Bienvenu
2 | <p>
3 | Voici mon premier saut de ligne <br />
4 | Et en voici un deuxième !
5 | </p>
6 | <p>Et voici un deuxième paragraphe</p>
```

Affichage :

```
| Bienvenue
|
| Voici mon premier saut de ligne
| Et en voici un deuxième !
|
| Et voici un deuxième paragraphe
```

3.2.3 Les listes

ul = unnumbered list

```
1 | Acheter
2 | <ul>
3 |   <li>du pain</li>
4 |   <li>du beurre</li>
5 |   <li>des fraises</li>
6 | </ul>
```

Affichage :

Acheter

- du pain
- du beurre
- des fraises

FIGURE 3.3 – liste

Imbrication de listes possible (créer une liste à puces dans une liste à puces).
ol = ordered list

```

1 | Les gagnants sont
2 | <ol>
3 |   <li>Dupont</li>
4 |   <li>Durand</li>
5 |   <li>Duprès</li>
6 | </ol>

```

Affichage :

Les gagnants sont

1. Dupont
2. Durand
3. Duprès

FIGURE 3.4 – liste ordonnée

dl = definition list

- <dt> donne le titre de la définition
- <dd> la définition elle-même
- Ex :

```

1 | <dl>
2 |   <dt>Lundi
3 |     <dd>Maths
4 |     <dd>Info
5 |   <dt>Mardi
6 |     <dd>Français
7 |   <dt>Mercredi
8 |     <dd>Histoire
9 | </dl>

```

Affichage :

3.2.4 Les liens

Lien : texte sur lequel on peut cliquer pour aller sur un autre endroit de la page ou sur une autre page

Affichage :

- par défaut, en bleu et souligné
- un curseur en forme de main apparaît lorsqu'on pointe dessus.
- si la page a déjà été ouverte, lien en violet.

Création d'un lien

```

<a href="http://iut-blois.univ-tours.fr">site de l'IUT</a>
<br/>
Ecrire à <a href="mailto:toto@univ-tours.fr">Toto</a>
<br/>
<a href="autrepage.html">une autre page</a>

```

Affichage :

site de l'IUT Ecrire à Toto une autre page
--

FIGURE 3.5 – Liens externes

Liens absous

On indique l'adresse complète.

```
<a href="http://www.iut-blois.univ-tours.fr/index.php">
```

Liens relatifs

On indique le chemin entre 2 pages situées dans un même dossier ou des dossiers différents.

- 2 pages situées dans des dossiers différents

```
| <a href="contenu/autrepage.html">
```

- Si plusieurs sous-dossiers

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

- Si le fichier cible est placé dans un dossier parent

```
| <a href=".../.../autrepage.html">
```

Exemple : page wikipedia sur Amboise : partie visible

Liens pour accéder aux autres parties de la page web en cours

Clic sur le lien interne "communes limitrophes" (infos présentes sur la même page !)

The screenshot shows the Wikipedia article for Amboise. The sidebar on the left contains a table of contents with sections like Géographie, Politique et administration, Histoire, and Héraldique. The main content area discusses the town's history, its status as a UNESCO World Heritage site, and its administrative details. A right sidebar provides a map, a photo of the castle, and Amboise's coat of arms. Below the sidebar, there is a detailed table of administrative information.

Pays	France
Région	Centre
Département	Indre-et-Loire
Arrondissement	Tours
Canton	Amboise
Intercommunalité	Communauté de communes Val d'Amboise
Autres	

Pour accéder rapidement à un point particulier d'une page !

Exemple : page wikipedia sur Amboise

Amboise

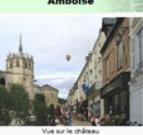
Pour les articles homonymes, voir Amboise (homonymie).

Amboise (étymologiquement « entre deux eaux », la Loire et l'Amasse) est une ville d'Indre-et-Loire située sur les bords de la Loire, dans l'arrondissement de Tours et la région Centre.

Le château fut sa renommée au célèbre château qu'il domine, à la porte de l'Oratoire, à la collégiale Saint-Cyr, au château du Chêne-à-Luc ou s'interdit Léonard de Vinci, à la demeure royale de Château-Gaillard construite par Charles VII en 1466, célèbre pour ses jardins aménagés par Dom Pacello de Mercogliano et à la piscine papale de Chateaubriant. Son nom est aussi attaché à la sanglante corporation de 1560.

C'est l'une des onze communes vignobles de l'appellation d'origine contrôlée (AOC) « Touraine-amboise ».

Amboise est située dans le périmètre du Val de Loire inscrit au patrimoine mondial de l'UNESCO¹.

Sommaire [masquer] <ul style="list-style-type: none"> 1 Géographie 1.1 Lieux-dits et écarts 1.2 Communes limitrophes 2 Politique et administration 2.1 Tendances politiques et résultats 2.2 Liste des maires 2.3 Endettement 3 Histoire 3.1 Préhistoire et Antiquité 3.2 Moyen Âge 3.3 Renaissance 3.4 Période moderne 3.5 Période contemporaine 3.6 Seconde Guerre mondiale 4 Héraldique 5 Économie 6 Monuments 7 Transports 8 Personnalités liées à la commune 8.1 XIX^e siècle et contemporaine 8.2 XX^e siècle 8.3 XVII^e siècle 8.4 XVIII^e siècle 9 Démographie 10 Jumelages 11 Notes et références 11.1 Notes 11.2 Références 12 Voir aussi 12.1 Bibliographie 12.2 Articles connexes 12.3 Liens externes 	 47° 24' 15" N 0° 58' 48" E carte  Vue sur le château  Blason Administration <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td> Pays</td> <td>France</td> </tr> <tr> <td> Région</td> <td>Centre</td> </tr> <tr> <td> Département</td> <td>Indre-et-Loire</td> </tr> <tr> <td> Arrondissement</td> <td>Tours</td> </tr> <tr> <td> Canton</td> <td>Amboise</td> </tr> <tr> <td> Intercommunalité</td> <td>Communauté de communes du Val d'Amboise</td> </tr> <tr> <td> Maire</td> <td>Christian Guyon</td> </tr> <tr> <td> Mandat</td> <td>2014-2020</td> </tr> <tr> <td> Code postal</td> <td>37400 ou 37500</td> </tr> <tr> <td> Code commune</td> <td>37001</td> </tr> </table> Démographie <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td> Densité</td> <td>320 hab./km²</td> </tr> <tr> <td> Géographie</td> <td></td> </tr> <tr> <td> Coordonnées</td> <td>47° 24' 15" Nord 0° 58' 48" Est</td> </tr> <tr> <td> Altitude</td> <td>Min. 92 m – Max. 127 m</td> </tr> <tr> <td> Superficie</td> <td>40,65 km²</td> </tr> </table> Localisation <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td> Délocalisation sur la carte</td> <td>France</td> </tr> <tr> <td colspan="2"></td> </tr> <tr> <td colspan="2">Voir la carte administrative de France</td> </tr> </table> Liens <ul style="list-style-type: none"> Site web www.ville-amboise.fr 	Pays	France	Région	Centre	Département	Indre-et-Loire	Arrondissement	Tours	Canton	Amboise	Intercommunalité	Communauté de communes du Val d'Amboise	Maire	Christian Guyon	Mandat	2014-2020	Code postal	37400 ou 37500	Code commune	37001	Densité	320 hab./km ²	Géographie		Coordonnées	47° 24' 15" Nord 0° 58' 48" Est	Altitude	Min. 92 m – Max. 127 m	Superficie	40,65 km ²	Délocalisation sur la carte	France			Voir la carte administrative de France	
Pays	France																																				
Région	Centre																																				
Département	Indre-et-Loire																																				
Arrondissement	Tours																																				
Canton	Amboise																																				
Intercommunalité	Communauté de communes du Val d'Amboise																																				
Maire	Christian Guyon																																				
Mandat	2014-2020																																				
Code postal	37400 ou 37500																																				
Code commune	37001																																				
Densité	320 hab./km ²																																				
Géographie																																					
Coordonnées	47° 24' 15" Nord 0° 58' 48" Est																																				
Altitude	Min. 92 m – Max. 127 m																																				
Superficie	40,65 km ²																																				
Délocalisation sur la carte	France																																				
																																					
Voir la carte administrative de France																																					

Géographie [[modifier](#)] [[modifier le code](#)]

Amboise s'étend, pour l'essentiel, sur la rive sud de la [Loire](#), à un endroit où le fleuve se sépare en deux bras. Il enserré une île (dénommée île Saint-Jean, en souvenir d'un prieur matinalisé par les croisés, Grande-île ou île-d'Or), qui fait partie du territoire communal.



La cité ancienne est surplombée par le plateau sur lequel est établi son [château](#).

Amboise s'est développée au nord de la Loire, avec le quartier de la gare, mais également au sud où en 1946, la commune a absorbé celle voisine de [Saint-Cyr-en-Val](#)².

Paris se trouve à 230 km.

Lieux-dits et écarts [[modifier](#)] [[modifier le code](#)]

⚠ Cette section est vide, insuffisamment détaillée ou incomplète. Votre aide est la bienvenue !

Communes limitrophes [[modifier](#)] [[modifier le code](#)]

Lussault-sur-Loire, Saint-Martin-le-Beau, Deire, La Croix-en-Touraine, Civray-de-Touraine, Souvigny-de-Touraine, [Saint-Rémy](#), Chargé, et du côté nord de la Loire : [Nazelles-Négron](#), [Poëze](#) et [Limeray](#).



Politique et administration [[modifier](#)] [[modifier le code](#)]

Tendances politiques et résultats [[modifier](#)] [[modifier le code](#)]

⚠ Article concerné : [Élections municipales de 2014 en Indre-et-Loire](#).

⚠ Cette section est vide, insuffisamment détaillée ou incomplète. Votre aide est la bienvenue !

Liste des maires [[modifier](#)] [[modifier le code](#)]

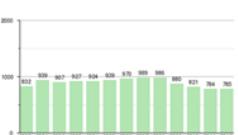
Période	Identité	Étiquette	Qualité
10 mai 1945 – 11 mai 1953	Émile Gounin		
11 mai 1953 – 2 juillet 1956	Maurice Mercier		
2 juillet 1956 – 19 mars 1959	Michel Debret		
19 mars 1959 – novembre 1992	André Chollet		
décembre 1992 – mars 2001	Bernard Debret		
mars 2001 – en cours	Christian Guyon		

Endettement [[modifier](#)] [[modifier le code](#)]³

Évolution de l'endettement (en milliers d'€)⁴ :



Évolution de l'endettement par habitant (en €)⁴ :



Exemple : page wikipedia sur Amboise : nouvelle partie visible de la page

Amboise

Pour les articles homonymes, voir Amboise (homonymie).

Amboise (étymologiquement « entre deux eaux », la Loire et l'Amasse) est une ville d'Indre-et-Loire située sur les bords de la Loire, dans l'arrondissement de Tours et la région Centre.

Le site doit sa renommée au célèbre château qui la domine, à la porte de l'empereur, à la collégiale Saint-Cyr, au château du Despuclos où s'interdit Léonard de Vinci, à la demeure royale de Château-Gaillard construite par Charles VII en 1466, célèbre pour ses jardins aménagés par Dom Pacello de Mercouplanc et à la piscine papale de Chateaubriant. Son nom est aussi attaché à la sanglier corruption de 1560.

C'est l'une des onze communes vignobles de l'appellation d'origine contrôlée (AOC) « Touraine-amboise ».

Amboise est située dans le périmètre du Val de Loire inscrit au patrimoine mondial de l'UNESCO¹.

Sommaire [masquer]

- 1 Géographie
- 1.1 Lieux-dits et écarts
- 1.2 Communes limitrophes
- 2 Politique et administration
- 2.1 Tendances politiques et résultats
- 2.2 Liste des maires
- 2.3 Endettement
- 3 Histoire
- 3.1 Préhistoire et Antiquité
- 3.2 Moyen Âge
- 3.3 Renaissance
- 3.4 Période moderne
- 3.5 Période contemporaine
- 3.6 Seconde Guerre mondiale
- 4 Héraldique
- 5 Économie
- 6 Monuments
- 7 Transports
- 8 Personnalités liées à la commune
- 8.1 XIX^e siècle et contemporaine
- 8.2 XX^e siècle
- 8.3 XVII^e siècle
- 8.4 XVIII^e siècle
- 9 Démographie
- 10 Jurisprudence
- 11 Notes et références
- 11.1 Notes
- 11.2 Références
- 12 Voir aussi
- 12.1 Bibliographie
- 12.2 Articles connexes
- 12.3 Liens externes

Géographie [modifier | modifier le code]

Amboise s'étend, pour l'essentiel, sur la rive sud de la [Loire](#), à un endroit où la rivière se sépare en deux bras. Il enserré une île (dénommée île Saint-Jean, en souvenir d'un prieur matinal dépossédé de son île par les croisés). Grande île ou île-d'Or, qui fait partie du territoire communal.



La cité ancienne est surplombée par le plateau sur lequel est établi son château.

Amboise a été dévastée par les crues de la Loire, au sud comme au nord, par de très puissantes digues (promenade du lit, en forte surélévation par rapport aux premières maisons). La faubourg était sur le vaste le fut cantonné dans sa partie la plus élevée et défendu des hautes eaux par une digue (TTB).

Paris se trouve à 230 km.

Lieux-dits et écarts [modifier | modifier le code]

⚠ Cette section est vide, insuffisamment détaillée ou incomplète. Votre aide est la bienvenue !

Communes limitrophes [modifier | modifier le code]

Lussaud-sur-Loire, Saint-Martin-le-Beau, Dierre, La Croix-en-Touraine, Civray-de-Touraine, Souvigny-de-Touraine, Saint-Réigle, Chargé, et du côté nord de la Loire : Nazelles-Nérion, Poëze-sur-Cisse et Limay.



Politique et administration [modifier | modifier le code]

Tendances politiques et résultats [modifier | modifier le code]

Article connexe : [Élections municipales de 2014 en Indre-et-Loire](#).

⚠ Cette section est vide, insuffisamment détaillée ou incomplète. Votre aide est la bienvenue !

Liste des maires [modifier | modifier le code]

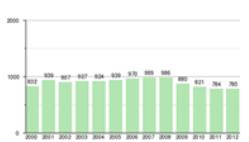
Période	Identité	Étiquette	Qualité
10 mai 1945 - 11 mai 1953	Émile Gounin		
11 mai 1953 - 2 juillet 1956	Maurice Mercier		
2 juillet 1956 - 19 mars 1959	Michel Débré		
19 mars 1959 - novembre 1992	André Chollet		
décembre 1992 - mars 2001	Bernard Débré		
mars 2001 - en cours	Christian Guyon		

Endettement [modifier | modifier le code]²

Évolution de l'endettement (en milliers d'€)³ :



Évolution de l'endettement par habitant (en €)⁴ :



Exemple : page wikipedia sur Amboise

Résultat du clic sur le lien interne "communes limitrophes"

Communes limitrophes [modifier | modifier le code]

Lussault-sur-Loire, Saint-Martin-le-Beau, Dierre, La Croix-en-Touraine, Civray-de-Touraine, Souvigny-de-Touraine, Saint-Règle, Chargé, et du côté nord de la Loire : Nazelles-Négron, Pocé-sur-Cisse et Limeray.

Nazelles-Négron
Pocé-sur-Cisse
Limeray
Lussault-sur-Loire
Amboise
Saint-Martin-le-Beau
Dierre, La Croix-en-Touraine
Chargé, Saint-Règle
Civray-de-Touraine

Politique et administration [modifier | modifier le code]

Tendances politiques et résultats [modifier | modifier le code]

Article connexe : Élections municipales de 2014 en Indre-et-Loire.
Cette section est vide, insuffisamment détaillée ou incomplète. Votre aide est la bienvenue !

Liste des maires [modifier | modifier le code]

Liste des maires			
Période	Identité	Étiquette	Qualité
18 mai 1945	11 mai 1953	Émile Gounin	
11 mai 1953	2 juillet 1966	Maurice Mercier	
2 juillet 1966	19 mars 1989	Michel Debré	
19 mars 1989	novembre 1992	André Chollet	
décembre 1992	mars 2001	Bernard Debré	
mars 2001	en cours	Christian Guyon	

Endettement [modifier | modifier le code]

Évolution de l'endettement (en milliers d'€)³ :

Évolution de l'endettement par habitant (en €)⁴ :

- Définir le point de branchement : l'*ancré* pour que le visiteur puisse sauter directement à la partie qui l'intéresse.

```
| <a id="#etiquette"/>
```

- Pour effectuer le lien

```
| <!-- depuis le même fichier -->
| <a href="#etiquette"> mon lien 1 </a>
| <!-- depuis 1 autre fichier de son site -->
| <a href="nomdefichier#etiquette"> mon lien 2 </a>
```

- Un lien peut être mis sur une image

```
| <a href="#etiquette">  </a>
```

On peut créer une ancre directement sur une balise autre que

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

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

Ex : une page avec des liens internes

```
<h1>Ma longue page</h1>
<p>
    Aller directement à :<br />
    <a href="#titre1">mon titre 1</a><br />
    <a href="#titre2">mon titre 2</a><br />
    <a href="#titre3">mon titre 3</a><br />
</p>
<h2 id="titre 1">mon titre 1</h2>
<p>... blablabla ...</p>

<h2 id="titre2">mon titre 2</h2>
<p>... blabla et rebla ...</p>

<h2 id="titre3">mon titre 3</h2>
<p>... blablabla ...</p>
```

- Lien vers une ancre située dans une autre page

```
| <a href="monautrepage.html#titre1"> autre page </a>
```

- Lien qui affiche une infobulle au survol

```
| Voici un nouveau lien avec une infobulle
| <a href="http://iut-blois.univ-tours.fr" title="IUT de Blois">vers l'IUT</a>
```

Affichage :

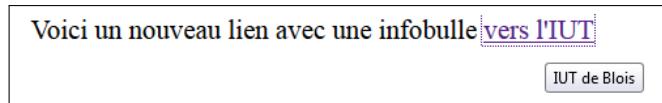


FIGURE 3.6 – iut de Blois

3.3 Les différentes zones d'une page web

Exemple d'agencement d'une page web



FIGURE 3.7 – zones d'une page web

3.3.1 En-tête et pied de page

illustration tirée de www.siteduzero.com

- L'en-tête peut contenir des images, liens, textes etc.
- plusieurs en-têtes possibles dans une page
 - ex : chaque section peut avoir son propre <header>.
 - Peut contenir un titre <h1>
- Définit avec <header>

```

1 | <header>
2 |   Le titre de la page web
3 | </header>

```

- le pied de page se trouve en général tout en bas du document.
- contient des informations : liens de contact, nom de l'auteur, mentions légales, etc.
- Définit avec <footer>

```

1 | <footer>
2 |   Pied de page<br />
3 |   Plan - Nous contacter - Copyright etc.
4 | </footer>

```

3.3.2 La navigation dans le site

- peut regrouper tous les principaux liens de navigation du site.
 - ex : menu principal d'un site
 - sous forme de liste à puces

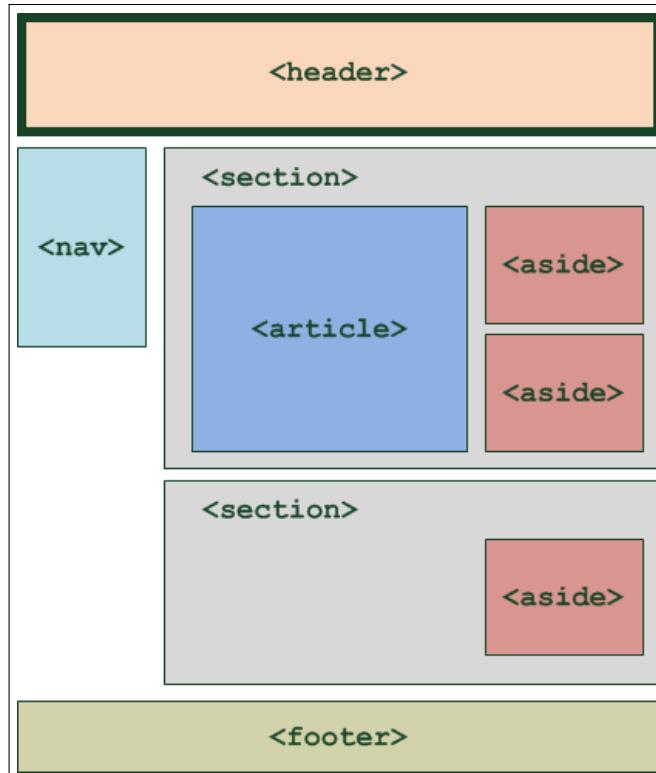


FIGURE 3.8 – en-tête

FIGURE 3.9 – pied de page

FIGURE 3.10 – navigation

- Définit avec <nav>

```

1 | <nav>
2 |   <ul>
3 |     <li><a href="index.html">Accueil</a></li>
4 |     <li><a href="forum.html">Forum</a></li>
5 |     <li><a href="contact.html">Contact</a></li>
6 |   </ul>
7 | </nav>

```

3.3.3 Les différents types de contenus

FIGURE 3.11 – structure

- Une *section* regroupe des contenus en fonction de leur thématique
 - Parties "texte" (le contenu) d'une page
 - Au centre de la page en général
- Définit avec <section>

```

1 | <section>
2 |   <h1>Ma section</h1>
3 |   <p>Bla bla bla bla</p>
4 | </section>

```

- Chaque section peut avoir son titre de niveau 1 (<h1>)

FIGURE 3.12 – structure d'une page web

- *informations complémentaires* au document que l'on visualise
 - généralement placées sur le côté
 - souvent dans une section
- Définit avec <aside>

```

1 | <aside>
2 |   Voici une autre info
3 | </aside>

```

Plusieurs blocs <aside> possibles dans une page.

FIGURE 3.13 – structure d'un document

- un *article indépendant* permet d'englober une portion généralement autonome de la page
 - partie de la page qui pourrait ainsi être reprise sur un autre site
 - souvent dans une section
- Définit par <article>

```

1 | <article>
2 |   <h1>Mon article</h1>
3 |   <p>Bla bla bla bla</p>
4 | </article>

```

3.3.4 Le comportement des balises HTML

2 types de comportement des balises HTML !

Comportement bloc

Crée automatiquement un retour à la ligne avant et après la balise.

Ces balises sont utilisées pour gérer l'apparence de la page web !

Quelques balises de type **bloc** :

```

1 | <p>, <h1>, <h2>, ...
2 | <footer>, <header>, ...
3 | <section>, <article>, ...
4 | etc.

```

balise universelle de **bloc** : <div>

Comportement inline

Obligatoirement à l'intérieur d'une balise bloc

Le texte s'écrit à la suite du texte précédent, sur la même ligne (donc *en ligne*)

ex :

```
1 | <em>, <strong>, <mark> ...
2 | <a>, <img/>
3 | etc.
```

balise universelle **inline** :

ex : bloc et inline

```
1 | <h1>Titre de mon paragraphe</h1>
2 | <p>Voici un <em>bloc</em> de paragraphe</p>
3 | <div>Voici un <strong>second</strong> bloc et
4 | une <span>balise</span> inline </div>
```



Notions utiles pour comprendre le fonctionnement des feuilles de style !

FIGURE 3.14 – Structure d'un document

```
1 | <!DOCTYPE html>
2 | <html>
3 |   <head>
4 |     <meta charset="utf-8"/>
5 |     <title>Ma page</title>
6 |   </head>
7 |   <body>
8 |     <header>
9 |       Mon entête
10 |     </header>
11 |     <nav>
12 |       <ul>
13 |         <li><a href="#">menu1</a></li>
14 |         <li><a href="#">menu2</a></li>
15 |         <li><a href="#">menu3</a></li>
16 |       </ul>
17 |     </nav>
18 |     <section>
19 |       <article>mon article</article>
20 |       <aside>A côté</aside>
21 |       <aside>Autre à côté</aside>
22 |     </section>
23 |     <section>
24 |       <aside>A côté</aside>
25 |     </section>
26 |     <footer>Mon pied de page</footer>
27 |   </body>
28 | </html>
```

Résultat de cette structure sans CSS



FIGURE 3.15 – Structure d'un document

3.4 Tableaux

3.4.1 Les tableaux

Balise <table>

- <caption> donne un titre au tableau
- <th> met des titres en haut des colonnes
- <tr> crée une ligne
- <td> crée une cellule dans la ligne

```

1 <table>
2   <caption style="caption-side:bottom">
3     Titre du tableau
4   </caption>
5   <tr>
6     <th>Titre 1</th><th>Titre 2</th>
7   </tr>
8   <tr>
9     <td>data 1,1</td><td>data 1,2</td>
10    </tr>
11   <tr>
12     <td>data 2,1</td><td>data 2,2</td>
13   </tr>
14 </table>

```

Titre 1 Titre 2

data 1,1 data 1,2

data 2,1 data 2,2

Titre du tableau

FIGURE 3.16 – tableau sans bordure

Exemple

```

1 <table>
2   <caption>Voici des données</caption>
3   <tr>
4     <td>data 1,1</td>
5     <td>data 1,2</td>
6   </tr>
7   <tr>
8     <td>data 2,1</td>
9   </tr>
10 </table>

```

Voici des données	
data 1,1	data 1,2
data 2,1	

FIGURE 3.17 – tableau

Exemple : Imbrication de tableaux

```

1 <table>
2   <tr>
3     <td>data 1,1</td>
4     <td>
5       <table>
6         <tr><td>data 1,2,1</td></tr>
7         <tr><td>data 1,2,2</td></tr>
8       </table>
9     </td>
10    </tr>
11    <tr>
12      <td>data 2,1</td>
13      <td>data 2,2</td>
14    </tr>
15 </table>

```

data 1,1	data 1,2,1
data 1,2,2	
data 2,1	data 2,2

FIGURE 3.18 – tableau dans un tableau

Fusionner les cellules de tableaux

2 attributs possibles :

- colspan fusionne des colonnes
- rowspan fusionne des lignes

```

1 <table>
2   <caption> Voici des données </caption>
3   <tr>
4     <td>data 1,1</td>
5     <td>data 1,2</td>
6   </tr>
7   <tr>
8     <td colspan="2">data 2,1</td>
9   </tr>
10 </table>
```

The diagram illustrates a table structure. It starts with a caption 'Voici des données'. Below it is a table with two rows. The first row contains two cells, each labeled 'data 1,1' and 'data 1,2' respectively. The second row contains a single cell spanning both columns, labeled 'data 2,1'.

FIGURE 3.19 – tableau avec colspan

3.4.2 Organisation des éléments du HTML5

Les balises sont aussi appelés **éléments**

- **Le flux (flow)** regroupe la plupart des éléments courants
- **Les métadonnées (metadata)** définissent des infos autour du contenu
ex : titre du document, style, relations externes, scripts
- **Le contenu sectionnant (sectioning)** définit les grandes zones du document HTML ou de l'application web
ex : blocs `<article>`, `<aside>`, `<nav>`, `<section>`
- **Les titres (heading)** : `<h1>` à `<h6>` et `<hgroup>`, affichés en bloc.
- **Le contenu de phrasé (phrasing content)** : éléments pouvant apparaître dans le flux de texte
 - affichés en-ligne `<audio>`, `<video>`, `<iframe>`, `<canvas>`, ``
 - éléments de formulaire `<input>`, `<textarea>`, `<button>`, `<select>`
 - balisage du texte ``, ``, `<i>`, ``, etc.
- **Le contenu embarqué(embedded)** : spécialisé
- **Le contenu interactif** : interaction avec l'utilisateur
ce qui peut être piloté au clavier/souris/doigt : les liens, les éléments média, les contrôles de formulaire

Troisième partie

CSS3

Chapitre 4

Les feuilles de style CSS

4.1 Introduction

4.1.1 Historique

- 1991 : création de HTML
- de 1991 à 1996 : mise en forme faite uniquement en HTML
 - ex : ``
- 1996 : création de CSS (Cascading Style Sheet = feuille de style en cascade)
- 1998 : CSS2, ajout de positionnement
- depuis 2000 : développement de CSS3, en partie mis en oeuvre par les navigateurs

4.1.2 Pourquoi ?

Des affichages différents ...



Captures d'écran 2013

Fonctionnalités CSS gérées par les différents navigateurs

source : <http://www.normansblog.de/demos/browser-support-checklist-css3/>

HTML permet de mettre en forme les pages d'un site web

MAIS ...

- Pour changer l'aspect du site, il faut :

Browser Rendering Engine	Firefox Gecko	Safari Webkit			Chrome Webkit	Internet Explorer Trident			Opera Presto		
Version	20	4	5	5.1+	26	6 – 8	9	10	11.1	11.6	12.1
Animations	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✓
Background Gradients	✓	✓	✓	✓	✓	✗	✗	✓	⚠	✓	✓
Background Size	✓	⚠	✓	✓	✓	✗	✓	✓	✓	✓	✓
Border Image	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
Border Radius	⚠	⚠	⚠	⚠	✓	✗	✓	✓	✓	✓	✓
Box Shadow	✓	⚠	✓	✓	✓	✗	✓	✓	✓	✓	✓
Columns	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓
Font Face	✓	✓	✓	✓	✓	⚠	✓	✓	✓	✓	✓
HSLA	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
Hyphens	✓	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Multiple Backgrounds	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
Opacity	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
RGBa	✓	⚠	⚠	✓	✓	✗	✓	✓	✓	✓	✓
Text Overflow	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Text Shadow	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓
Transforms	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
Transforms 3D	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗
Transitions	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓
Total CSS3 Support *	97.2%	83.4%	88.9%	97.2%	94.4%	8.3%	55.6%	94.4%	80.6%	83.3%	88.9%

FIGURE 4.1 – Fonctionnalités css selon les navigateurs

- Modifier toutes les pages
- Assurer le respect de la charte graphique du site !
- Les pages sont lentes à charger
- HTML ne permet de gérer :
 - les marges, indentations, distance entre lignes, polices

Création d'une page web HTML : Limitations

Page HTML sans CSS

```

1 <html>
2   <head> </head>
3   <body>
4     <h1>Exemple de code HTML</h1>
5     <h2>Premier sous-titre</h2>
6     <p>Voici un paragraphe<br>
7       et bla bla bla
8       bla<br> bla bla bla</p>
9     <h2>Deuxième sous-titre</h2>
10    <p>Voici un deuxième paragraphe<br>
11      et rebla bla bla</p>
12  </body>
13 </html>
```

Affichage

Solution : Page HTML avec feuille de style CSS

Code HTML (idem que le précédent)

```

1 <html>
2   <head>
3     <link rel="stylesheet"
4       type="text/css"
5       href="mon_css.css">
6   </head>
7   <body>
8   ...
9   </body>
10 </html>
```

Affichage

Code CSS

```

1 body{
2   color:white;
3   background-color:#11aaff;
4   font-family:sans-serif;
5 }
6 p{
7   font-size:12px;
8 }
```

Exemple de code HTML

Premier sous-titre

Voici un paragraphe
et bla bla bla bla bla
bla bla bla

Deuxième sous-titre

Voici un deuxième paragraphe
et rebla bla bla

FIGURE 4.2 – exemple sans css

FIGURE 4.3 – exemple de style css

```

9 | h1 {
10 |   font-size:20px;
11 |   font-style:italic;
12 |   font-weight:bold;
13 |   text-align:center;
14 | }
15 | h2 {
16 |   font-size:16px;
17 |   font-weight:bold;
18 | }
```

Affichage

Exemple de code HTML

Premier sous-titre

Voici un paragraphe
et bla bla bla bla bla
bla bla bla

Deuxième sous-titre

Voici un deuxième paragraphe
et rebla bla bla

FIGURE 4.4 – exemple de style css

Amélioration : code HTML avec feuille de style CSS et boîtes

Code HTML : ajout des blocs <div>

```

1 | <html>
2 |   <head>
3 |     <link rel="stylesheet"
4 |       type="text/css"
5 |       href="mon_css2.css">
6 |   </head>
7 |   <body>
8 |     <header>
9 |       <h1>Exemple de code HTML</h1>
10 |     </header>
11 |     <section>
12 |       <h2>Premier sous-titre</h2>
13 |       ...
14 |     </section>
15 |   </body>
16 | </html>
```

Affichage

code CSS : style des boîtes <div>

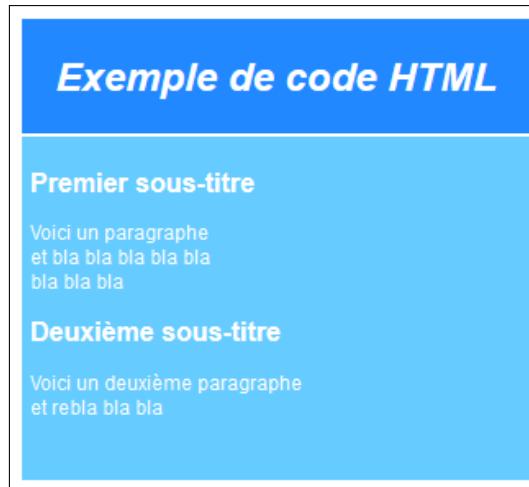


FIGURE 4.5 – exemple de style css

```

1 | .header {
2 |   padding:5px;
3 |   margin:1px;
4 |   width:300px;
5 |   background-color:#2288FF;
6 |   height:60px;
7 |   float:left;
8 |   text-align:center;
9 |
10| }
11| section {
12|   padding:5px;
13|   margin:1px;
14|   width:300px;
15|   background-color:#66ccFF;
16|   float:left;
17|   clear:both;
18| }
```

Affichage



FIGURE 4.6 – html sans css

Un autre exemple de style css pour la même page

```

1 | body{
2 |   font-family:cursive;
3 | }
4 | p{
5 |   font-size:12px;
6 | }
7 | h1{
8 |   font-size:24px;
9 |   font-style:italic;
10|  font-weight:bold;
11|  border:solid 2px grey;
12|  padding: 5px;
13| }
14| h2{
15|   font-size:16px;
16|   font-weight:bold;
```

```

17 |     margin:20px;
18 |
19 | header {
20 |     padding:5px;
21 |     margin:10px;
22 |     width:300px;
23 |     height:40px;
24 |     float:left;
25 |     text-align:center;
26 |
27 | section {
28 |     padding:5px;
29 |     margin:10px;
30 |     width:300px;
31 |     float:left;
32 |     clear:both;
33 |

```

Affichage

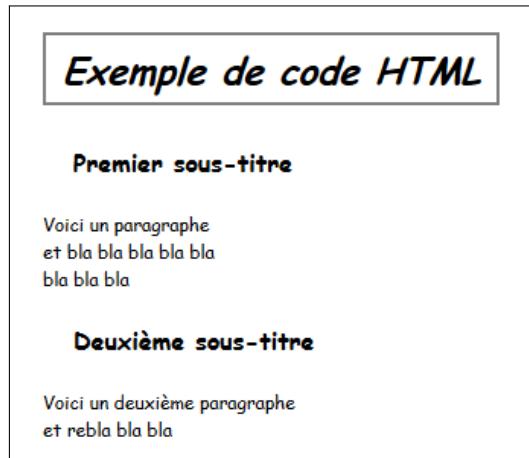


FIGURE 4.7 – Autre exemple de style css

- Séparation du contenu (HTML) et de la mise en forme (CSS).
 - Modifier facilement l'aspect d'un site sans en modifier le contenu
 - Cohésion de la présentation tout au long du site
 - Réduire le temps de chargement des pages
 - Correction de certains écueils d'HTML
 - Contrôle des polices, de la distance entre les lignes, des marges et des indentations
 - Moins de code, mieux structuré → sites plus faciles à maintenir et à créer.

Fichiers textes

- Éditeurs de css
- Ex : *top Style 3* gratuit !!!

4.1.3 Définition de propriétés des éléments HTML

Définir les propriétés des éléments HTML

```

selecteur {
    propriété1 : valeur1 ;
    propriété2 : valeur2 ;
    propriété3 : valeur3 ;
}

```

Sélecteur : le nom d'un élément HTML sur lequel on veut mettre un style

Propriété : ex : la couleur, la fonte, les bordures etc.

Ex :

```

1 | p {
2 |     color : red;
3 |     font-size : 12px;
4 |

```

4.1.4 Où spécifier les styles ?

3 méthodes

Dans un fichier .css à part, lié dans l'entête <head>

```
1 | <link rel="stylesheet" href="monstyle.css">
```

Dans un fichier .html, dans l'entête <head>

```
1 | <style type="text/css">
2 | h1{color:green}
3 | </style>
```

Dans n'importe quelle balise du code html

```
1 | <element style="...>
2 |
3 | <li style="color:red;">Voici une liste rouge</li>
```

— Entre /* */

— <!-- -- et --> acceptés (commentaire HTML), permet de cacher le CSS pour un navigateur qui ne les comprend pas !

Prise en charge des erreurs : **aucune** !

— Propriétés inconnues

— Valeurs invalides (unité !)

4.1.5 Compatibilité

Des sites listent les compatibilités de html5/css3 sur les différents navigateurs

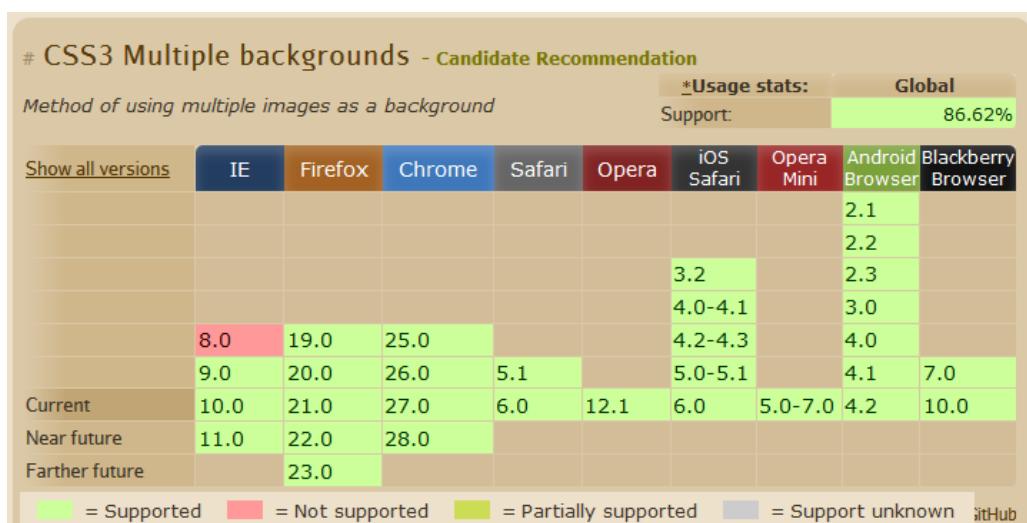


FIGURE 4.8 – Compatibilité du css

Source <http://caniuse.com/>

4.2 Mettre un style

4.2.1 Les couleurs et les fonds

Codes de Couleurs

Toute combinaison de chiffres entre #000000 et #FFFFFF est une couleur

color

Décrit la couleur du texte d'un élément.

S'applique à tous les éléments (p, body, span, div etc.)

Ex :

Nom	RGB HEX	RGB INTEGER
aqua	#00FFFF	rgb(0,255,255)
Black	#000000	rgb(0,0,0)
blue	#0000FF	rgb(0,0,255)
fuchsia	#FF00FF	rgb(255,0,255)
gray	#808080	rgb(128,128,128)
green	#008000	rgb(0,128,0)
lime	#00FF00	rgb(0,255,0)
maroon	#800000	rgb(128,0,0)
navy	#000080	rgb(0,0,128)
olive	#808000	rgb(128,128,0)
purple	#800080	rgb(128,0,128)
red	#FF0000	rgb(255,0,0)
silver	#C0C0C0	rgb(192,192,192)
teal	#008080	rgb(0,128,128)
white	#FFFFFF	rgb(255,255,255)
yellow	#FFFF00	rgb(255,255,0)



FIGURE 4.9 – Les couleurs de base

```

1 | p {
2 |   color: red;
3 | }
4 | body {
5 |   color: rgb(255,0,0);
6 |

```

background-color

Pour choisir la couleur d'arrière-plan d'un élément

background-image

Pour spécifier l'image d'arrière-plan d'un élément

```

1 | p {
2 |   background-color: yellow;
3 |

```

Texte avec un fond coloré

FIGURE 4.10 – texte avec un fond coloré

```

1 | p {
2 |   background-image: url(shadocks.jpg);
3 |

```



FIGURE 4.11 – image de fond

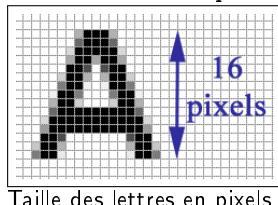
4.2.2 Les fontes (polices de caractères)

font-size

permet d'indiquer la taille de la fonte

- taille absolue : en px, cm, mm, pt
- taille relative : en % ou en em

ex : font-size: 16px;

**Attributs de fontes**

- font-family : par exemple Times, Arial, Verdana, serif
- font-style : normal | italic
- font-variant : normal | small-caps
- font-weight : normal | bold "bolder" ou "lighter"

text-decoration

text-decoration : overline | underline | line-through (rayé) | blink | none (par défaut).

text-align

text-align : left | right | center | justify vertical-align : top | middle | bottom
généralement pour les cellules de tableaux (selon navigateur)

Ex : balise strong stylée
Style css :

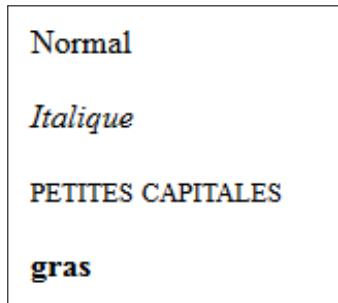


FIGURE 4.12 – attributs de fontes

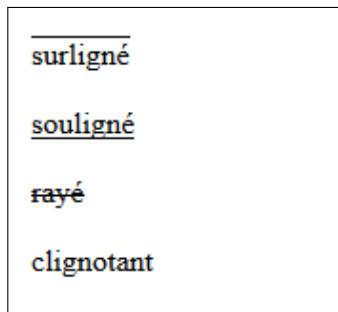


FIGURE 4.13 – décoration des textes

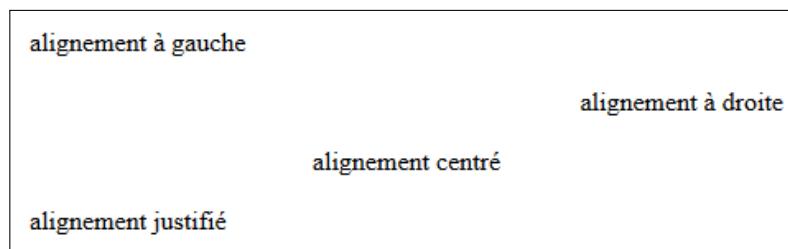


FIGURE 4.14 – alignement des textes

```

1 | strong {
2 |   font-family: Arial, sans-serif;
3 |   font-size: 20px;
4 |   font-weight: bold;
5 |   text-decoration: underline;
6 |   font-variant: small-caps;
7 |   color: #AA0055;
8 |
9 |
Code html :
1 | Pour faire <strong> un style important</strong>
```

Pour faire **UN STYLE IMPORTANT**

FIGURE 4.15 – un nouveau style pour strong

4.2.3 Les bordures et les marges

- Les éléments de type bloc peuvent avoir des marges et des bordures.

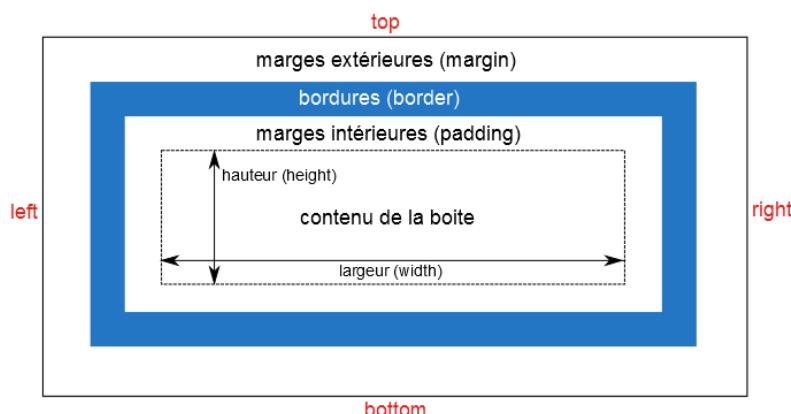


FIGURE 4.16 – description d'un bloc (boîte)

border

- border-color et border-[top, right, bottom, left]-color
ex : border-left-color: red;
- border-width et border-[top, right, bottom, left]-width
valeurs possibles : thin | medium | thick | une longueur (px, cm, etc.)
- border-style et border-[top, right, bottom, left]-style

```

1 | p {
2 |   border-style: solid;
3 |   border-width: thick;
4 |   border-color: green;
5 |
6 |   ou
7 |   p {
8 |     border: solid thick green;
9 | }
```

```

1 | <p>Voici un texte entouré</p>
2 | <p>et un autre</p>
```

Les types de bordures :

- none : pas de bordure (par défaut);
- et les autres ...

Autre exemple :

```
1 | <div>voici une petite boîte avec une bordure gauche</div>
```

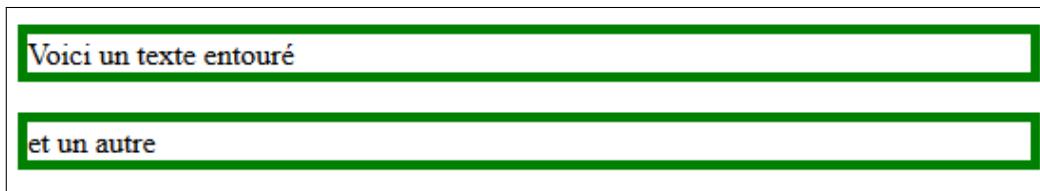


FIGURE 4.17 – exemple de bordures

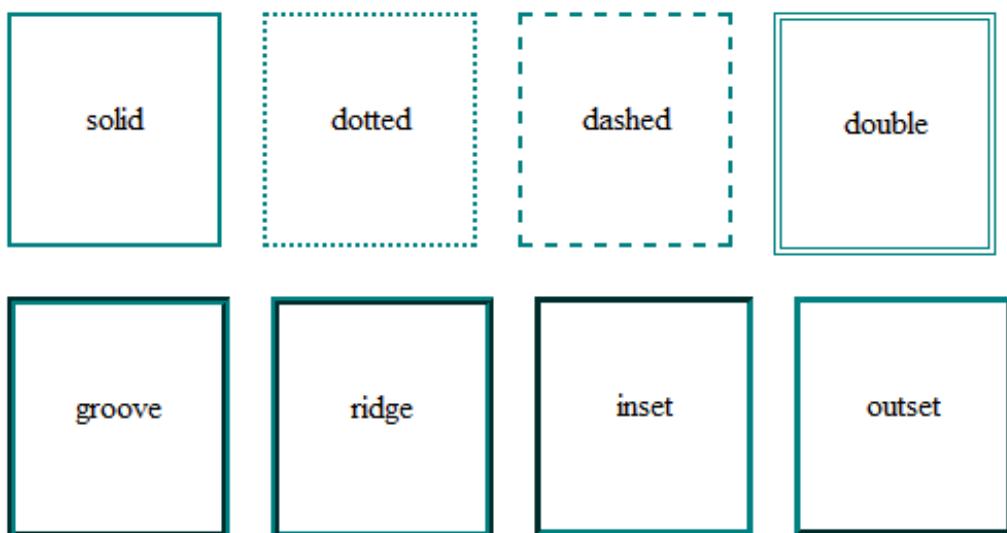


FIGURE 4.18 – les différents types de bordures

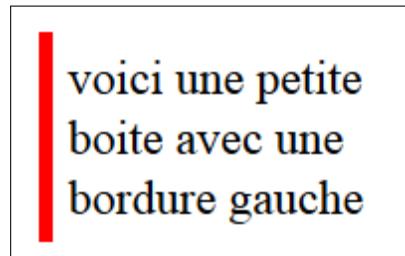


FIGURE 4.19 – boite avec une bordure à gauche

```

1 | div {
2 |   border-left-style: solid;
3 |   border-left-width: thick;
4 |   border-left-color: red;
5 |

```

margin
— margin-top, margin-right, margin-bottom, margin-left
— valeur : une longueur | un pourcentage (de la largeur de la boîte) | auto (calcul) | inherit

Ex : sans marge

```
1 | <p>un paragraphe sans marge</p>
```

```

1 | p {
2 |   background-color: lightgrey;
3 |

```



un paragraphe sans marge

FIGURE 4.20 – un paragraphe sans marge

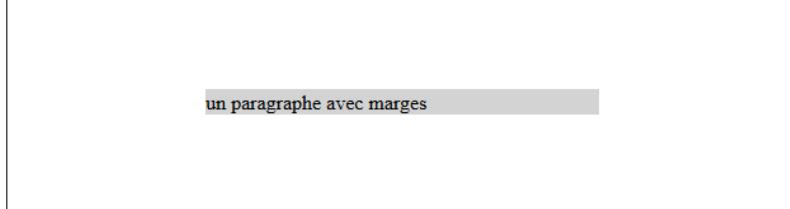
Autre exemple : avec marges

```

1 | p {
2 |   background-color: lightgrey;
3 |   margin-top: 50px;
4 |   margin-bottom: 50px;
5 |   margin-right: 150px;
6 |   margin-left: 150px;
7 |
8 | ou
9 | p {
10 |   background-color: lightgrey;
11 |   margin: 50px 150px 50px 150px;
12 |

```

```
1 | <p>un paragraphe avec marges</p>
```



un paragraphe avec marges

FIGURE 4.21 – style de paragraphe avec une marge

idem pour le padding (padding-top, padding-right, padding-bottom, padding-left)

4.3 Classes etc.

4.3.1 Héritage

Héritage

- Au sens général : transmission de quelque chose, d'une génération à une autre
- Au sens informatique : permet de transmettre les caractéristiques de certains "objets" (parent) vers un autre (enfant) qui en dérive
 - permet d'organiser les objets en structure hiérarchique

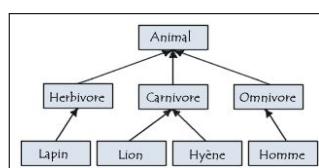


FIGURE 4.22 – Héritage

- Au sens des CSS : mécanisme qui permet, lors de la déclaration d'un nouveau style, d'y inclure les caractéristiques d'un autre

- Par défaut, un style appliqué à un élément html sera également appliquée aux éléments qu'il contient (enfants)

Ex :

- Style css :

```
1 | body { color: blue; }
```

- Code html :

```
1 | <body>
2 |   voici une page
3 |   <h1>avec un titre</h1>
4 |   <p>voici un paragraphe</p>
5 |   <p>et un <em>autre</em></p>
6 | </body>
```

— Le texte dans le corps (body) est en bleu y compris dans les balises h1, p, etc.

4.3.2 Les éléments

- Rappel : élément = balise HTML

— ex : p, span, div, body, h1 etc.

Regroupement d'éléments

- Ex : Les titres de niveaux 1, 2 et 3 sont écrits en rouge

```
1 | h1, h2, h3 {
2 |   color: red;
3 | }
```

- Ex : Les paragraphes et titres h1 ont une marge de 10px

```
1 | p, h1 {
2 |   margin: 10px;
3 | }
```

Sélecteurs contextuel

Mettre un *style particulier* sur une balise qui se trouve *dans une autre balise*

Ex : Tous les span dans des p apparaîtront en rouge souligné

Style css :

```
1 | p span {
2 |   color: red;
3 |   text-decoration: underline;
4 | }
```

Code html :

```
1 | <p>blabla et <span>bla</span></p>
2 | <span>bla et bla</span>
```

Affichage :

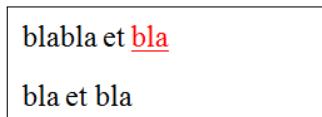


FIGURE 4.23 – application d'un style sur les span qui sont dans un p

Autre ex : tous les liens a contenus dans un paragraphe p seront gris

Style css :

```
1 | p a {
2 |   color: gray;
3 | }
```

Code html :

```
1 | <p> mon paragraphe avec <a>un lien</a> surtout !</p>
```

4.3.3 Les classes de style

Une classe

- permettent d'affecter des styles différents à des balises identiques
- Dans le style css

```

1 élément .maclasse {
2   propriété1: valeur1;
3   propriété2: valeur2;
4 }
```

- Dans le code html

```

1 <élément class="maclasse">
2   un texte avec du style
3 </élément>
```

Ex : Paragraphes p de classe special écrits en bleu centré

```

1 p.paragraphebleu{
2   color: blue;
3   text-align: center;
4 }

1 <p> bla bla bla et bla bla </p>
2 <p class="paragraphebleu"> bla bla bla et bla bla </p>
```

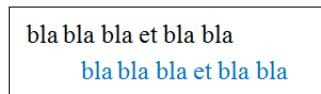


FIGURE 4.24 – application de la classe paragraphebleu

ex : style css

```

1 p.monstyle {
2   font-family: Arial;
3   font-size: 20pt;
4 }
```

Code html

```

1 <p class="monstyle">un paragraphe stylé</p>
2 <p>un paragraphe sans style</p>
```

Affichage :



FIGURE 4.25 – application de la classe monstyle

Plusieurs éléments peuvent utiliser une même classe de style :

ex :

```

1 h1.rouge, a.rouge {
2   color: red;
3 }

1 <h1>Titre</h1>
2 <h1 class="rouge">Titre rouge</h1>
3 <a>Lien</a>
4 <a class="rouge">Lien rouge</a>
```

Un élément HTML peut avoir plusieurs classes :

ex :

```

1 p.rouge {
2   color: red;
3 }
4 p.important {
5   font-weight: bold;
6 }
```



FIGURE 4.26 – application de la classe rouge sur h1 et a

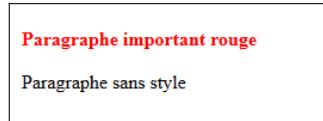


FIGURE 4.27 – application de 2 classes différentes (rouge et important) sur p

```

1 |<p class="rouge important">Paragraphe important rouge</a>
2 |<p>Paragraphe sans style</a>
```

Classe universelle

- On ne précise pas de balise
 - classe pouvant être utilisée dans n'importe quelle balise !
- Ex : Eléments de classe special2 écrits en blanc sur fond noir

```

1 | .fondnoir {
2 |   color:white;
3 |   background-color:black;
4 |   padding:10px;
5 |
6 | .bleu {
7 |   color: blue;
8 | }
```

Affichage :

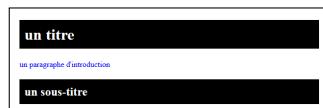


FIGURE 4.28 – classe bleu et fondnoir

```

1 |<h1 class="fondnoir">un titre</h1>
2 |<p class="bleu">un paragraphe d'introduction</p>
3 |<h2 class="fondnoir">un sous-titre</h2>
```

4.3.4 Les identifiants

id

- identifiant unique au sein de la page
 - fonctionne exactement de la même manière que class
- MAIS ne peut être utilisé qu'une fois dans le code d'une page html
- servent notamment à localiser des éléments HTML grâce à javascript

```

1 |<element id="monid">
2 |   Style pour cet élément
3 |</element>
```

Ex :

```

1 | #logo {
2 |   width:100px;
3 |   border:1px solid ;
4 |
5 | a#lien1 {
6 |   background-color: green;
7 | }
```

```

1 | 
2 | <a id="lien1">Mon lien</a>
```

Affichage :

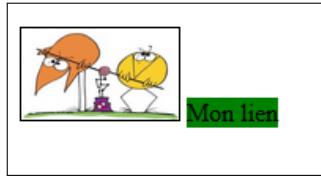


FIGURE 4.29 – style avec identifiant id="logo"

Ex : apparaît sur un fond jaune et encadré en bleu

```

1 | #fondjaune {
2 |   background-color:yellow;
3 |   border:blue 2px solid;
4 | }
```

Utilisation de ce style dans une page html :

Ex :

```

1 | <p id="fondjaune">
2 |   mon paragraphe fond jaune encadrement bleu
3 | </p>
```

Affichage :

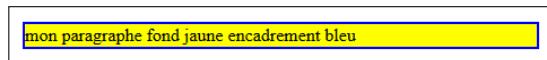


FIGURE 4.30 – style avec identifiant id="fondjaune"

4.3.5 Les pseudo-classes

- Permettent d'affiner le style appliqué à certaines balises
- En définissant une réaction à un événement
- Ou bien à la position relative de la balise au sein des autres balises.
- *Attention : le nom des pseudo-classes est prédéfini, il n'est donc pas possible de créer ses propres pseudo-classes*
- Différents types de pseudo-classes
 - dynamiques, de lien, de langue, first-child, de page ...
- Modifient le style d'une balise en fonction d'un événement (mouvement de la souris, clic, ou bien appui sur une touche du clavier)
- 3 types différents :
 - hover
 - focus
 - active
- *Attention : ne sont pas reconnues de la même façon par tous les navigateurs*

Les pseudo-classes dynamiques

- **hover** : Style lors d'un survol par le curseur de souris
 - ex :
- **focus** : style lorsque le focus lui est donné (ex : lors d'un clic dans une case de formulaire)
 - ex :
- **active** : style lorsque l'utilisateur clique sur l'élément (entre le moment où l'utilisateur clique sur le bouton de la souris et celui où il le relâche)
 - ex :

```
1 | a:active {color: #FF0000;}
```

Les pseudo-classes de lien

- link : style des liens hypertextes n'ayant pas encore été consultés

```
1 | a:link {color: green;}
```

- visited : style des liens hypertextes que le client a déjà visité

```
1 | a:visited {color: red;}
```

4.4 Positionnement

4.4.1 Comportements

Comportement inline

Les éléments se placent toujours les uns à côté des autres afin de rester dans le texte (ex : liens, images)

- seules les propriétés de mise en forme s'appliquent (police, couleurs)
- propriétés généralement héritées



FIGURE 4.31 – comportement des éléments en ligne

Comportement bloc

Les éléments occupent toute la largeur possible avec saut de ligne entre chacun (ex : div, p, h1, li etc.).

- on peut définir des marges et bordures

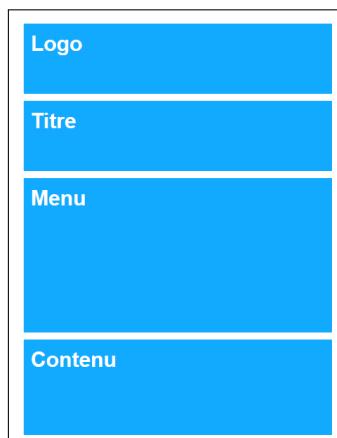


FIGURE 4.32 – comportement des éléments blocs

Une boîte a une marge extérieure (margin), un marge intérieur (padding) ...

Propriétés qu'on peut préciser sur un élément bloc !

width

largeur en pourcentage (%) ou pixels (px)

Ex : largeur en %

height

hauteur en pourcentage (%) ou pixels (px)

border

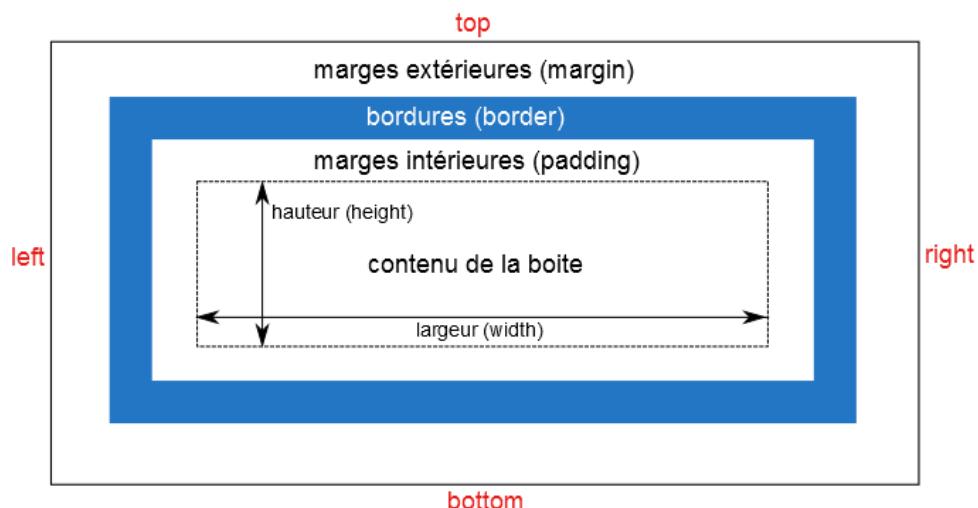


FIGURE 4.33 – Description d'une boîte

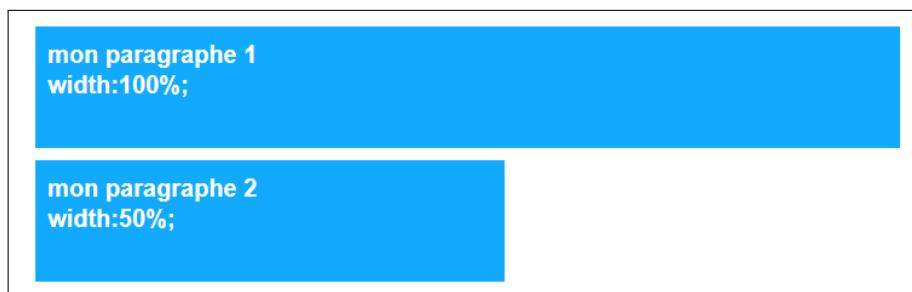


FIGURE 4.34 – largeur de paragraphe en %

Permet de spécifier le style de la bordure de la boîte (déjà vu !) border-top, border-right, border-bottom, border-left
 border-style : solid, dashed, ..., border-color ...
 ex : border : 10px solid #00FFAA ;

margin
 marges extérieures, en pixels (px)

padding
 marges intérieures

background-color
 Définit la couleur interne, valable sur le contenu, les marges intérieures, jusqu'au bordures

overflow
 — overflow: visible | hidden | scroll | auto ;
 — agit lorsque le contenu est plus grand que l'espace disponible pour l'élément

Ex :

```
1 | div {
2 |   width: 200px;
3 |   height: 100px;
4 |   overflow: scroll;
5 | }
```

Code html :

```
1 | <div>Voici un paragraphe avec scrollbar</div>
```

Affichage :

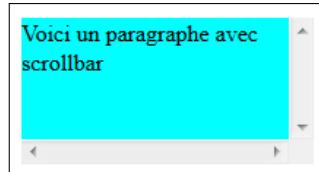


FIGURE 4.35 – paragraphe avec scrollbar

display
 Permet de forcer le comportement d'un élément

— 1 élément *inline* peut avoir un rendu *bloc* ou 1 élément *bloc* un rendu *inline*

Ex : bloc → inline

```
1 | li {
2 |   display: inline;
3 | }
```

Code html :

```
1 | <ul>
2 |   <li><a href="page1.html">HTML</a></li>
3 |   <li><a href="page2.html">CSS</a></li>
4 |   <li><a href="page3.html">JavaScript</a></li>
5 | </ul>
```

Affichage sans le css :



FIGURE 4.36 – menu sans le css

Affichage avec le css :
 Ex : inline → bloc

```
1 | strong {
2 |   display: block;
3 | }
```

HTML CSS JavaScript

FIGURE 4.37 – menu inline grâce au css

Voici une balise en ligne qui s'affiche en bloc

FIGURE 4.38 – balise strong sans le css

Code html

```
1 | Voici <strong> une balise </strong> <strong> en ligne </strong> qui s'affiche en <strong> bloc </strong>
```

Affichage sans le css :

Affichage avec le css :

**Voici
une balise
en ligne
qui s'affiche en
bloc**

FIGURE 4.39 – balise strong devenue block avec le css

2 balises universelles = sans aucune signification particulière !

- : balise inline à placer dans paragraphe de texte, pour sélectionner certains mots uniquement.
- <div> </div> : balise bloc, qui entoure un bloc de texte. Utilisée pour créer des boîtes !

Positionnement des boîtes/blocs

Propriété position

- relative : par rapport à sa position calculée
- absolute : par rapport au document ou à un parent

Propriétés à éviter car difficile à gérer complètement !

Coordonnées de positionnement : top, bottom, left, right en pixels (px)

Meilleure solution

définir correctement les agencements à l'aide de marges et d'éléments flottants, pour mieux s'adapter au poste client

Ex : Position absolue

```
1 | h2 {  
2 |   position: absolute;  
3 |   left: 100px;  
4 |   top: 150px;  
5 | }
```

Code html :

```
1 | <h2>mon titre</h2>  
2 | <p>avec la position absolue, un élément peut être placé n'importe où sur la page</p>
```

Affichage sans le css :

Affichage avec le css :

Principalement en fonction de la définition d'affichage.

- *largeur* : Choisir la plus petite largeur d'écran supportée par le plus grand nombre
- *Longueur* : ne pas dépasser 3 à 5 hauteurs d'écran.
- Autre solution : Déetecter la résolution des visiteurs à l'aide d'un script javascript, et rediriger le visiteur vers une page de la bonne largeur.

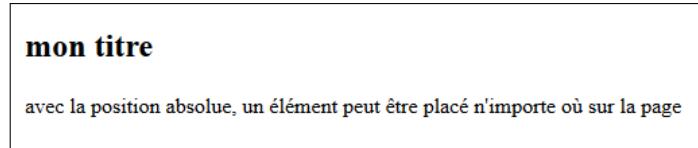


FIGURE 4.40 – position sans le css

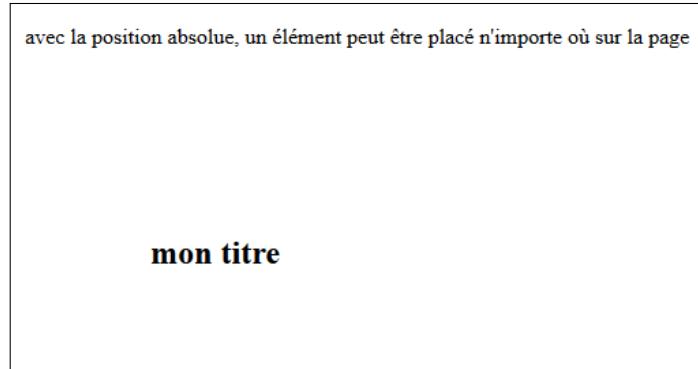


FIGURE 4.41 – position absolue d'un h2

4.4.2 Les grilles

- Grille = ensemble de lignes horizontales et verticales qui se croisent !
 - Les éléments sont placés sur la grille en fonction de ces rangées/lignes et colonnes
 - Hauteur et largeur des rangées et colonnes à taille fixe ou variable
 - Placement des éléments grâce au numéro ou nom de la ligne/colonnes
 - Il y a un algorithme de placement automatique pour les éléments non placés
 - Création de colonnes/rangées automatique pour du contenu
 - Contrôle de l'alignement des éléments
 - Contrôle des contenus qui se chevauchent
 - le point de départ d'une grille est le conteneur.
- Les grilles comportent généralement :
 - de repères (axes) verticaux et/ou horizontaux, séparés par des gouttières
 - de marges externes (pour les éventuels bords de la fenêtre) et internes (de chaque axe).
- Les grilles verticales

On peut choisir le nombre de colonnes à insérer dans une largeur fixe.
- Les grilles verticales et horizontales

Ligne : verticale (ligne de grille de colonnes) ou horizontale (ligne de grille de rangées)
Piste (ou plage) de grille : espace situé entre 2 lignes de grille adjacentes (ce sont les colonnes ou les rangées de la grille)
Cellule : espace situé entre 2 lignes de rangée et 2 lignes de colonne **zone** : espace entouré par 4 lignes de grille
- Agencement des grilles (blocs)
 - Après avoir choisi 1 type de grille, il faut la "remplir" en disposant les blocs au bon endroit (images, textes, etc.).
 - Exemples de combinaisons :
- Exemple de page html :

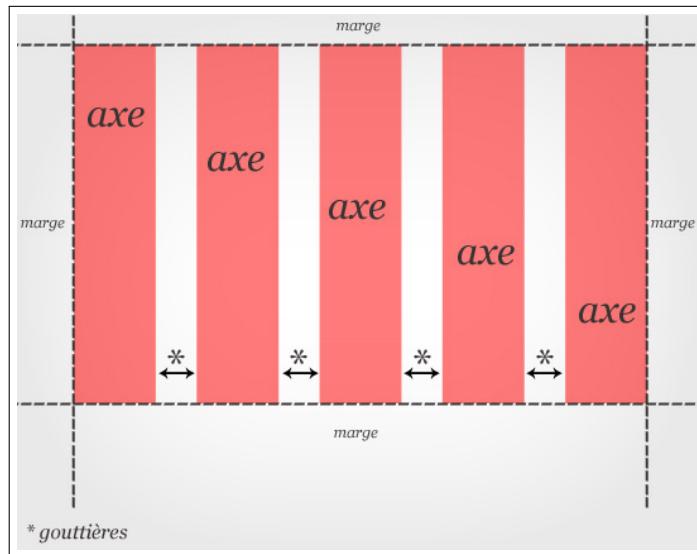
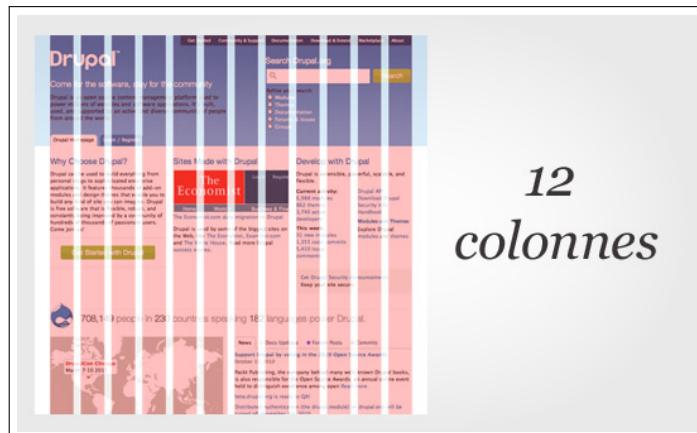
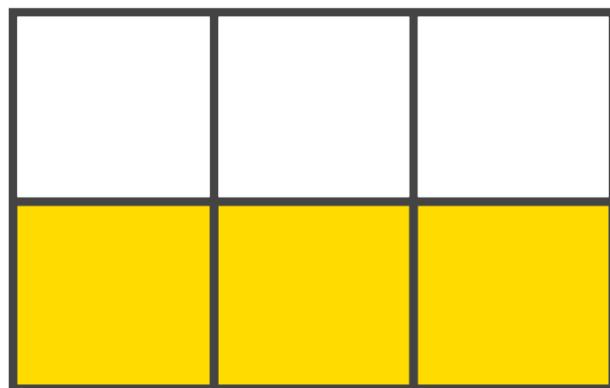
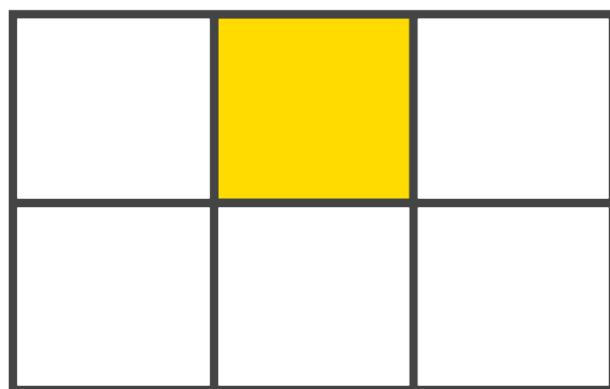
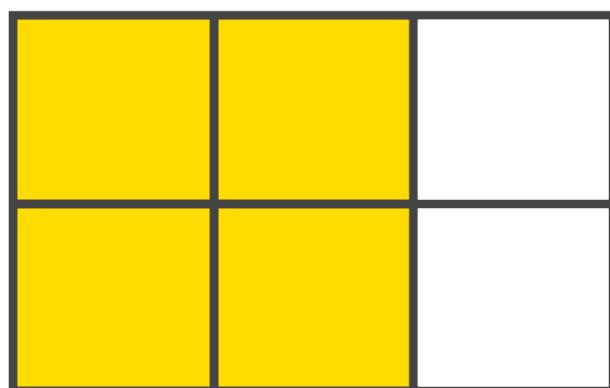
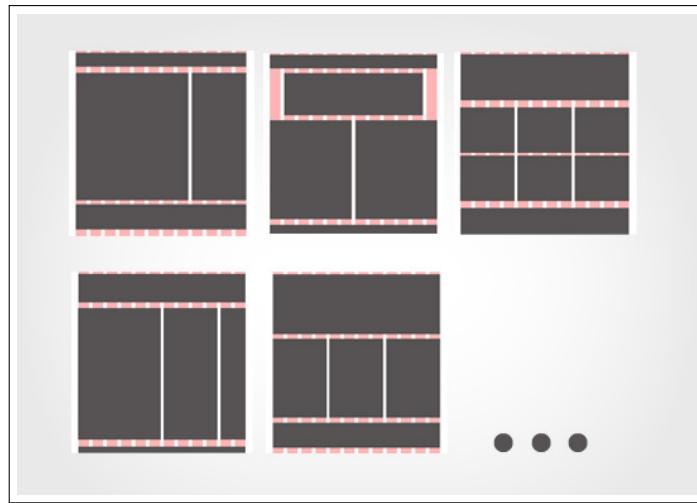
FIGURE 4.42 – Anatomie d'une grille, source : <https://www.alsacreations.com>FIGURE 4.43 – Colonnes d'une grille, source : <https://www.alsacreations.com>FIGURE 4.44 – Rangées et colonnes, source : <https://www.alsacreations.com>

FIGURE 4.45 – Ligne, source : <https://la-cascade.io>FIGURE 4.46 – Piste, source : <https://la-cascade.io>FIGURE 4.47 – Cellule, source : <https://la-cascade.io>FIGURE 4.48 – zone, source : <https://la-cascade.io>

FIGURE 4.49 – Agencement d'une grille, source : <https://www.alsacreations.com>

```

1 <div class="wrapper">
2   <header>L'en-tête</header>
3   <nav>
4     <ul>
5       <li><a href="">Titre 1</a></li>
6       <li><a href="">Titre 2</a></li>
7       <li><a href="">Titre 3</a></li>
8     </ul>
9   </nav>
10  <article>
11    <h1>Titre 1</h1>
12    <p>
13      bla bla bla ...
14    </p>
15  </article>
16  <aside>Infos 1</aside>
17  <aside>Infos 2</aside>
18  <aside>Pub</aside>
19  <footer>Pied de page</footer>
20 </div>

```

— Le conteneur (wrapper = "enveloppeur")

- est créé par la propriété `display:grid` ou `display:inline-grid` sur un élément
- Dans le css, `.wrapper` est déclaré comme conteneur de grille !

```

1 .wrapper {
2   display: grid;
3 }

```

— Affichage de l'exemple (avec l'inspecteur de grille de firefox) :

FIGURE 4.50 – Conteneur de grille

— Les pistes

- grid-template-columns et grid-template-rows pour définir des colonnes et des rangées

```

1 | .wrapper {
2 |   display: grid;
3 |   grid-template-columns: 100px 200px 100px;
4 |

```

Affichage avec 3 colonnes :

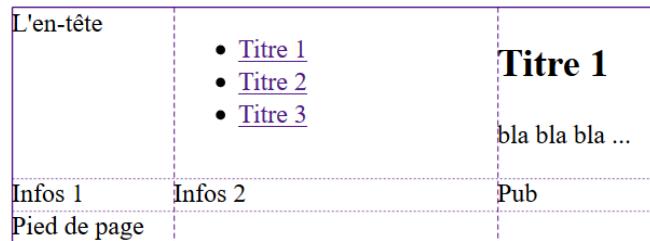


FIGURE 4.51 – Conteneur de grille

- L'unité fr : fr = fraction de l'espace disponible dans le conteneur de la grille, permet d'avoir des grilles flexibles en fonction de l'espace disponible

— Ex :

```

1 | .wrapper {
2 |   display: grid;
3 |   grid-template-columns: 1fr 2fr 1fr;
4 |

```



FIGURE 4.52 – Conteneur de grille

- **repeat()** : permet de répéter des pistes

grid-template-columns: 1fr 1fr 1fr; peut s'écrire grid-template-columns: repeat(3, 1fr);

- ex : 1 colonne de 20px de large, puis 6 fois une piste de 1fr, et 1 colonne de 20px de large
- *On peut utiliser des dimensions absolues et relatives en même temps !*

```

1 | .wrapper {
2 |   display: grid;
3 |   grid-template-columns: 20px repeat(6, 1fr) 20px;
4 |

```

- grid-auto-rows et grid-auto-columns : Si un élément est placé en dehors de la grille définie avec des grid-template..., la grille ajoute alors des colonnes ou rangées automatiquement

- minmax() : pour donner une taille minimum, qui s'agrandit pour s'adapter au contenu

- ex : rangées au min. 50px et au max. auto

```

1 | .wrapper {
2 |   display: grid;
3 |   grid-template-columns: 1fr 2fr 1fr;
4 |   grid-auto-rows: minmax(50px, auto);
5 |

```

Exemple de code html

```

1 | <div class="wrapper">
2 |   <header class="tete">L'en-tête</header>
3 |   <nav class="menu">
4 |     <ul>
5 |       <li><a href="#">Titre 1</a></li>
6 |       <li><a href="#">Titre 2</a></li>
7 |       <li><a href="#">Titre 3</a></li>
8 |     </ul>
9 |   </nav>
10 |

```



FIGURE 4.53 – rangées auto

```

11 | <section class="partie1">
12 |   <article class="art1">
13 |     <h1>Titre 1</h1>
14 |     <p> bla bla bla ... </p>
15 |   </article>
16 |   <aside class="info1">Infos 1</aside>
17 |   <aside class="info2">Infos 2</aside>
18 |   <aside class="pub">Pub</aside>
19 | </section>
20 | <footer class="pied">Pied de page</footer>
21 | </div>
22 |

```

- Placement des zones en fonction des lignes de la grille avec `grid-column-start`, `grid-column-end`, `grid-row-start`, `grid-row-end`

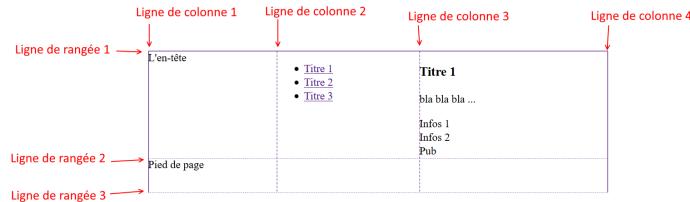


FIGURE 4.54 – les lignes de rangées et les lignes de colonnes

- ex : Placement de l'en-tête et du menu

```

1 | .wrapper { ... }
2 | .tete {
3 |   background-color:yellow;
4 |   grid-column-start: 2;
5 |   grid-column-end: 4;
6 |   grid-row-start: 1;
7 | }
8 | .menu {
9 |   background-color:lightgray;
10|   grid-column-start: 1;
11|   grid-row-start: 1;
12|   grid-row-end: 3;
13| }

```

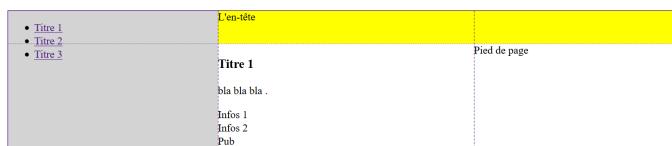


FIGURE 4.55 – placement

- on ajoute le placement de la section et du pied de page

```

1 | ...
2 | .partie1 {
3 |   grid-column-start: 2;
4 |   grid-column-end: 4;
5 |   grid-row-start: 2;
6 | }
7 | .pied {
8 |   background-color:yellow;
9 |   grid-column-start: 1;
10|   grid-column-end: 3;
11|   grid-row-start: 3;
12|   grid-row-end: 3;
13| }

```

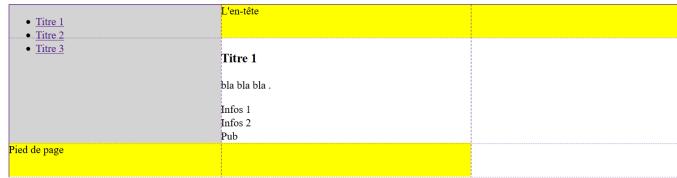


FIGURE 4.56 – placement

- Les gouttières = espace entre les colonnes ou les rangées
- `grid-column-gap` et `grid-row-gap`

```

1 | .wrapper {
2 |   display: grid;
3 |   grid-template-columns: 1fr 2fr 1fr;
4 |   grid-auto-rows: minmax(50px, auto);
5 |   grid-column-gap: 20px;
6 |   grid-row-gap: 5px;
7 | }
```

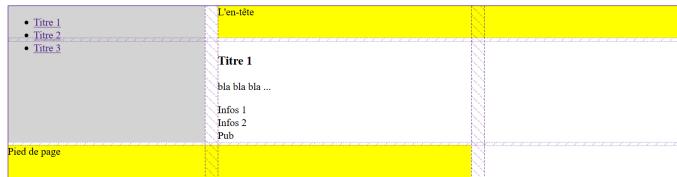


FIGURE 4.57 – placement

- Grilles imbriquées
- 1 grille imbriquée n'est pas liée à la grille qui la contient
- Elle n'hérite pas des `grid-gap` ni des lignes de la grille parent donc les lignes ne s'alignent pas.

```

1 | .partie1 {
2 |   ...
3 |   display: grid;
4 |   grid-template-columns: 3fr 1fr;
5 |   grid-template-rows: 2fr 1fr 1fr;
6 |   grid-gap: 5px;
7 | }
8 | ...
```

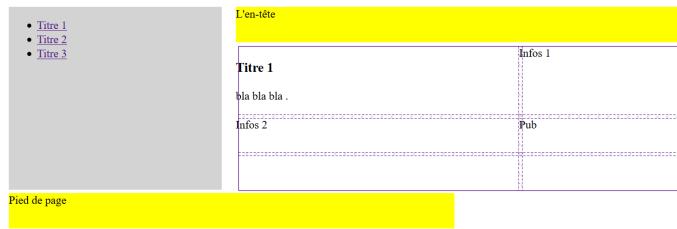


FIGURE 4.58 – placement

- Affichage final (avec la grille `.partie1` apparente)

- Un autre moyen de placer les zones : `grid-area`

on donne un nom à chacune des zones, par exemple `tt` pour `tete`, `mn` pour `menu`, `pt1` pour `partie`, `ft` pour `pied`!

```

1 | .tete {
2 |   background-color: yellow;
3 |   grid-area: tt;
4 | }
5 | .menu {
6 |   background-color: lightgray;
7 |   grid-area: mn;
8 | }
9 | .partie1 {
```

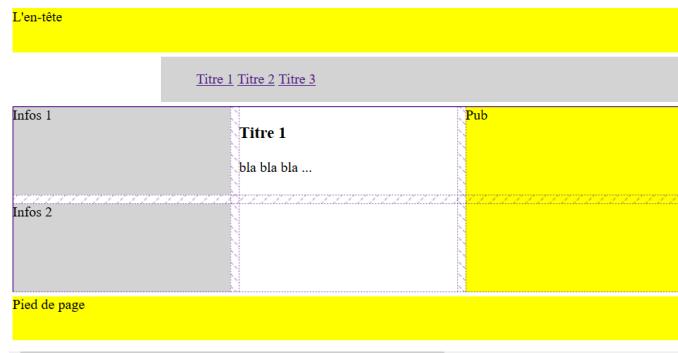


FIGURE 4.59 – placement

```

10 |   grid-area: pt1;
11 |   ...
12 | }
13 | .pied {
14 |   background-color:yellow;
15 |   grid-area: ft ;
16 |

```

— Un autre moyen de placer les zones : grid-area

Puis on modifie le wrapper en indiquant où se trouve chaque zone dans les rangées et colonnes !

Ici, on a créé 8 colonnes et des rangées automatiques.

```

1 | .wrapper {
2 |   display: grid;
3 |   grid-template-columns: 1fr 1fr 1fr 1fr 1fr 1fr 1fr 1fr;
4 |   grid-auto-rows: minmax(50px, auto);
5 |   grid-column-gap: 20px;
6 |   grid-row-gap: 5px;
7 |   grid-template-areas:
8 |     "tt tt mn mn mn mn mn"
9 |     "tt tt pt1 pt1 pt1 pt1 pt1 pt1"
10 |    "tt tt pt1 pt1 pt1 pt1 pt1 pt1"
11 |    "tt tt pt1 pt1 pt1 pt1 pt1 pt1"
12 |    "ft ft ft ft ft ft ft ft";
13 |

```

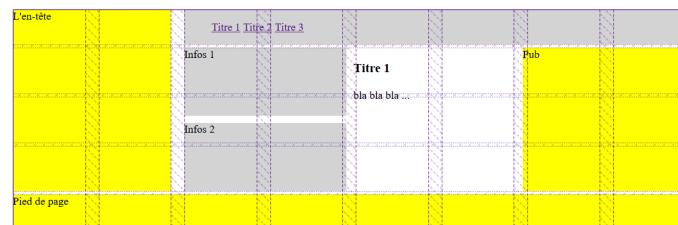


FIGURE 4.60 – Résultat avec la grille apparente

— Résultat sans la grille



FIGURE 4.61 – Résultat avec la grille apparente

— On peut utiliser les grilles pour créer des tableaux, agencer les images et le texte d'une manière spéciale etc.

4.5 RWD

4.5.1 Qu'est-ce que le Responsive Web Design ?

Responsive Web Design (RWD) = Web adaptatif, conception d'une interface auto-adaptable.

Technologie CSS3 media queries

Extension de la règle @media pour adapter la mise en page du site suivant l'outil utilisé pour le consulter.

ex : ordinateur, tablette, smartphone, télévision, imprimante ...

- Permet d'utiliser des règles CSS différentes
en fonction des caractéristiques du terminal de consultation

Mobile First (le mobile en premier)

Construction d'une interface pour terminaux mobiles avant celle pour un ordinateur

- Contrainte + forte : faible espace d'affichage
=> épurer au maximum les éléments visibles par l'utilisateur final.



FIGURE 4.62 – Adaptation du site selon le support

<http://www.paulolyslager.com/responsive-design-hype-solution/>

4.5.2 Les media queries

- Utiliser la balise @media
- ex : si le media est une imprimante :
 - ne pas imprimer les parties nav, aside et footer
 - texte du corps de la page en noir

```

1 | @media print {
2 |   nav, aside, footer {
3 |     display:none;
4 |   }
5 |   body {
6 |     color:black;
7 |   }
8 |

```

Des opérateurs logiques

- and, only, not, "ou" obtenu en listant différentes media queries à la suite, séparées par des virgules

Ex : pour les écrans de largeur < à 640 pixels (max-width : 640px)
 Code html :

```
1 | <link rel="stylesheet" media="screen and (max-width: 640px)" href="smallscreen.css" type="text/css"/>
```

Style css :

```
1 | @media screen and (max-width: 640px) {  
2 |   .bloc {  
3 |     display:block;  
4 |     clear:both;  
5 |   }  
6 | }  
7 | }
```

Autres exemples :

```
1 | /* Sur tous types d'écran, quand la largeur de la fenêtre est comprise entre 1024px et 1280px */  
2 | @media all and (min-width: 1024px) and (max-width: 1280px) { ... }  
3 |  
4 | /* Sur les téléviseurs */  
5 | @media tv { ... }  
6 |  
7 | /* Sur tous types d'écrans orientés verticalement */  
8 | @media all and (orientation: portrait) { ... }
```

Fonctionnalités

peuvent être préfixés par min- et max- pour définir des valeurs minimales ou maximales à respecter.

- aspect-ratio : ratio du périphérique de sortie (par exemple 16/9)
- resolution : résolution du périphérique (en dpi, dppx, ou dpcm)
- height : hauteur de la zone d'affichage (fenêtre).
- width : largeur de la zone d'affichage (fenêtre).
- orientation : orientation du périphérique (portrait ou paysage).
- media : type d'écran de sortie. Quelques-unes des valeurs possibles :
 - screen : écran "classique";
 - handheld : périphérique mobile;
 - print : impression;
 - tv : télévision;
 - projection : projecteur;
 - all : tous les types d'écran.

Exemple :

```
1 | p { color: blue; }  
2 |  
3 | @media screen and (max-width: 1024px) {  
4 |   p { color: white;  
5 |     background-color: gray;  
6 |     font-size: 14pt; }  
7 | }
```

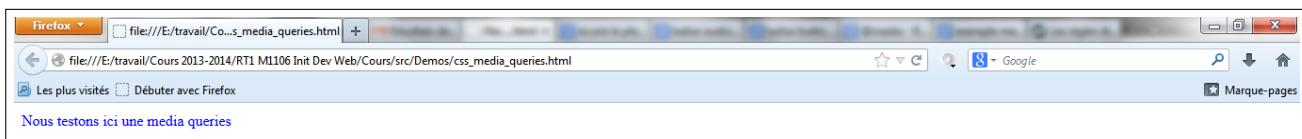


FIGURE 4.63 – affichage si largeur > 1024 px



FIGURE 4.64 – affichage si largeur < 1024 px

- Adaptation facile pour différentes résolutions
- Redéfinition de la position des objets sur la grille et/ou de la grille elle-même
- Rappel du code html de l'exemple précédent :

```

1 | ...
2 | <body>
3 | <div class="wrapper">
4 |   <header class="tete">L'en-tête</header>
5 |   <nav class="menu">
6 |     <ul>
7 |       <li><a href="">Titre 1</a></li>
8 |       <li><a href="">Titre 2</a></li>
9 |       <li><a href="">Titre 3</a></li>
10 |     </ul>
11 |   </nav>
12 |   <section class="partie1">
13 |     <article class="art1">
14 |       <h1>Titre 1</h1>
15 |       <p> bla bla bla ... </p>
16 |     </article>
17 |     <aside class="info1"> Infos 1</aside>
18 |     <aside class="info2"> Infos 2</aside>
19 |     <aside class="pub">Pub</aside>
20 |   </section>
21 |   <footer class="pied">Pied de page</footer>
22 | </div>
23 | ...

```

- On souhaite avoir un style différent selon qu'on ait sur un écran "large" ou un écran "mobile" type smartphone
- On réfléchit d'abord à ce qui va être **commun** aux 2 types d'écrans !
 - les zones **tete**, **menu**, **pied**
 - la "décoration" des zones **art1**, **info1**, **info2**, **pub**
 - les **grid-area** de la "grille" seront valables pour les 2 présentations
 - par contre, les **grid-area** de la "sous-grille" **.partie1** sont supprimés car ils ne sont pas communs

```

1 | .tete {
2 |   background-color:yellow;
3 |   grid-area: tt;
4 | }
5 | .menu {
6 |   background-color:lightgray;
7 |   grid-area: mn;
8 | }
9 | .art1{ }
10|.info1{ background-color:lightgray; }
11|.info2{ background-color:lightgray; }
12|.pub{ background-color:yellow; }
13|.pied{ background-color:yellow; grid-area: ft; }

```

- On définit ensuite ce qui va être la règle à appliquer sur les écrans de taille supérieure à 500px.
- On ajoute dans le css la règle : @media all and (min-width: 500px) ...
- Cette règle contient le **.wrapper** et les **.partie1**, **.art1**, etc. qui seront différents selon l'écran

```

1 | @media screen and (min-width: 500px){
2 |   .wrapper {
3 |     display: grid;
4 |     grid-template-columns: 1fr 1fr 1fr 1fr 1fr 1fr 1fr 1fr;
5 |     grid-auto-rows: minmax(50px, auto);
6 |     grid-column-gap: 20px;
7 |     grid-row-gap: 5px;
8 |     grid-template-areas:
9 |       "tt tt mn mn mn mn mn"
10|.info1{ background-color:lightgray; }
11|.info2{ background-color:lightgray; }
12|.pub{ background-color:yellow; }
13|.pied{ background-color:yellow; grid-area: ft; }
14 |   }
15 |   .partie1 {
16 |     grid-area: pti;
17 |     display:grid;
18 |     grid-auto-columns: minmax(100px, auto);
19 |     grid-auto-rows: minmax(100px, auto);
20 |     grid-gap: 10px;
21 |   }
22 |   .art1{ grid-area: 1 / 2 / 3 / 3; }
23 |   .info1{ grid-area: 1 / 1 / 2 / 2; }
24 |   .info2{ grid-area: 2 / 1 / 3 / 2; }
25 |   .pub{ grid-area: 1 / 3 / 3 / 4; }
26 | }

```

- Enfin on définit ce qui va être la règle à appliquer sur les écrans mobiles
 1. On met @media handheld ...
 2. On fait une copie du **.wrapper** et on le modifie : 1fr = 1 colonne dans la grille
 3. On modifie le **grid-template-areas** en 1 colonne

```

1 | @media handheld{
2 |   .wrapper {
3 |     display: grid;
4 |     grid-template-columns: 1fr;
5 |     grid-auto-rows: minmax(20px, auto);
6 |     grid-row-gap: 5px;

```



FIGURE 4.65 – affichage pour smartphone

```

7 |   grid-template-areas:
8 |   "tt"
9 |   "mn"
10 |   "ai"
11 |   "i1"
12 |   "i2"
13 |   "pub"
14 |   "pt1"
15 |   "ft";
16 |
17 | .art1{ grid-area: ai; }
18 | .info1{ grid-area: i1; }
19 | .info2{ grid-area: i2; }
20 | .pub{ grid-area: pb; }
21 |

```



Titre 1

bla bla bla ...



FIGURE 4.66 – affichage pour smartphone

Quatrième partie

Les réseaux, Internet, le web ... ?

Chapitre 5

Internet et le Web

5.1 D'où vient Internet ?

5.1.1 Les premiers réseaux

Internet, synonymes : web, réseau, net, toile, cyberespace...

- Au départ, réseaux locaux : ordinateurs reliés par câbles
- 1962 : Connexion de réseaux locaux (Licklider, DARPA)
- 1969 : réseau ARPANET connection à 4 universités : UCLA, UC Santa Barbara, SRI, and Utah.

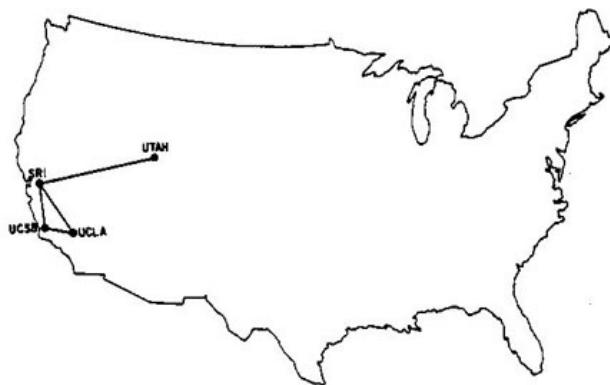


FIGURE 5.1 – Arpanet en 1969

- 1977 : Arpanet s'est étendu
- Source : <http://classes.design.ucla.edu/Spring06/161A/projects/camile/arpanet/>
- 1975 : X.25, standard de télécommunication
 - A la fin des années 80 : l'Europe, l'Asie, l'Australie se connectent !

5.1.2 Les années 80 en France

- Années 80, service de télématique : minitel
 - Ecran : texte, 25 lignes * 40 colonnes, 8 nuances de gris
 - Jeu de caractères : images en mode mosaïque, art ASCII.
 - Equivalents étrangers : *NAPLPS* (USA), *Prestel* (UK), *Ceefax* (UK), *Bildschirmtext* (DE)
 - 3611 (annuaire tel), 3615 (kiosque de services)
- 2009 : plus que 107 connexions, 106 utilisateurs
- Septembre 2011 : Fin de service

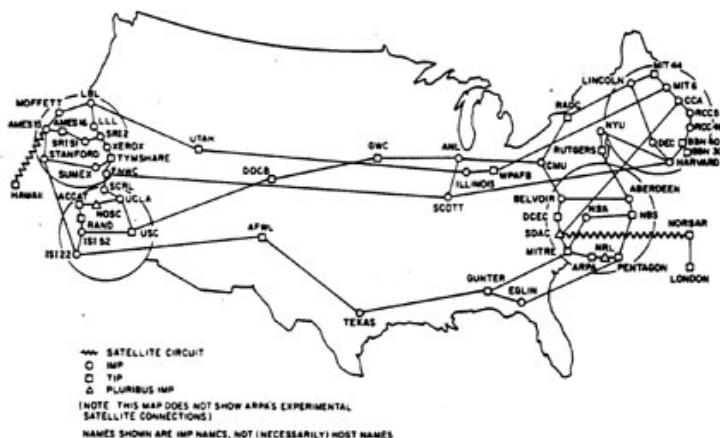


FIGURE 5.2 – Arpanet en 1977

FIGURE 5.3 – html sans css



FIGURE 5.4 – minitel

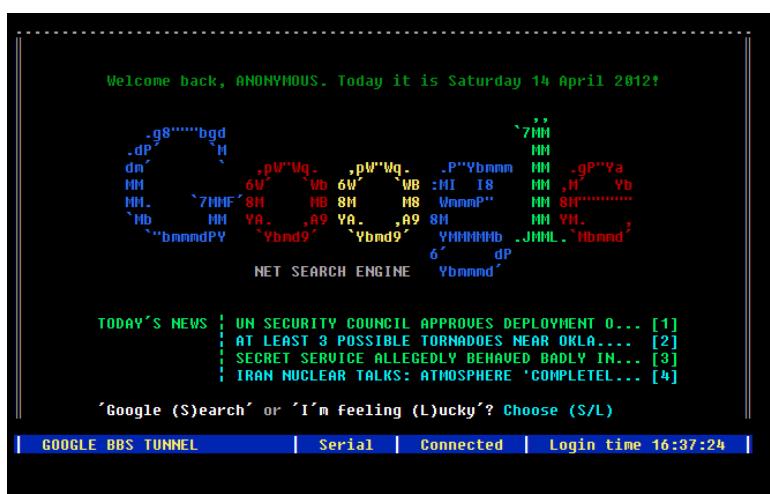


FIGURE 5.5 – minitel - interface google

5.1.3 Internet, un réseau décentralisé

- Les serveurs "routent" les messages
- 1 serveur défaillant = "remplacé" par un autre capable de réaliser les mêmes tâches (services)

Les outils associés à Internet :

- 1971 : Transmission de courriels / emails (Tomlinson)
- 1980 : Télétransmission et stockage d'informations / ftp
- 1989 : Consultation de pages web (www)
- Depuis les années 2000, de nouveaux usages :
 - Consultation de contenus audio / vidéo
 - Téléphonie (IP), vidéo, visioconférence
 - Jeux, réalité virtuelle
 - Forums, chat, réseaux sociaux
 - Partage (p2p, streaming)
 - e-commerce (vente directe, enchères)
 - ...

5.2 Le web

- 1989 : création du WWW : World Wide Web
- Par Tim Berners-Lee et Robert Caillau au CERN



FIGURE 5.6 – Le WWW

- Organisé comme un réseau de documents avec liens hypertextes entre documents (adresse web, URL)
- le Web est une partie d'internet !

5.2.1 Les navigateurs

1993 : **NCSA Mosaic**, 1er navigateur stable, fonctionnel, multiplateformes

1995 : **Netscape Navigator** développé par Marc Andreessen (Mosaic). source : http://www.epocalc.net/pages/mon_internet.htm

1995 : Internet Explorer 1 de Microsoft.

Navigateurs web actuels : Google Chrome, Mozilla Firefox, Internet Explorer, Safari, Opera...

Variantes de navigateurs conçues pour les smartphones/tablettes

- iPhone : Safari Mobile
- Android : Chrome Mobile.
- Windows Phone : Internet Explorer Mobile
- Blackberry : noyau commun à Safari et Chrome

5.2.2 Traffic / Nombre de sites web

source <http://blog.nicolargo.com/>

source <http://www.telegeography.com/telecom-maps/global-traffic-map/>

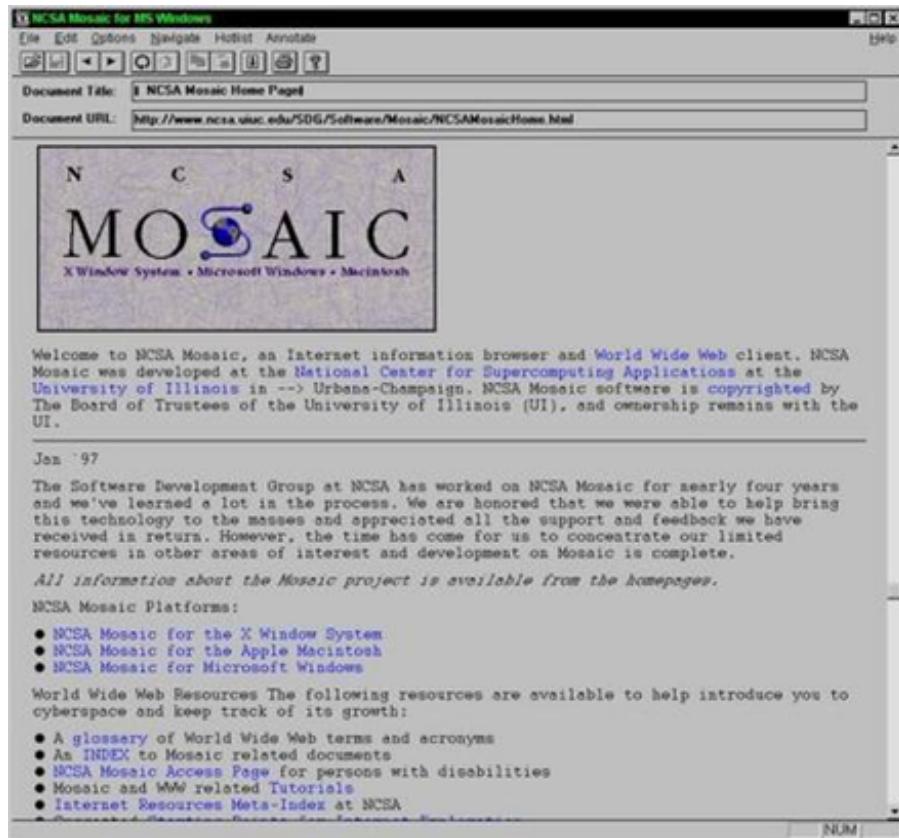


FIGURE 5.7 – NCSA Mosaic

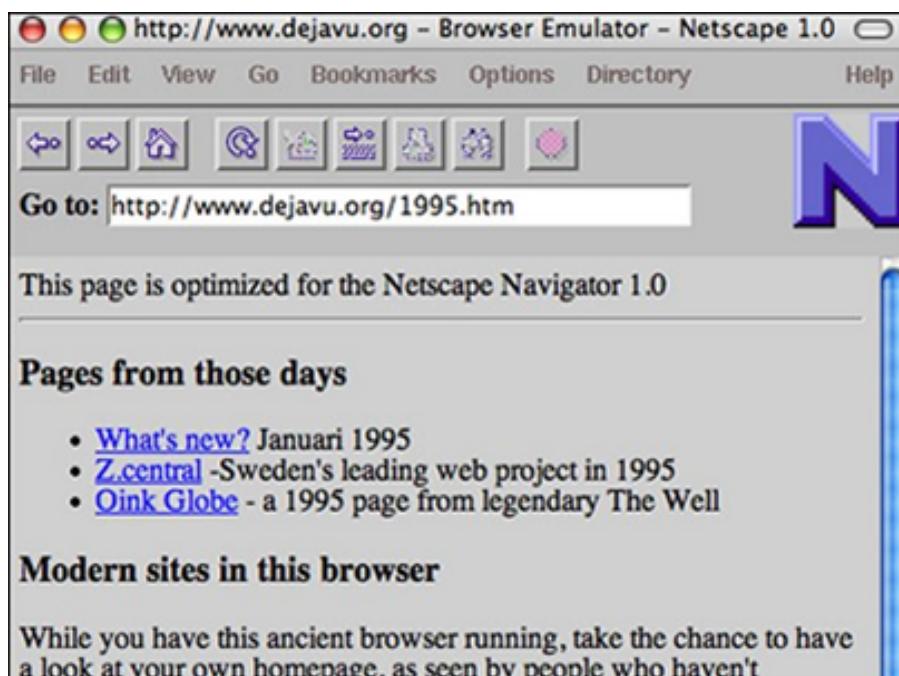


FIGURE 5.8 – netscape



FIGURE 5.9 – navigateur



FIGURE 5.10 – html sans css



FIGURE 5.11 – traffic internet en Europe

5.2.3 Qu'est-ce que le web ?

Web

Service internet permettant de naviguer à travers des pages web

- utilise le protocole **http** (hyperText Transfer Protocol)
- ou sa version sécurisée **https** (pour le commerce électronique)

5.3 Les sites web

5.3.1 Les acteurs du web

- Moteurs de recherche en Europe : 94% Google, 2,3% Bing, 1,4% Yahoo
- Systèmes clients (OS) : 90% Windows, 5% Mac, 2% iOS, 1% Linux
- Clients web (navigateurs) : 57% Internet Explorer, 23% Firefox, 10% Chrome, 6% Safari, 2% Opera
- Serveurs web : 70% Apache, 20% Microsoft IIS, 6% Nginx, 1% Litespeed

Top 500 website (Alsexa) :

- Google, Facebook, Youtube, Yahoo!, Windows Live, Baidu.com, Blogger.com, Wikipedia, QQ.com, Twitter

5.3.2 Qu'est ce qu'un site web ?

Site Web

Ensemble de fichiers HTML (pages web), liés par des liens hypertextes, stockés sur un serveur web

Serveur Web

Ordinateur connecté en permanence à internet, hébergeant les pages web)

Plusieurs catégories de sites web :

- sites vitrine, catalogue, marchands vente et paiement en ligne
- sites d'information, sites institutionnels, sites personnels
- sites communautaires
- sites intranet

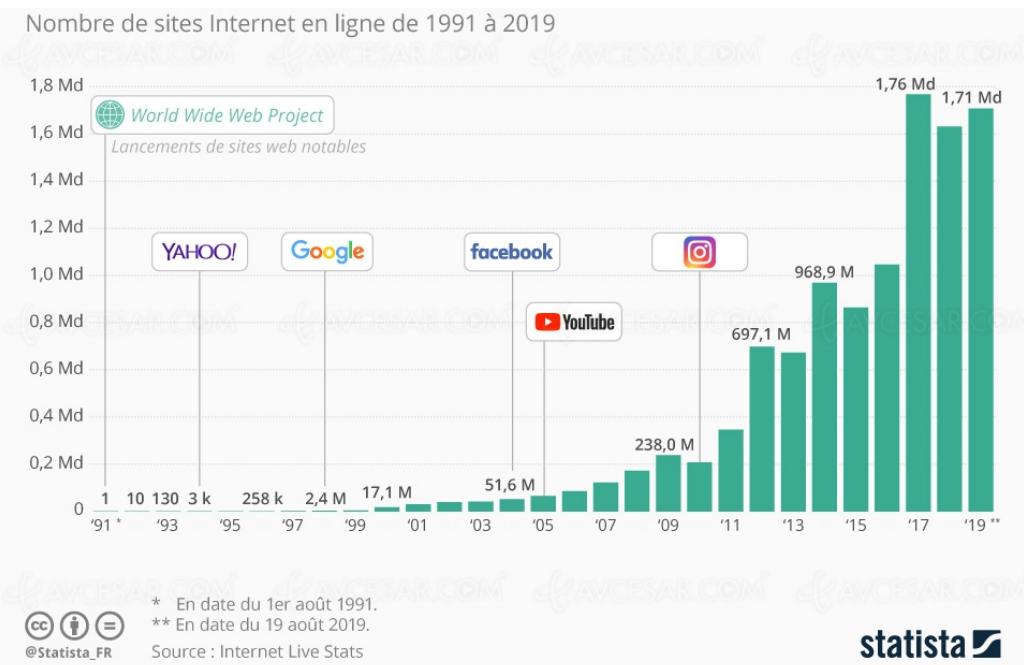
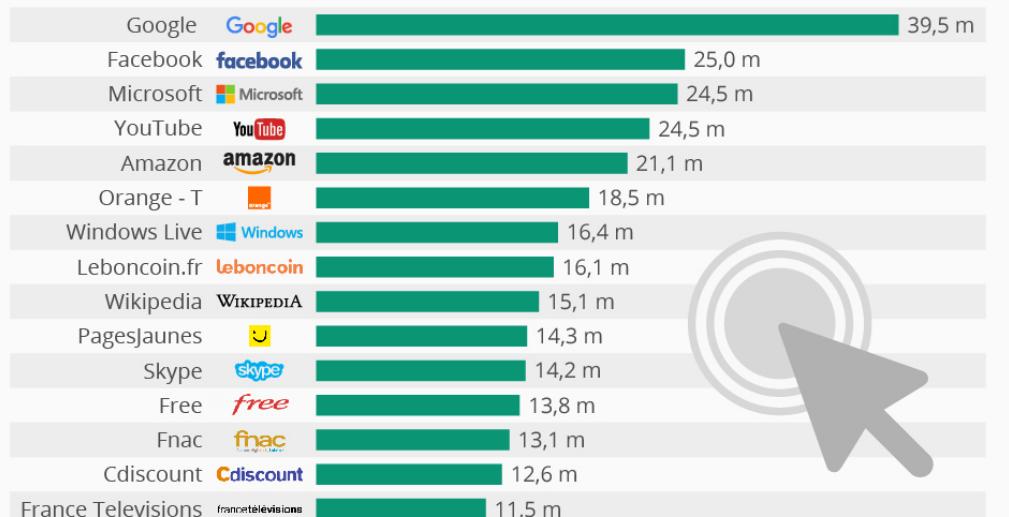


FIGURE 5.12 – Nombres de sites internet

source <https://www.arobasenet.com> 3 000 000 000 000 de pages Web recensées : la partie visible du Web !

Les sites internet les plus visités en France

Nombre de visiteurs uniques par mois en décembre 2015 (en millions)



@StatistaCharts Source : Médiamétrie

statista

FIGURE 5.13 – Sites les plus visités

85% des sites webs sont inactifs
donc environ 255 millions de sites actifs !

source <https://fr.statista.com>
source Wikipedia
source <https://fr.statista.com>
source <https://fr.statista.com>

5.3.3 Nom de domaine

nom de domaine

URL

Un site internet est défini par son URL
Ex : 1 URL fournie par un hébergeur
<http://www.votre-fournisseur.com/votrenom>
<http://www.votrenom.com>

- Outils pour le choix du nom de domaine
 - Whois : outils permettant de vérifier la disponibilité d'un nom de domaine, voire d'en connaître son propriétaire.
 - GANDI ou Network Solutions pour les .com, .net, et .org
 - Afnic pour les .fr
- Déposer un nom de domaine
 - Auprès d'un organisme international (l'Internic)
 - ou de l'un des sous-organismes (pour la France il s'agit de l'Afnic)
 - .com : entreprises commerciales
 - .edu : organismes éducatifs
 - .gov : organismes gouvernementaux
 - .mil : organismes militaires
 - .net : organismes ayant trait aux réseaux
 - .org : entreprises à but non lucratif
 - Sous-domaines : ex : .asso.fr, .gouv.fr, .tm.fr etc.

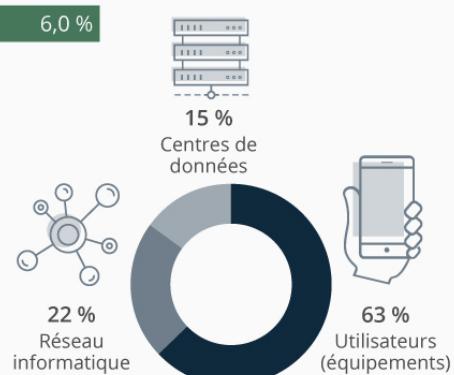
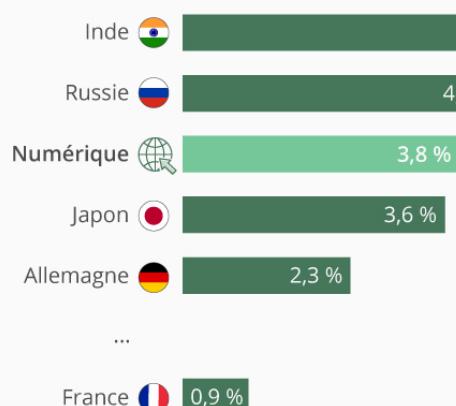


./images/parts_de_marche_des_navigateurs_web.jpeg

FIGURE 5.14 – Parts de marché des navigateurs web, juillet 2021

Le numérique émet 4 fois plus de CO2 que la France

Part des émissions mondiales de CO2 du numérique comparée à celle des pays sélectionnés *



Répartition des émissions du numérique selon la source



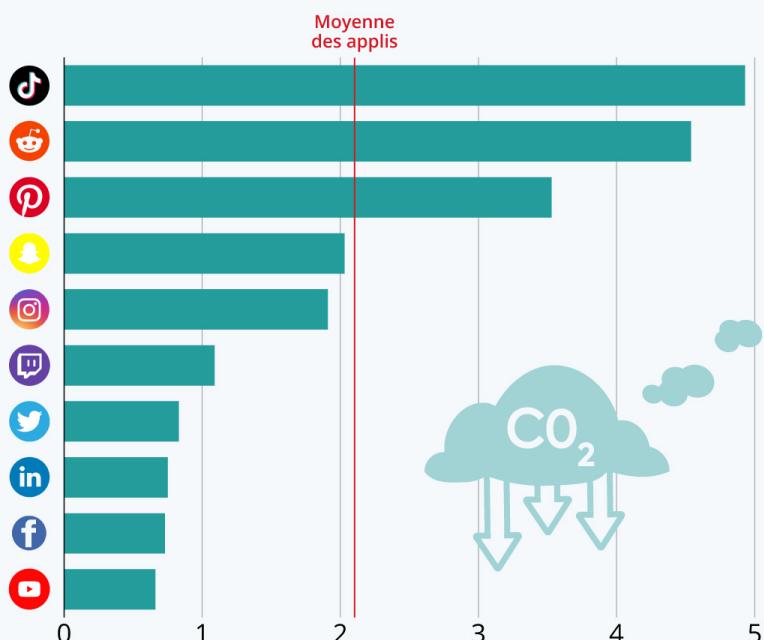
* en 2019 pour le numérique, 2018 pour les pays.
Sources : GreenIT, Agence internationale de l'énergie



FIGURE 5.15 – Emission numérique

Réseaux sociaux : quelle empreinte carbone ?

Estimation du niveau d'émission de CO₂ des applications sélectionnées pour 1 minute d'utilisation, en gEqCO₂ *



* basée sur la consommation d'énergie et le volume de données échangées lors de tests réalisés en France sur smartphone Galaxy S7 (Android 8).

Source : Greenspector



statista

FIGURE 5.16 – Empreinte carbone réseaux sociaux