

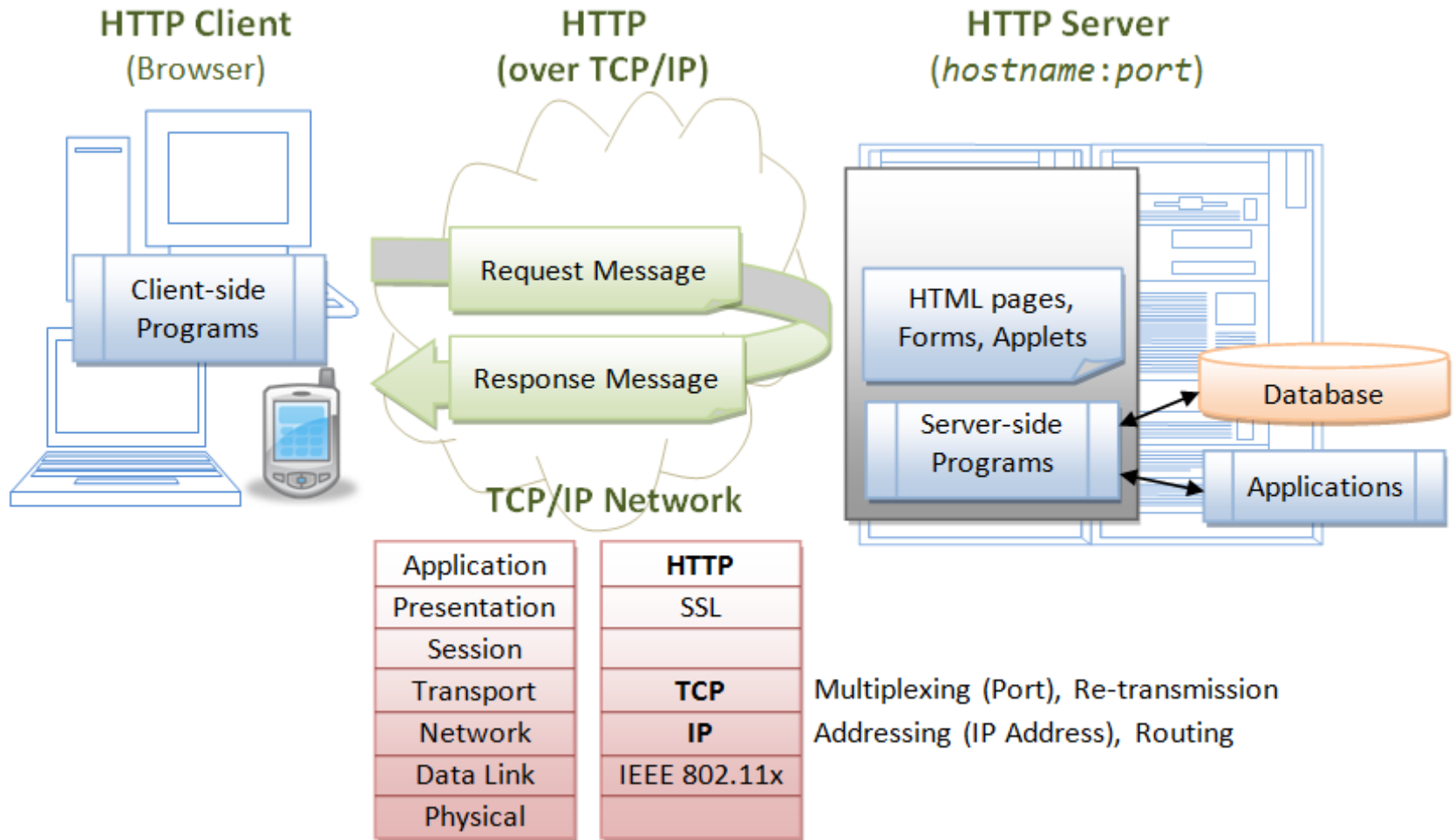
Plan du contenu

HTML5 /CSS3

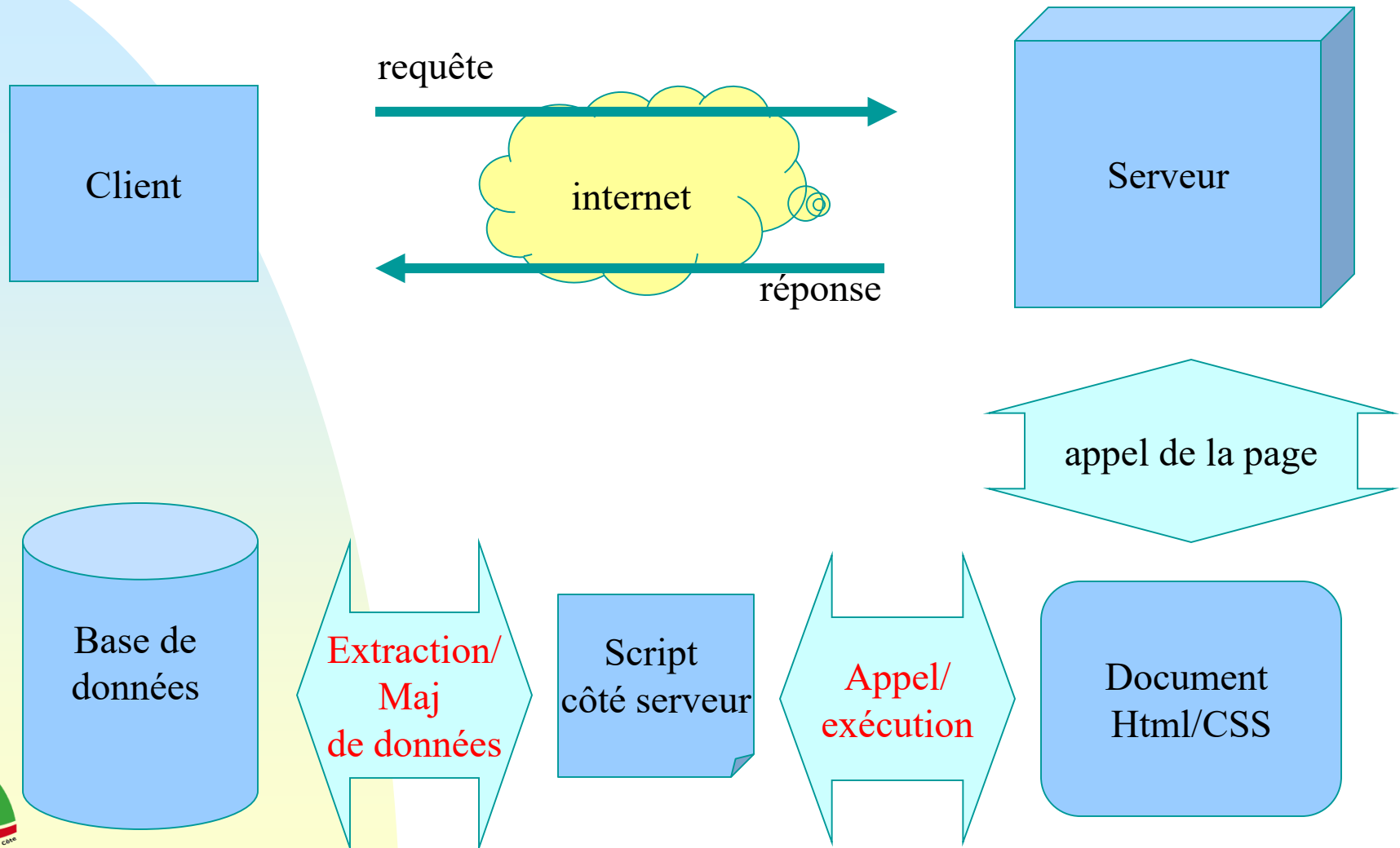
Principe du client/serveur web

1. HTML : présentation
2. HTML5 : plan / tructure d'une page web
3. HTML5 : syntaxe générale
4. HTML5 : les balises et attributs obsolètes
5. HTML5 : les balises et attributs actuels
6. CSS3 : les propriétés
7. CSS3 : les sélecteurs / intégration / cascade
8. CSS3 : background, formatage de texte et positionnement
9. CSS3 : Intégration dans le HTML5 et cascade
10. CSS3 : Intégration du javascript
11. HTML5 : Intégration des Images, de l'Audio et de la Vidéo
12. HTML5 : Les Tableaux
13. HTML5 : Les Formulaire

0. Principe/Architecture du client/serveur web



0. Principe/Architecture du client/serveur web



1. HTML : présentation

HTML (Hypertext Markup Language) est le langage de balisage utilisé pour la **structuration** des documents textuels en pages web. Il permet aux auteurs d'identifier des éléments qui structurent le document, comme les titres, les paragraphes, les listes, etc.

D'autres éléments permettent d'ajouter des liens hypertextes, des formulaires interactifs et des médias, comme du son ou de la vidéo, aux pages web.

Un document HTML balisé représente la couche de structuration d'une page web. Elle sert de base à la couche de présentation (comment les éléments doivent être diffusés ou affichés) et à la couche de comportement (scripts et interactivité).



1. HTML : présentation

1991

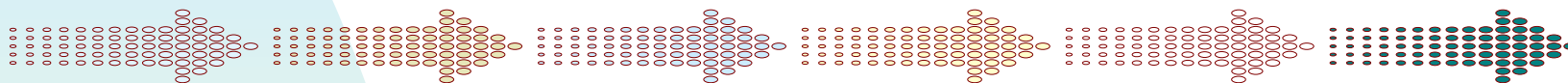
HTML a été créé par Tim Berners-Lee au CERN, le centre européen de recherche nucléaire. La première version de HTML était très simple et contenait seulement une poignée d'éléments tels que `<a>`.

1995

HTML 2.0 est sorti et a introduit de nouveaux éléments tels que `<form>`, `<input>`, `<select>` et `<textarea>` pour les formulaires en ligne.

1997

HTML 3.2 est sorti et a introduit des éléments tels que `<table>` et `<iframe>`, ainsi que l'utilisation de CSS pour la mise en page.



1999

HTML 4.01 est sorti en 1999 et a introduit de nouvelles fonctionnalités telles que les feuilles de style en cascade (CSS) et les éléments `` et `<div>` pour un positionnement plus précis des éléments sur la page.

2000

XHTML (eXtensible HTML) a été publié en 2000 et a remplacé l'utilisation de balises HTML en faveur de balises XML. XHTML a été créé pour rendre le code HTML plus logique, plus propre et plus facilement compréhensible par les navigateurs Web.

2022

HTML5 a été publié en 2014 après des années de développement pour répondre aux besoins croissants de l'Internet, en particulier l'augmentation de la vidéo, de l'audio et des applications interactives en ligne. HTML5 est conçu pour être plus simple, plus sémantique, plus accessible et plus facile à manipuler pour la création de pages et applications Web.

1. HTML : présentation

Amélioration de la sémantique :

HTML5 introduit de nouveaux éléments de balisage pour une meilleure sémantique, tels que <header>, <footer>, <nav>, <article>, <section>, etc. Cela permet aux développeurs de mieux structurer le contenu de leurs pages Web.



Meilleur support du multimédia : HTML5 a introduit des balises vidéos et audios pour permettre la lecture de fichiers multimédias sans l'utilisation de plug-ins tels que Adobe Flash.



Amélioration de la compatibilité mobile

HTML5 a introduit des fonctionnalités pour faciliter le développement de sites Web mobiles, telles que le support de la géolocalisation, l'accès aux contacts, la possibilité de stocker des données hors ligne et l'utilisation de l'accéléromètre

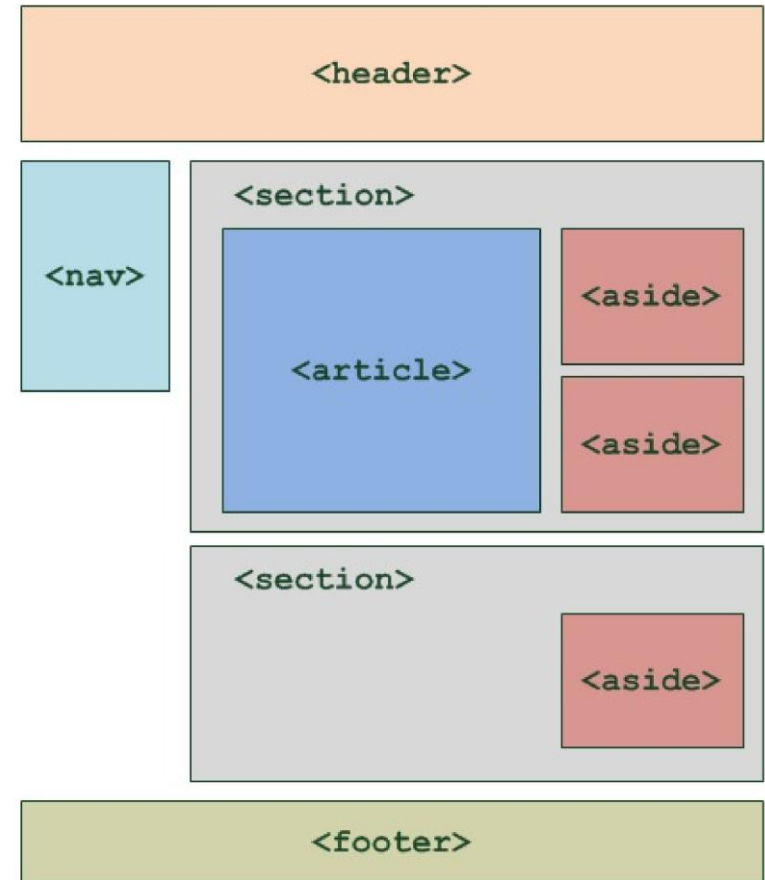
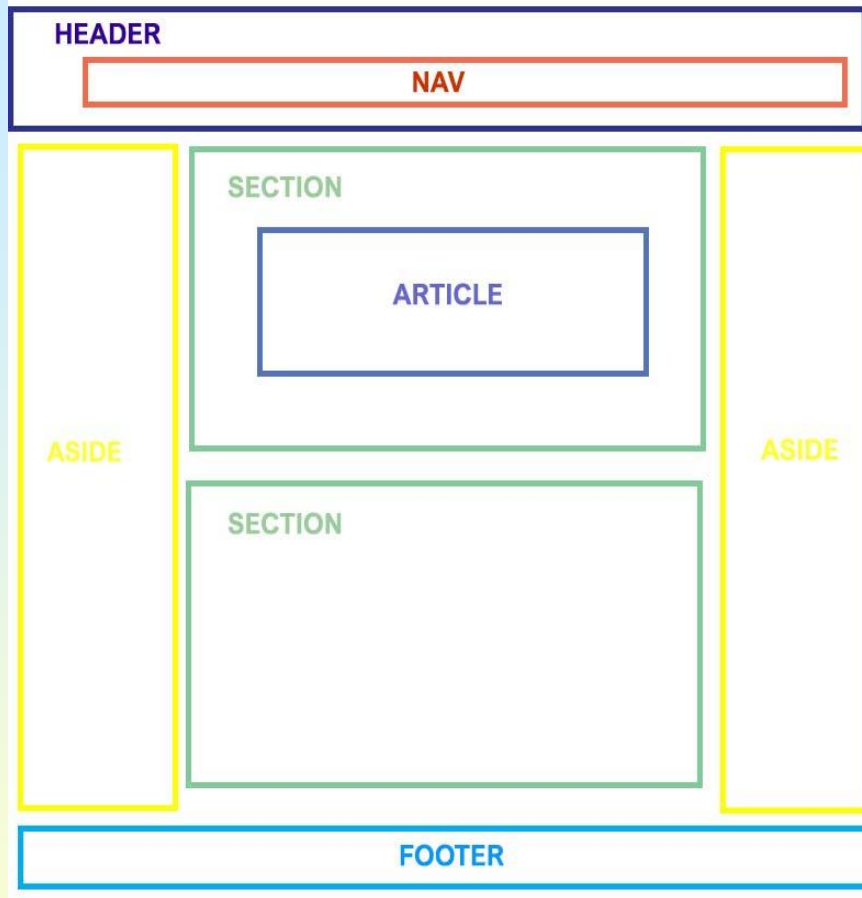


Intégration des nouvelles technologies

HTML5 intègre des technologies de pointe telles que WebSockets pour une communication bidirectionnelle entre le client et le serveur, WebRTC pour la communication en temps réel et les API de stockage pour la gestion des données hors ligne



2. HTML5 : Structure/Plan d'une page web



2. HTML5 : Structure/Plan d'une page web

Légende :

- La Balise `<header>...</header>` représente du contenu introductif, généralement un groupe de contenu introductif ou de contenu aidant à la navigation. Il peut contenir des éléments de titre, mais aussi d'autres éléments tels qu'un logo, un formulaire de recherche, le nom d'auteur, etc.
- La Balise `<nav>...</nav>` représente une section d'une page ayant des liens vers d'autres pages ou des fragments de cette page. Autrement dit, c'est une section destinée à la navigation dans un document (avec des menus, des tables des matières, des index, etc.).
- La balise `<section>...</section>` représente une section générique d'un document, par exemple un groupe de contenu thématique. Une section commence généralement avec un titre.

2. HTML5 : Structure/Plan d'une page web

Légende (suite):

- La balise `<article>...</article>` représente une composition autonome dans un document, une page, une application ou un site, destinée à être distribuée ou réutilisée de manière indépendante (par exemple, dans le cadre d'une syndication). Exemples : un message de forum, un article de magazine ou de journal, ou un article de blog, une fiche produite, un commentaire soumis par un utilisateur, un widget ou gadget interactif, ou tout autre élément de contenu indépendant.
- La balise `<aside>...</aside>` (en français, « aparté ») représente une partie d'un document dont le contenu n'a qu'un rapport indirect avec le contenu principal du document. Les apartés sont fréquemment présents sous la forme d'encadrés ou de boîtes de légende.
- La balise `<footer>...</footer>` représente le pied de page de la section ou de la racine de sectionnement la plus proche. Un élément `<footer>` contient habituellement des informations sur l'autrice ou l'auteur de la section, les données relatives au droit d'auteur (*copyright*) ou les liens vers d'autres documents en relation.

3. HTML5 : Syntaxe générale

- **< !Doctype html>** Permet toujours de rester dans le mode d'interprétation du langage conforme de la famille SGML, d'éviter le mode quirks et de préciser le type de document à suivre.
- **<html lang="fr">** Cette balise permet de spécifier la langue utilisée aux syntaxes.
- **<head>**
 - **<title> Titre de la page </title>**
 - **<meta charset="utf-8">** Permet d'enregistrer le fichier sous le bon encodage.
 - **<link rel="stylesheet" href="style.css">** Permet d'inclure des instructions css directement dans la page html.
 - **<script src="script.js"></script>** permet d'ajouter des scripts (JavaScript) qui vont s'exécuter coté client dans le navigateur dès leur chargement grâce à l'attribut src sachant que le type par défaut de la balise est text/javascript.
- **</head>**
- **<body>**
 - **<!--Commentaire-->**
 - **< Contenu >**
- **</body>**
- **</html>**

4. HTML5 : les balises et attributs obsolètes

VOIR FICHER JOINT

Html5_balises_et_attributs_Obsoletes.pdf

5. HTML5 : les balises

VOIR FICHIER JOINT

Liste complète des balises HTML5 et définitions simples.html

6. CSS3 : Propriétés

Textes : police, style, décoration, couleur,



Blocs : marges, cadre couleur



Format flottant,

Tableaux : cadre, légende, fond



Listes : puce, numérotation



Liens : couleur, décoration

Titre

bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla

| bla | bla | bla | bla | bla | bla |
|-----|-----|-----|-----|-----|-----|
| x | t | a | d | i | l |
| y | u | b | e | j | m |
| z | v | c | f | K | n |

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla :

- bla bla bla
- bla bla
- bla bla bla

[bibliographie](#)

Survol

Fond : couleur, image (position, répétition,

BIBLIOGRAPHIE

6. CSS3 : Propriétés

Le CSS peut être placé dans :

- (1) fichier externe .css
- (2) élément style
- (3) attribut style

règle

sélecteur { [**déclaration** ;]+ }

↓

① p {

propriété:valeur

↓ ↓

font-family:Calibri;

color:purple;

/* sélectionne les éléments p */

}

② ④ *.ex { /* éléments qui ont class=ex */

border: 1px solid silver;

/* équivalent à .ex (* implicite) */

① ⑤ h1: hover { /* survol de h1 */

background-color:#CCCCCC

}

③ #p2 { /* l'élément qui a id=p2 */

text-align:right;

}

style.css

Calibri ▾

A ▾

Sélecteur

- ① élément
- ② *
- ③ #id
- ④ .classe
- ⑤ :pseudo-classe

6. CSS3 : Propriétés

Les propriétés vont nous permettre de choisir quel(s) aspect(s) (ou “styles”) d’un élément HTML on souhaite modifier.

background-color: red;
background: url(images/bg.gif) no-repeat top right;

```
p{  
  color: blue;  
  border: 2px solid orange;  
  padding: 5px;  
}
```

- ❑ **border-width: 1em 2em 3em 4em** — Les quatre valeurs représentent respectivement le côté haut, le côté droit, le côté bas et le côté haut, toujours dans cet ordre (le sens horaire).
- ❑ **border-radius: 1em 2em 3em 4em** — Les quatre valeurs s'appliquent respectivement au coin en haut à gauche, en haut à droite, en bas à droite et en bas à gauche, toujours dans cet ordre (le sens horaire).

6. CSS3 : Propriétés et valeurs

font-style: italic;
font-weight: bold;
font-size: .8em;
line-height: 1.2;
font-family: Arial, sans-serif;

margin-top: 10px;
margin-right: 5px;
margin-bottom: 10px;
margin-left: 5px;

background-color: #000;
background-image:
url(images/bg.gif);
background-repeat: no-repeat;
background-position: top right;

7. CSS3 : Sélecteurs / intégration / cascade

Sélecteur de balise

Le sélecteur de balise permet tout simplement d'appliquer du style à toutes les balises concernées de la page

```
baliseHTMLconcernée {propriété: valeur; }
```

Exemple :

```
img { width:120px; height:100px; }
```

Sélecteur de classe

Certains éléments de document HTML possèdent un attribut **class** ou **id**. nous voulions changer la taille de plusieurs éléments dans la page, mais sans les changer toutes. Nous utiliserons alors un sélecteur de **type class** qui sera attribué uniquement aux éléments concernés. Le sélecteur class est représenté par un **.** suivi de son nom.

En CSS

```
.voiture { width:120px;
           height:80px; }
.moto { width:120px;
        height:120px; }
```

Intégration dans HTML

```

... l'image aura pour dimensions 120px x 80px

... l'image aura pour dimensions 120px x 120px
<div class="auto">
... la balise aura pour dimensions 120px x 100px
```

7. CSS3 : Sélecteurs / intégration / cascade

Sélecteur d'identifiant

Pour sélectionner un unique élément de la page, il faut donner à la balise HTML un identifiant.
#identifiant { ... }

En CSS

```
#texterouge { color: red ; }
```

Intégration dans HTML

```
<h2>titre affiché normalement </h2>
```

```
<h2 id="texterouge">titre affiché en rouge </h2>
```

Hierarchie des sélecteurs

Pour appliquer un style à un élément à condition qu'il se trouve lui même dans un autre éléments. Pour cela, il suffit de séparer les noms de balises par un espace

pour modifier uniquement la taille de police de caractère des balises
<h1> contenu dans un bloc <div> :

```
div h1 { font-size: 22px; }
```

Pour modifier les éléments <a> (lien) contenu dans une ligne de liste , ces derniers sont contenus dans une <div> dont l'id est "menu" :

```
div #menu ul li a {  
font-size: 22px; text-decoration:  
underline; color: green; }
```

7. CSS3 : Sélecteurs / intégration / cascade

Multi-sélections

Pour appliquer une règle de style à plusieurs balises différents sans avoir à réécrire la dite règle. Pour cela il vous suffit de déclarer plusieurs éléments en les séparant par une virgule

En CSS

```
#article, #titre, #modele {  
font: 14px #000000 arial; }
```

Sélecteurs d'attributs

Un moyen pour sélectionner une balise ayant un attribut donné. Plutôt que de différencier chaque balise par un id spécifique, vous pouvez les sélectionner grâce à leur attribut

exemple, le style des balises
type texte et de type bouton très
utilisées dans les formulaires

```
input [type="text"] {  
border: solid 2px #919191; }
```

```
input [type="submit"] {  
border: solid 1px #919191;  
font: black 10px arial; }
```

7. CSS3 : Sélecteurs / intégration / cascade

Pseudo-classes

vous pouvez apporter des précisions aux sélecteurs en utilisant des pseudo-classes accolées à une balise. On distingue notamment :

```
:first-child, :last-child, :link, :hover, ... .
```

exemple la pseudo-classes :hover afin de modifier la couleur à rose d'un lien au passage ou survol de la souris :

```
a:hover {  
  color: pink; }
```

Sélecteur universel

Ce sélecteur est représenté par une étoile et permet d'appliquer une règle de style à l'ensemble des balises du document.

Si, par exemple, pour ôter toutes les marges par défaut de toutes les balises, voici la règle ci-contre :

```
* { margin: 0; padding:0; }
```

Les commentaires

/* voici un commentaire */

7. CSS3 : quelques sélecteurs

Pour pouvoir appliquer un style à un contenu, il va déjà falloir le cibler, c'est-à-dire trouver un moyen d'indiquer qu'on souhaite appliquer tel style à un contenu en particulier. D'où l'importance des sélecteurs. Il existe différents types de sélecteurs dont quelques uns :

| Sélecteurs | Utilité |
|-----------------------------------|---|
| H1{ } | <u>Permet de cibler les elements inclus dans la balise h1</u> |
| .box{ } | <u>Permet de cibler les elements d'une classe</u> |
| #unique { } | <u>Permet de cibler un ID</u> |
| a[<u>title</u>] { } | Permet de cibler des éléments en fonction de la présence d'un attribut donné |
| a[href="https://example.com"] { } | Permet de baser la sélection sur la présence d'un attribut avec une valeur bien précise |

7. CSS3 : quelques sélecteurs

Ce groupe de sélecteurs inclut les pseudo-classes, qui ciblent des éléments dans un état donné

| <u>Sélecteurs</u> | <u>Utilité</u> |
|---------------------------------|---|
| <code>a:hover { }</code> | la pseudo-classe <u>:hover</u> sélectionne un élément seulement s'il est survolé par le pointeur de la souris |
| <code>p::first-line { }</code> | <code>::first-line</code> sélectionne la première ligne d'un texte contenu dans un élément (la balise p). |
| <code>article > p { }</code> | Permet de sélectionner les paragraphes enfants directs de <article> grâce au combineur enfant (>) |
| * | sélectionne tout dans le document (ou à l'intérieur de l'élément parent s'il est enchaîné avec un autre élément et un combineur descendant, par exemple). |
| <code>li[class]</code> | On cible ainsi tous les éléments de la liste |

8. CSS3 : background, formatage de texte et positionnement

| Propriétés | Type de valeurs | Rôle |
|--------------------|---|---|
| Font-size | <ul style="list-style-type: none"> ❖ valeurs de type absolu (en pixel ou en point) ❖ Valeurs de type relatif (en em, ex ou en pourcentage) | Permet de modifier la taille d'un texte |
| Font-style | <ul style="list-style-type: none"> ❖ Normal (valeur par défaut) ❖ Italic (change le texte en italique) ❖ Oblique (penche le texte) ❖ Inherit (hérite des propriétés de l'élément parent). | permet de fixer l'inclinaison d'un texte |
| Font-weight | <ul style="list-style-type: none"> ❖ Normal (la valeur par défaut) ; Lighter (version allégée de la police) ❖ Bold (la police est en gras) ; ❖ Bolder (la police est encore plus en gras) ❖ Inherit (hérite des styles de ses parents). | permet de fixer le poids d'un texte |
| Font-family | | permet de choisir la police de notre texte. |

8. CSS3 : background, formatage de texte et positionnement

| Propriétés | Valeurs | Description |
|------------------------------|---|---|
| color | rgb(rouge, vert, bleu), rgba(rouge,vert,bleu,transparence) | Couleur du texte |
| Background-color | rgb(rouge, vert, bleu), rgba(rouge,vert,bleu,transparence) | Couleur de fond |
| Background-image | url('image.png') | Image de fond |
| Background-attachment | fixed, scroll | Répétition du fond |
| Background-repeat | repeat-x, repeat-y, no-repeat, repeat | Position du fond |
| Background-position | top, center, bottom, left, right | Super propriété du fond. |
| background | | Combine : background-image, background-repeat, background-attachment, background-position |
| opacity | | Transparence |

9. CSS3 : intégration dans le HTML / Cascade

1ère méthode : code CSS dans une balise HTML

intégrer le code CSS directement dans votre balise HTML en utilisant l'attribut style. Cette technique est à éviter puisqu'elle ne permet pas de dissocier le contenu de la mise en forme. La mise à jour des pages devient alors difficile et cela supprime la clarté du code dans votre document.

Cette solution existe mais réservée la à une utilisation ponctuelle, pour par exemple définir un style sur un seul élément n'apparaissant que sur une seule page de votre site

```
<body>
...
  <div style="background-image:url(images/fond.jpg); border-color: #849460;
    border-style:solid; border-width: 2px; border-radius: 10px; width:300px;
    margin:auto; text-align:left;    padding-left:80px;">
    Contenu de votre élément
  </div>
...
</body>
```

9. CSS3 : intégration dans le HTML / Cascade

2ème méthode : Feuille de style interne

Ecrire du code CSS dans l'entête d'un document HTML, dans la balise <head>, en encadrant avec une balise <style type="text/css">

Cette méthode permet de retirer la mise en forme du corps du document, mais elle ne s'appliquera qu'à la page concernée et non à l'ensemble du site.

```
<head>
  <style type="text/css">
    #header{position: relative; text-align: center; width: 900px; margin: auto;
    height: 200px; background-image: url(image.png); }
    body { font: 16px "Carnivalee Freakshow"; margin: auto; position: relative;
    width: 900px; height: auto; background: url(images/fondpage.png) repeat-x center; }
    #boite1 { position: relative; margin: auto; margin-left: 20px; height: auto; width:
    643px; float:left; background-image: url(images/catalog.png); }
    #footer { font-size: 11px Arial, sans-serif; line-height: 14px; position: relative;
    width: 900px; height: 80px; background-image: url(images/piedpage.png) no-
    repeat center; }
  </style>
  ...
</head>
```

9. CSS3 : intégration dans le HTML / Cascade

3^{ème} méthode : feuille de style externe

Elle consiste à rédiger le code CSS dans un document externe afin qu'il puisse être appelé par les différentes pages de notre site. La séparation du fond et de la forme permet de retrouver et de corriger plus rapidement le texte des pages, de facilement mettre à jour le design et rendre vos document HTML plus léger.

Pour déclarer une feuille de styles externe, il suffit de l'appeler dans le <head> de votre document HTML.

Soit le document de style nommé : monStyle.css

```
@charset "utf-8";
/* CSS Document */
#header{
position: relative; text-align: center; width:
900px; margin: auto; height: 200px;
background-image: url(image.png); }
body {
font: 16px "Carnivalee Freakshow"; margin:
auto; position: relative; width: 900px; height:
auto;
background: url(images/fondpage.png) repeat-x
center; }
```

Intégration de la feuille dans le document HTML

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

9. CSS3 : intégration dans le HTML / Cascade

```

<!DOCTYPE html PUBLIC
  "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">

```

```

<html>

```

```

<head>

```

```

  <title>test CSS</title>

```

```

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

```

```

  <style type="text/css">

```

```

    <!--

```

```

      p { color:blue }

```

```

      em { color:red }

```

```

    -->

```

```

  </style>

```

```

</head>

```

```

<body>

```

```

  <h1 style="text-align:center; color:navy">titre</h1>

```

```

  <p>C'est un <em>test</em> de style.</p>

```

```

  <p id="p2">Un 2eme <em>paragraphe</em></p>

```

```

  <p class="ex" style="color:black">test de classe</p>

```

```

</body>

```

```

</html>

```



**préférences
utilisateur**

1 externe

2 global

3 local

Priorités



**style du
navigateur**



titre

C'est un *test* de style.

Un 2eme *paragraphe*

test de classe

9. CSS3 : intégration dans le HTML / Cascade

Hierarchie des feuilles CSS

Il y a une hiérarchie dans l'interprétation des feuilles de style par votre navigateur. Une règle de style intégrée au sein du code HTML, par l'attribut "style" prédomine sur une feuille de style interne, intégrée dans le <head> de votre page, qui elle-même prédomine sur une règle de style établie dans une feuille de styles externe, appelée dans le <head> de votre document par une balise <link>.

Ainsi :

Feuille de style externe (appelée dans le >head> par
`<link rel="stylesheet" media="screen" type="text/css" href="monstyle.css"/>`)
`img { width:120px; height:80px; }`

peut être modifiée par une feuille de style interne :

```
<style type="text/css">
img { width:120px; height:100px; }
</style>
```

Elle-même est modifiable au sein du corps de page par

```
<img style="width:120px; height:120px;"... >
```

9. CSS3 : intégration dans le HTML / Cascade

la **cascade** des styles signifie que l'ordre d'apparition des règles dans le CSS a une importance ; quand deux règles applicables ont la même spécificité, c'est la dernière déclarée qui sera utilisée pour la mise en forme.

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

Au final h1 est coloré en bleu - ces règles ont les mêmes sélecteurs, elles ont donc la même spécificité ; dans ce cas, c'est la règle écrite en dernier dans le CSS qui l'emporte.

9. CSS3 : intégration dans le HTML / Cascade

La spécificité mesure essentiellement combien la sélection est précise :

Un sélecteur d'élément est peu spécifique — il cible tous les éléments d'un type donné dans la page — son score est donc faible ;

Un sélecteur de classe est plus spécifique — dans la page, il ne cible que les éléments dont l'attribut class a la valeur choisie — son score est plus important.

```
.main-heading {
```

```
  color: red;
```

```
}
```

```
h1 {
```

```
  color: blue;
```

```
}
```

```
<h1 class="main-heading">This is my heading.</h1>
```

Au final h1 est coloré en rouge — le sélecteur de classe donne une plus grande spécificité à sa règle, et du coup c'est cette règle qui est choisie même si elle apparaît plus tôt dans le CSS.



9. CSS3 : intégration dans le HTML / Cascade

Certaines valeurs pour une propriété CSS se transmettent des éléments parents vers leurs enfants, d'autres non.

Par exemple, si l'on fixe une **color** et une **font-family** pour un élément, tout élément contenu dans le premier sera mis en forme avec la même couleur et la même police, à moins qu'on lui ait appliqué directement des règles.

```
.main {
    color: rebeccapurple;
    border: 2px solid #ccc;
    padding: 1em;}
.special {
    color: black;
    font-weight: bold;
}
<ul class="main">
    <li>Item One</li>
    <li>Item Two
</ul>
```


9. CSS3 : intégration dans le HTML / Cascade

CSS propose quatre valeurs spéciales universelles pour contrôler l'héritage. Ces valeurs sont applicables à toute propriété CSS.

INHERIT : La propriété correspondante prend la valeur définie dans l'élément parent. Dans les faits, cela "active l'héritage" .

INITIAL : La propriété correspondante prend la valeur par défaut définie dans la feuille de style du navigateur. Si aucune valeur n'est définie par défaut dans le navigateur et que la propriété est transmise par héritage la propriété est redéfinie à inherit.

UNSET : Redéfinit la propriété à sa valeur naturelle : si la propriété est transmise par héritage, le comportement est le même que inherit, sinon il est identique à initial.

9. CSS3 : intégration dans le HTML / Cascade

```
body {  
    color: green;  
}
```

```
.my-class-1 a {  
    color: inherit;  
}
```

```
.my-class-2 a {  
    color: initial;  
}
```

```
.my-class-3 a {  
    color: unset;  
}
```

10. HTML5 : intégration de JavaScript

Javascript s'intègre aussi dans le code HTML (tout comme CSS). Donc, notre page Web peut contenir trois syntaxes différentes: Html, **CSS** et JavaScript.

Mais pour que le navigateur sache où commence notre script et où finit il, il faut le mettre à l'intérieur des délimiteurs du script,

1. la balise <script>.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
  </head>
  <body>
    <script>
      alert("Bonjour");
    </script>
  </body>
</html>
```

10. HTML5 : intégration de JavaScript

Nous avons deux autres possibilités d'intégration du code javascript à savoir :

1ère

```
<script language="javascript">  
    // code JS  
</script>
```

2ème

```
<script type="text/javascript">  
    // code JS  
</script>
```

```
<script type="text/javascript" src="monFichierExterne.js">  
    // autre code JS  
</script>
```

- La deuxième déclaration est par contre plus appréciée vu qu'elle fournit deux informations, à savoir le type et le MIME. Le type est **text** car Javascript c'est du texte, et le MIME précise qu'il ne s'agit pas de n'importe quel texte mais du **JavaScript**.

11. HTML5 : intégration de l'audio et de la vidéo

Intégration de l'image

`background-image: url(borderh.gif);` *image de fond dans une page html.*

`list-style-image: url(dot-rouge.png);` *image utilisée comme puce devant les éléments de listes.*

``

Image spécifiant la taille et le texte si l'image ne peut s'afficher.

Intégration d'audio

`<audio>`

`<!-- Des informations-->`

`<!-- sur l'audio-->`

`</audio>`

`<audio>`

`<source src="chemin/vers/masource.mp3" type="audio/mp3">`

`</audio>`

Intégration vidéo

`<video>`

`<!-- Des informations-->`

`<!-- sur la vidéo-->`

`</video>`

`<video>`

`src="http://masource.com/lavideo.avi">`

`</video>`

`<video>`

`<source src="chemin/vers/masource.mp4" type="video/mp4">`

`<source src="chemin/vers/masource.ogg" type="video/ogg">`

`<source src="chemin/vers/masource.webm" type="video/webm">`

`</video>`

12. HTML5 : les tableaux

La balise `<table>` indique au navigateur qu'il l'affichage d'un tableau. la balise `<caption>` définit le titre du tableau. La balise `<th>` définit la ligne d'en-tête du tableau HTML. La balise `<tr>` définit une ligne dans le tableau de et la balise `<td>` définit une cellule standard dans une table de HTML..

Exemple

```
<table>
  <caption>Comparatif des formules d'abonnement</caption>
  <tr> <th>Abonnement</th> <th>Durée</th> <th>Prix (fcfa)</th> </tr>
  <tr> <td>mensuel</td> <td>1 mois</td> <td>5.000 </td> </tr>
  <tr> <td>semestriel</td> <td>6 mois</td> <td>25.000 €</td> </tr>
  <tr> <td>annuel</td> <td>12 mois</td> <td>40.000</td> </tr>
</table>
```

Rendu

Comparatif des formules d'abonnement

| Abonnement | Durée | Prix (fcfa) |
|------------|---------|-------------|
| mensuel | 1 mois | 5.000 |
| semestriel | 6 mois | 25.000 |
| annuel | 12 mois | 40.000 |

Dépendamment
du style des
éléments

13. HTML5 : les formulaires

Un formulaire permet à un internaute de saisir des informations à envoyer sur le web

```
<form action="cible.php" method="get">  
...  
...  
</form>
```

- L'attribut action définit la page cible du formulaire. Si laissé vide, c'est la même page.
- L'attribut method définit la forme d'envoi du formulaire,
 - get envoie les informations via l'URL (HTTP Get).
 - post envoie les informations via HTTP Post. L'attribut enctype définit l'encodage. Pour l'envoi de fichiers, utiliser enctype="multipart/form-data".

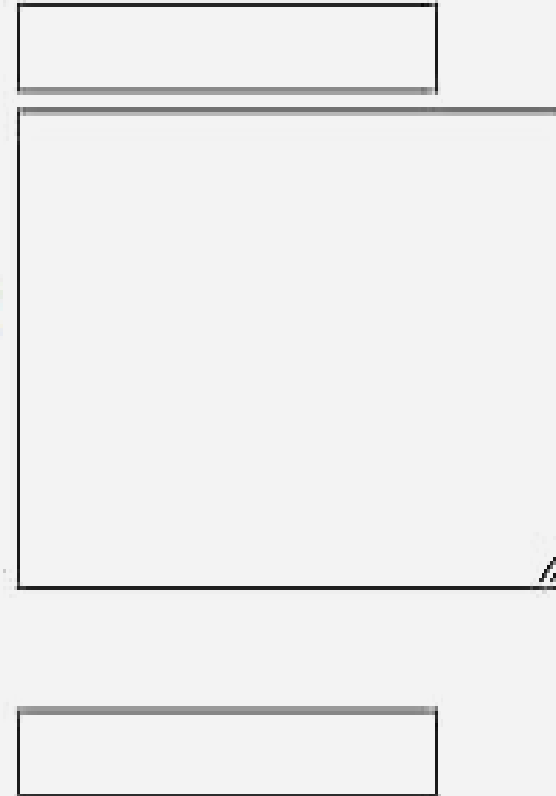
13. HTML5 : les formulaires

Formulaire classique

```
<input type="text" name="prenom" />
```

```
<textarea name="commentaire" rows="7" cols="22"></textarea>
```

```
<input type="password" name="pass" />
```

The image shows a visual representation of the HTML form elements. It consists of three rectangular boxes stacked vertically. The top box is a single-line text input field. The middle box is a multi-line text area, significantly larger than the text input field. The bottom box is a single-line password input field, which typically displays characters as dots or asterisks. These boxes are arranged to correspond with the HTML code snippets provided on the left.

13. HTML5 : les formulaires

Formulaire classique

```
<input type="password" name="pass" />
```

```
<input type="checkbox" name="voiture" />
```

☐

```
<input type="radio" name="sexe" value="masculin" />
```

```
<input type="radio" name="sexe" value="feminin" />
```

☐ ☐

```
<select name="chats">
```

```
  <option value="1">Un chat</option>
```

```
  <option value="2">Deux chats</option>
```

```
  <option value="plus">Plus que deux chats</option>
```

```
</select>
```

```
<input type="submit" value="Soumettre" />
```

13. HTML5 : les formulaires

Regroupement et étiquetage des champs (<fieldset>, <legend>, <label>)

Exemple

```
<form action="cible.php" method="post">
  <fieldset>
    <legend>Identité</legend>
    <label for="nom">Nom:</label>
    <input type="text" name="nom" />
    <label for="prenom">Prénom:</label>
    <input type="text" name="prenom">
  </fieldset>
</form>
```

Rendu

Dépendamment
du style des
éléments

Identité

Nom:

Prénom:

13. HTML5 : les formulaires

Nouveaux types de champs < input /> HTML5

| | |
|--|--|
| <code><input type="color" /></code> | <code><input type="search" /></code> |
| <code><input type="date" /></code> | <code><input type="tel" /></code> |
| <code><input type="datetime" /></code> | <code><input type="time" /></code> |
| <code><input type="datetime-local" /></code> | <code><input type="url" /></code> |
| <code><input type="email" /></code> | <code><input type="week" /></code> |
| <code><input type="month" /></code> | |
| <code><input type="number" min="0" max="10" value="4" step="2" /></code> | |
| <code><input type="range" min="0" max="10" value="4" step="2" /></code> | |

Exemple pour une feuille de style

```
input
{
    border:3px solid #cccccc;
    font-size:20px;
    border-radius:5px;
}
input:invalid
{
    background-color: #f0dddd;
}
```

/ autres pseudo-classes : valid, invalid, required, optional, in-range, out-of-range, ... */* 43

[première page](#)

13. HTML5 : les formulaires

Nouveaux attributs de champs HTML5

| | |
|----------------|--|
| Required | <code>required="required"</code> , le champ doit être non-vide pour que le formulaire soit validé |
| Placeholder | p.ex. <code>placeholder="email ici"</code> . Affiche une valeur indicative dans le champ. |
| Autocomplete | <code>autocomplete="on"/autocomplete="off"</code> . Prise en charge de l'historique du navigateur |
| Autofocus | <code>autofocus="autofocus"</code> , à l'ouverture de la page, cela donne le focus au champ |
| Form | permet de placer un champ en dehors du formulaire, en associant au formulaire par ID |
| Pattern | p.ex. <code>pattern="[A-Za-z0-9]"</code> . Expression régulière du format du champs. |
| min, max, step | concerne <code>number</code> , <code>range</code> , <code>date</code> , <code>datetime</code> , <code>datetime-local</code> , <code>month</code> , <code>week</code> , <code>time</code> |
| pattern | p.ex. <code>pattern="[A-Za-z0-9]"</code> . Expression régulière du format du champs. |
| ... | ... |