



# HTML / CSS

Alexis Ducerf – [alexis.ducerf@DeerCoders.com](mailto:alexis.ducerf@DeerCoders.com)



Web

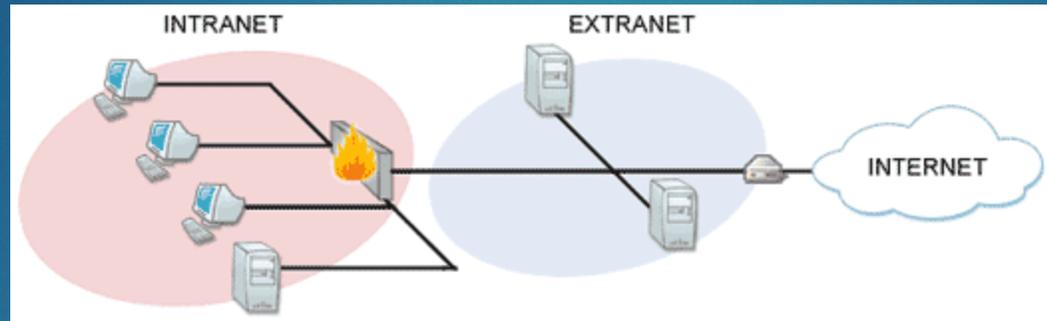
# Les protocoles du web

- ▶ **IP** : Un numéro d'identification qui est attribué de façon permanente ou provisoire à chaque appareil connecté au réseau
- ▶ **TCP** : il permet la connexion entre deux machines
- ▶ **DNS** : Le système de nom de domaine (Domain Name System) a été créé pour établir une correspondance entre un nom de domaine et une adresse IP
- ▶ **FTP** : Protocole de transfert de fichier (File Transfer Protocol), il crée un flux de données entre le serveur et le client
- ▶ **HTTP** : Le protocole de transfert hypertexte (HyperText Transfer Protocol) est le principal canal de diffusion de données sur Internet, principalement des fichiers HTML
- ▶ **SMTP – POP/IMAP** : protocoles de transports de courriers électroniques

# Quelques chiffres

- ▶ 1973 : Création du protocole TCP/IP
- ▶ 1983 : Adoption du protocole TCP/IP et du mot « Internet »
- ▶ 1983 : Premier serveur de noms de sites (serveur DNS)
- ▶ 1990 : Annonce publique du World Wide Web
- ▶ 2000 : Explosion de la bulle internet (368 540 000 ordinateurs connectés)
- ▶ 2014 : La barre du milliard de sites web est franchie

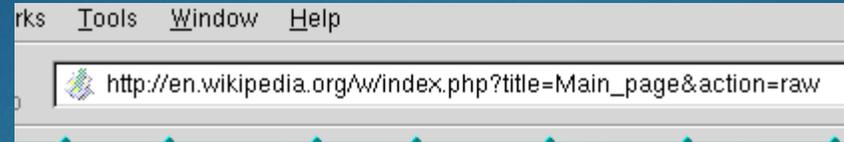
# Intranet / extranet



On fera toujours la différence entre :

- ▶ Intranet : un réseau informatique utilisé à l'intérieur d'une entreprise ou de toute autre entité organisationnelle qui utilise les mêmes protocoles qu'Internet
- ▶ Extranet : Un extranet (ou réseau interne étendu) est un réseau de type internet conçu pour faciliter les échanges entre une organisation sociale et ses correspondants extérieurs.

# URL



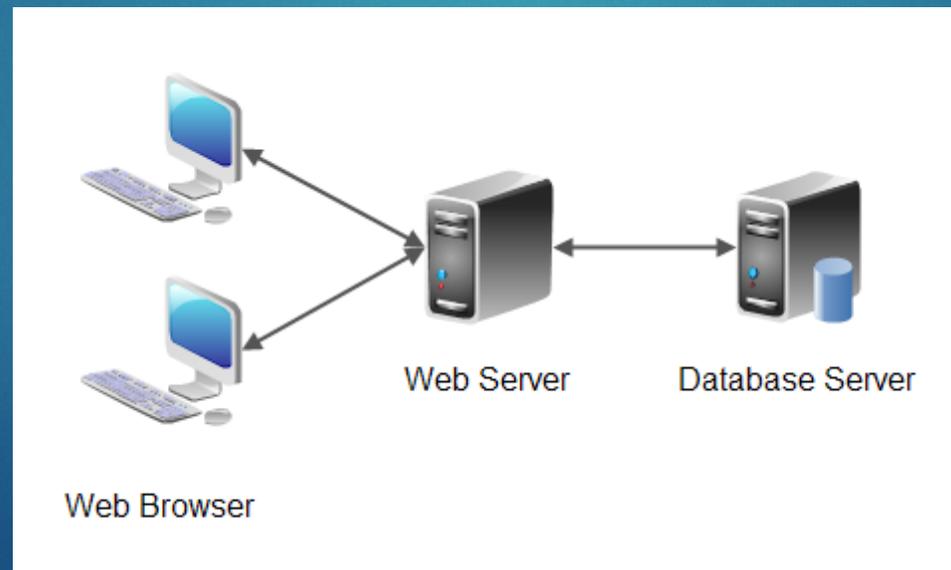
- ▶ Uniform Resource Locator
- ▶ En France, d'après le Journal officiel du 16 mars 1999, « URL » peut être traduit par adresse réticulaire ou adresse universelle.
- ▶ Les liens sont le sang du web

# L'architecture client serveur

On parlera d'architecture 3 tiers. L'architecture permet de faire la communication entre le client et la base de données en passant par le serveur web.

## Exemple :

Le navigateur Chrome appelle le serveur Apache/PHP et PHP fera une requête sur la base de données MySQL.



# La structure d'un message HTTP

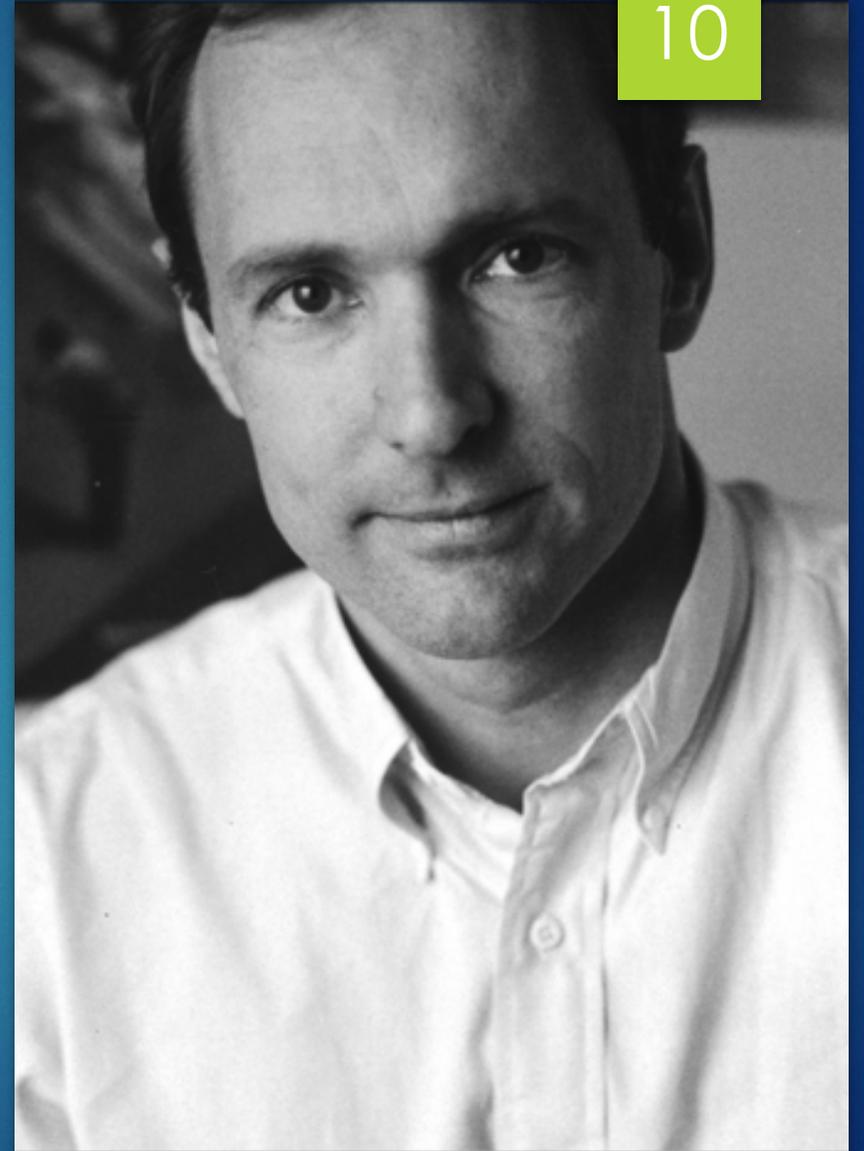
```
User-Agent: curl/7.16.3 libcurl/7.16.3 OpenSSL/0.9.7l  
zlib/1.2.3  
Host: www.example.com  
Accept-Language: en, mi Date: Mon, 27 Jul 2009 12:28:53 GMT  
Server: Apache  
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT  
ETag: "34aa387-d-1568eb00"  
Accept-Ranges: bytes  
Content-Length: 51  
Vary: Accept-Encoding  
Content-Type: text/plain
```

# HTML?

```
TYPE HTML PUBLIC "-//IETF//  
</>  
<head>  
<meta name="TITLE"  
<meta name="KEYWOF  
<meta name="DESC  
<link rel="sto  
<script lang  
</head>  
<body bgcolor
```

# HTML

- ▶ Idée de Tim Berners-Lee en 1989
- ▶ Hypertext Markup Language
- ▶ Standards W3C (HTML5 & HTML4.01)
- ▶ Basé sur SGML
- ▶ Arrivée de HTML5 ([wikipedia](#))



# Doctype

## HTML 4.01 Strict

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

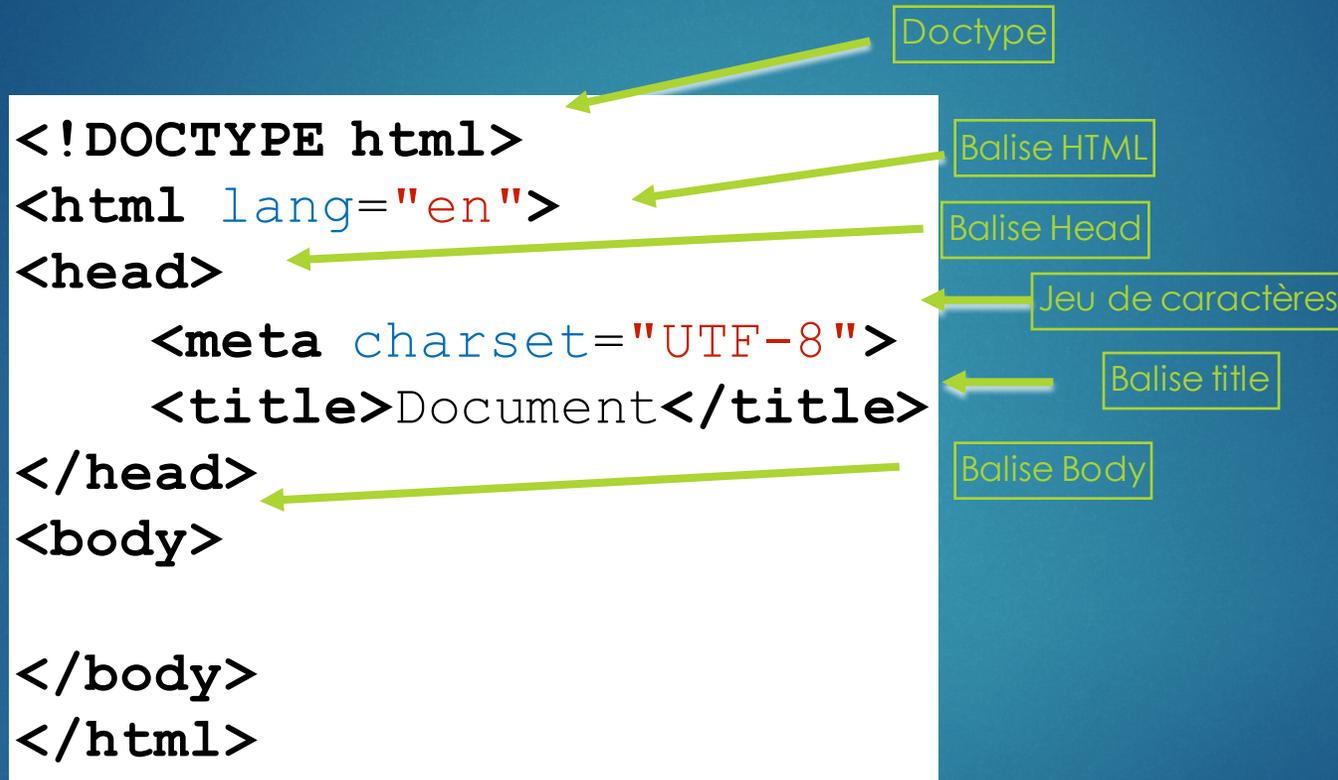
## HTML 4.01 Transitional

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

## HTML 5

```
<!DOCTYPE html>
```

# Squelette



NB : Le `<title>` est un des éléments les plus importants en SEO !

# Balises et indentation

Une balise doit toujours se fermer :

```
<html></html>
```

Pour une meilleure lisibilité, on utilisera l'indentation :

```
<head>  
  <meta charset="UTF-8">  
  <title>Document</title>  
</head>
```

# Les titres

6 niveaux de titres

```
<h1>Titre 1</h1>  
<h2>Titre 2</h2>  
<h3>Titre 3</h3>  
<h4>Titre 4</h4>  
<h5>Titre 5</h5>  
<h6>Titre 6</h6>
```

**Titre 1**

**Titre 2**

**Titre 3**

**Titre 4**

Titre 5

Titre 6

*NB : Il faut essayer de n'avoir que un H1 par page pour le SEO*

# Les paragraphes

15

**<p>**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Placeat ex nam, id maxime porro sed quasi amet, esse dolorum cupiditate consequuntur.

**<br>**

Nemo commodi aspernatur, id eius iure et. Expedita, omnis.

**<br>**

**<br>**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Deserunt ipsum

recusandae unde, nam, nemo voluptate distinctio tempore vero quos hic maxime

accusamus maiores voluptates placeat eius vitae quasi et animi?

**</p>**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Placeat ex nam, id maxime porro sed quasi amet, esse dolorum cupiditate consequuntur.

Nemo commodi aspernatur, id eius iure et. Expedita, omnis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Deserunt ipsum recusandae unde, nam, nemo voluptate distinctio tempore vero quos hic maxime accusamus maiores voluptates placeat eius vitae quasi et animi?

- Le **<p>** signifie qu'il y a un paragraphe
- Le **<br>** signifie qu'il y a un retour à la ligne

# Les listes

## Les listes à puces

```
<ul>
  <li>Puce
1</li>
  <li>Puce
2</li>
  <li>Puce
3</li>
</ul>
```

- Puce 1
- Puce 2
- Puce 3

## Les listes numérotées

```
<ol>
  <li>Puce
1</li>
  <li>Puce
2</li>
  <li>Puce
3</li>
</ol>
```

1. Puce 1
2. Puce 2
3. Puce 3

# Les liens

```
<a href="http://www.monsite.fr" title="Texte au survol" >Mon lien</a>
```

- Le *href* nous donne l'url pour accéder au site
- Le *title* nous donne le titre au survol
- Le texte dans les deux balises sera le texte affiché

```
<a href="#ancre" title="Texte au survol">Mon ancre</a>
```

- Le fait de mettre un # nous envoie directement à l'id donné

```
<a href="http://www.monsite.fr" title="Texte au survol" target="_blank">Mon  
lien dans une nouvelle page</a>
```

- Le *target \_blank* ouvre la page dans un nouvel onglet

# Les images

```

```

- L'attribut `src` indique où trouver l'image
- L'attribut `alt` indique une information alternative si l'image ne s'affiche pas. Elle est utile en référencement car les moteurs de recherche se servent de cette information pour indexer une image
- Les attributs `width` et `height` ne sont pas obligatoires, ils permettent de fixer une taille à l'image. Si on ne spécifie que l'attribut `width`, l'attribut `height` se dimensionnera proportionnellement et inversement
- L'attribut `srcset` permet de fournir une image de taille différente suivant la résolution de l'écran

# Table

```
<table border="1">  
  <tr>  
    <td>Un</td>  
    <td>Deux</td>  
    <td>Trois</td>  
  </tr>  
  <tr>  
    <td>Quatre</td>  
    <td>Cinq</td>  
    <td>Six</td>  
  </tr>  
</table>
```

Un	Deux	Trois
Quatre	Cinq	Six

- La balise `<tr>` signale une nouvelle ligne
- La balise `<td>` signale une nouvelle cellule

# Les formulaires

20

```
<form action="#" method="post">  
</form>
```

Le formulaire est en post, les informations ne seront donc pas visibles dans l'url

```
<form action="valider.html" method="get">  
</form>
```

Le formulaire est en get, les informations seront donc visibles dans l'url. Si aucune méthode n'est précisée, la méthode par défaut est le get

```
<form action="#" method="post" enctype="multipart/form-data">  
</form>
```

Le formulaire contient un type cryptage qui permet l'upload de fichiers

# Button

```
<form action="#" method="post">
  <input type="submit" value="Valider">

  <input type="button" value="Ne fait rien">

  <button type="submit">Valider</button>
</form>
```

Il existe 2 façons de faire un bouton, via :

- <button> qui est la méthode HTML5
- <input>
- Le type *submit* permet d'envoyer le formulaire
- Le type *button* sert essentiellement en javascript

# Input texte

```
<form action="#" method="post">
  <label for="name">Nom :</label>
  <input type="text" name="name" id="name">

  <input type="text" placeholder="First name" value="Alexis">

  <button type="submit">Valider</button>
</form>
```

Le label permet de lier le texte au input, ainsi le champ input a automatiquement le focus

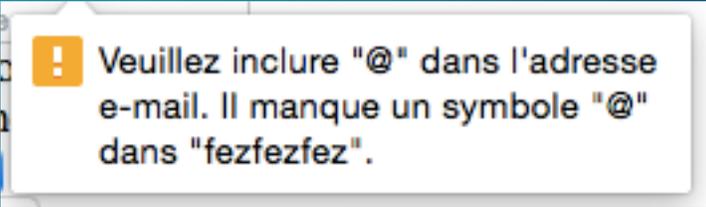
L'attribut *placeholder* permet de prévenir l'utilisateur de ce qu'il doit saisir dans le champ

L'attribut *value* donne une valeur au champ

# Input email

```
<form action="#" method="post">  
  <input type="email" name="email">  
  <button type="submit">Valider</button>  
</form>
```

Le champ de type email permet d'avoir un clavier spécial sur smartphone et fait une vérification sur l'email (sur certains navigateurs)

A white error message box with a grey border and a drop shadow. It contains a yellow warning icon (an exclamation mark inside a square) followed by the text: "Veillez inclure "@" dans l'adresse e-mail. Il manque un symbole "@" dans "fezfezfez"."

! Veuillez inclure "@" dans l'adresse e-mail. Il manque un symbole "@" dans "fezfezfez".

# Input password

```
<form action="#" method="post">  
  <input type="password" name="password"  
placeholder="Mot de passe">  
  <button type="submit">Valider</button>  
</form>
```

Le champ de type password permet de cacher le mot de passe tapé

NB : En aucun il chiffre le mot de passe !

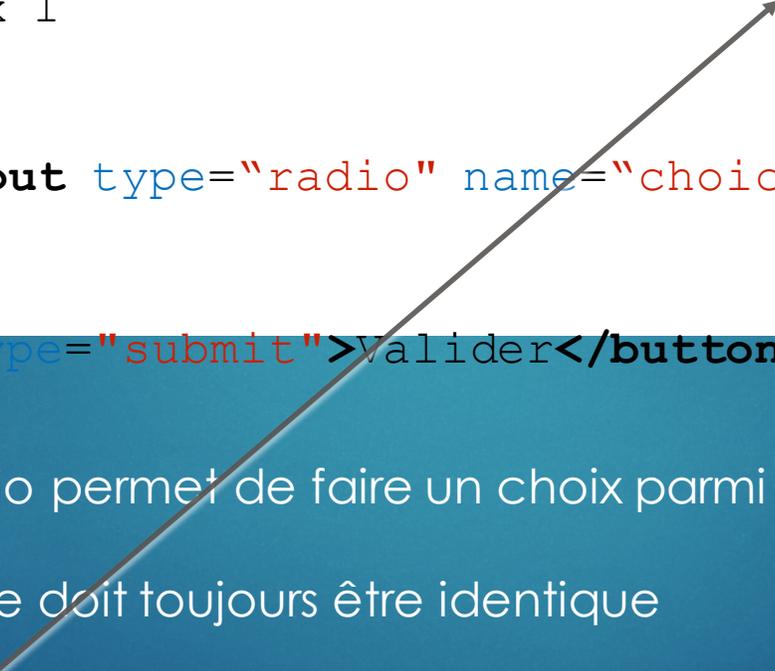
# Input checkbox

```
<form action="#" method="post">
  <label>
    <input type="checkbox" name="sport[]"
value="Football">Football
  </label>
  <label>
    <input type="checkbox" name="sport[]"
value="Basketball">Basketball
  </label>
  <button type="submit">Valider</button>
</form>
```

- Le champ de type checkbox permet d'avoir des cases à cocher
- Dans l'attribut name, il y a [] car on veut récupérer un tableau de valeurs

# Input radio

```
<form action="#" method="post">
  <label>
    <input type="radio" name="choice" value="1"
checked> Choix 1
  </label>
  <label>
    <input type="radio" name="choice" value="2">
Choix 2
  </label>
  <button type="submit">Valider</button>
</form>
```



- Le champ radio permet de faire un choix parmi plusieurs choix
- L'attribut name doit toujours être identique
- L'attribut checked permet de le mettre coché par défaut

# Textarea

```
<form action="#" method="post">  
  <textarea name="message" cols="30"  
rows="10"></textarea>  
  <button type="submit">Valider</button>  
</form>
```

L'attribut `cols` donne la largeur en colonnes et l'attribut `rows` donne la hauteur en rangées

# Select

```
<form action="#" method="post">
  <select name="age" >
    <option value="1"
selected>1</option>
    <option value="2">2</option>
    <option value="3">3</option>
  </select>
  <button type="submit">Valider</button>
</form>
```

- Le select permet de créer une liste déroulante
- Chaque choix est inséré dans une balise option
- Si un champ contient l'attribut *selected*, il sera sélectionné par défaut

# Readonly / Required

```
<form action="#" method="post">  
  <input type="text" name="champ1" required>  
  <input type="text" name="champ2" readonly>  
</form>
```

- L'attribut *required* informe l'utilisateur que le champ est obligatoire, il ne fonctionne pas sur l'ensemble des navigateurs
- L'attribut *readonly* rend le champ impossible à modifier

# HTML 5

- ▶ Structure de page : `<section>`, `<article>`, `<nav>`, `<aside>`, `<header>`, `<footer>`
- ▶ Audio / vidéo : `<audio>` et `<video>`
- ▶ Dessin : `<canvas>`

# Compatibilité

31

- ▶ Normalize.css (à placer dans la balise <head>)

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/normalize/3.0.3/normalize.min.css">
```

- ▶ HTML5Shiv (à placer en fin de balise <body>)

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.min.js "></script>
```

- ▶ Vérifier sur [CanIUse.com](http://CanIUse.com)

CSS



CSS

# Feuille de style Interne/externe

33

- ▶ Feuille de style interne au fichier HTML, compris dans une balise `<style>`
- ▶ Feuille de style externe, appelé dans une balise `<link>`

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/normalize.min.css">
```

- ▶ Style en *inline* directement sur une élément HTML

```
<div style="background-color:red;">Display block</div>
```

# Id / class

```
<style>
  #monId {
    background-color: red;
  }

  .maClasse {
    background-color: green;
  }
</style>
```

- ▶ Un *ID* est unique dans une page html et s'appelle via un #

```
<div id="monId"></div>
```

- ▶ Une *classe* peut être utilisé plusieurs fois sur une page html et s'appelle via un .

```
<div class="maClasse"></div>
```

- ▶ L'ordre de priorité est *balise>id>class*

# Display

35

```
<div style="background-color:red;">Display block</div>  
Display <span style="background-color:yellow">inline</span>
```

Display block  
Display inline

- ▶ Un display *block* prendra toute la place qui est disponible. C'est le comportement natif de la balise `<div>`
- ▶ Un display *inline* prendra seulement la place qui est nécessaire. C'est le comportement natif de la balise `<span>`

# float

36

```
<span style="float:left;background-color:red;">Float left</span>  
<span style="float:right;background-color:blue;">Float right</span>
```

Float left

Float right

- ▶ L'attribut float permet de mettre un élément à gauche ou à droite
- ▶ Pour le libérer il faut placer un élément avec un *clear:both*

# Styliser un élément

37

```
body {  
    background-color: #d0e4fe;  
}  
  
h1 {  
    color: orange;  
    text-align: center;  
}  
  
p {  
    font-family: "Times New Roman";  
    font-size: 20px;  
}
```

- Le code couleur est en hexadécimal
- La taille de texte est en pixels
- Une feuille CSS s'exécute de haut en bas
- Une ligne se termine toujours par un ;

# CSS3

38

CSS3 apporte beaucoup d'effet d'animation et d'améliorations grâce entre autres aux :

- ▶ transition
- ▶ keyframe
- ▶ border-radius
- ▶ text-shadow
- ▶ perspective

Mais aussi grâce à de nouveaux sélecteurs tel que `:nth-child()`, `:first-child()`, `:last-child()`, `:checked`, `:empty`, +

# Liens utiles

39

- ▶ <https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3>
- ▶ <https://www.codecademy.com/fr/tracks/web>
- ▶ <http://www.grafikart.fr/tutoriels/html-css>