

Développement Web

Pr. Fatima Zahra MOUTAI

Planning

- **Rappel au développement Web**
- **Introduction à JavaScript**
- **Programmation avancée en JavaScript**
- **Introduction à TypeScript**
- **Frameworks et bibliothèques JavaScript**
- **Développement d'applications web avec TypeScript**

Chapitre 1 : Rappel au développement Web

Introduction

- **La programmation web** est la programmation informatique qui permet de créer des sites web.
- **Un site web** est un ensemble de pages web reliées par des hyperliens, défini et accessible par une adresse web.
- **Une page web** désigne le contenu visible dans la fenêtre d'un navigateur.
- **Un serveur web** est capables de fournir des ressources pour les pages Web. .

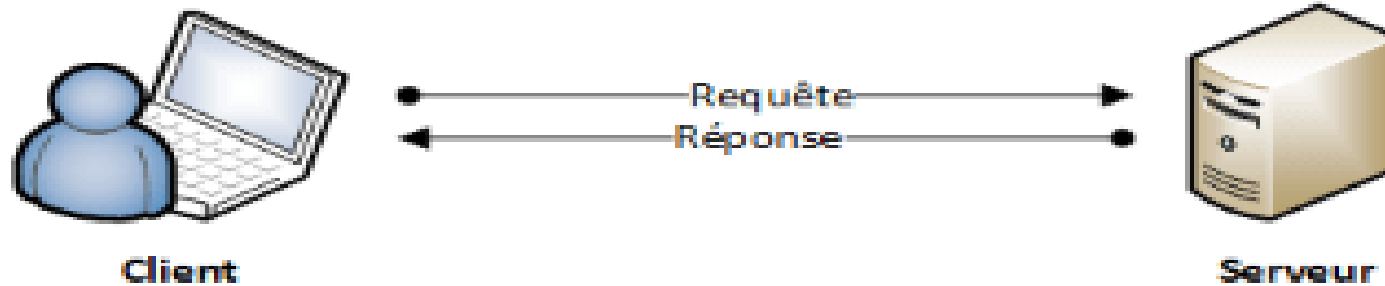
Internet

- Internet signifie interconnexion de réseaux.
- Les réseaux peuvent être de natures très diverses : réseau routier, réseau téléphonique, réseau radio ...
- Un réseau informatique est un ensemble de moyens matériels et logiciels permettant de connecter différents utilisateurs.
- Internet est un regroupement d'un ensemble de réseaux :
 - **réseaux personnels ou PAN (Personal Area Network).**
 - **réseaux locaux (LAN=Local Area Network) quelques m à quelques km à débit élevé.**
 - **MAN (Metropolitan Area Network) jusqu'à 100 km.**
 - **réseaux étendus (WAN=Wide Area Network) souvent internationaux.**
- Internet est le plus grand réseau informatique du monde.

Web

- Il ne faut pas confondre le Web (ou WWW pour World Wide Web) et Internet !
- Internet est un réseau physique.
- Le Web n'est qu'une des applications d'Internet.
- Le Web est un ensemble de ressources (les pages Web) réparties sur différents serveurs du réseau Internet.
- Le Web permet de consulter, avec un navigateur, des pages accessibles sur des sites.

Architecture client-serveur



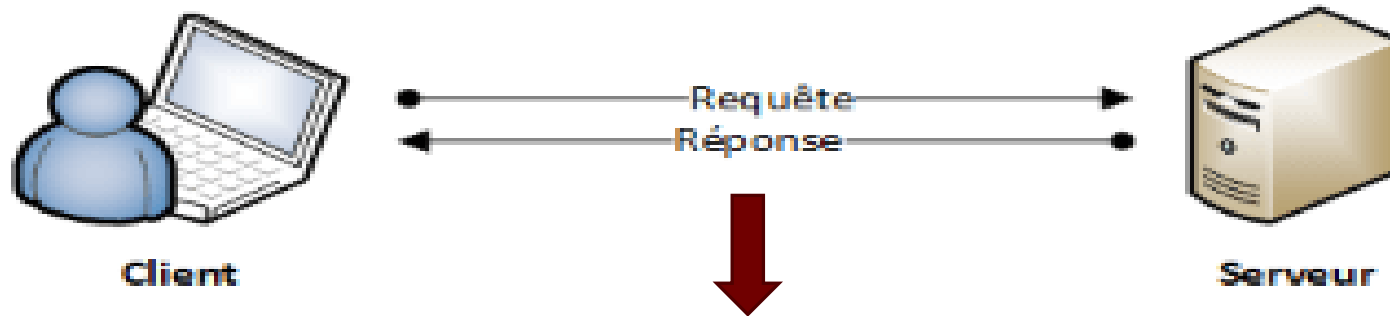
Ordinateur qui a le droit d'accès à la ressource sur le serveur

- Il est actif (ou maître),
- Il envoie des requêtes au serveur,
- Il attend et reçoit les réponses du serveur.

Ordinateur sur lequel se trouve une ressource.

- Il attend, il est à l'écoute, prêt à répondre aux requêtes envoyées par des clients.
- Dès qu'une requête lui parvient, il la traite et envoie une réponse.

Architecture client-serveur



- Le client et le serveur doivent bien sûr utiliser le même protocole de communication.
- Le dialogue entre le client et le serveur repose sur toute une série de **protocoles** les uns dans les autres.

HTTP

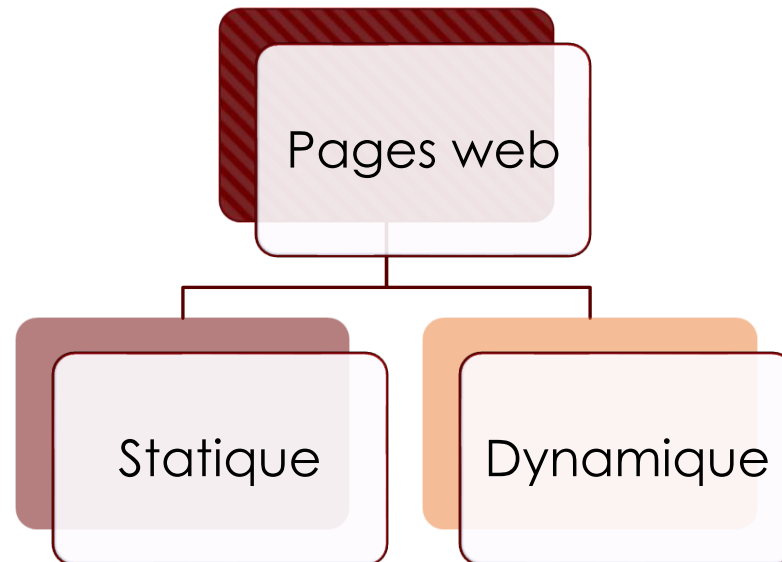
HTTP (Hyper-Text Transfert Protocole)

- Protocole qui permet au client de récupérer des données du serveur.
- Ces données peuvent être statiques (contenu qui ne change pas : HTML, PDF, Image, etc..) ou dynamiques (Contenu généré dynamiquement au moment de la requête : PHP, JSP, ASP...).
- Ce protocole permet également de soumissionner les formulaires.

Fonctionnement :

- Le client se connecte au serveur.
- Le client demande au serveur un document : Requête HTTP.
- Le serveur renvoi au client le document (status=200) ou d'une erreur (status=404 quand le document n'existe pas).
- Déconnexion

Classification des pages web



Le contenu de la page est interprété directement de la part du navigateur.

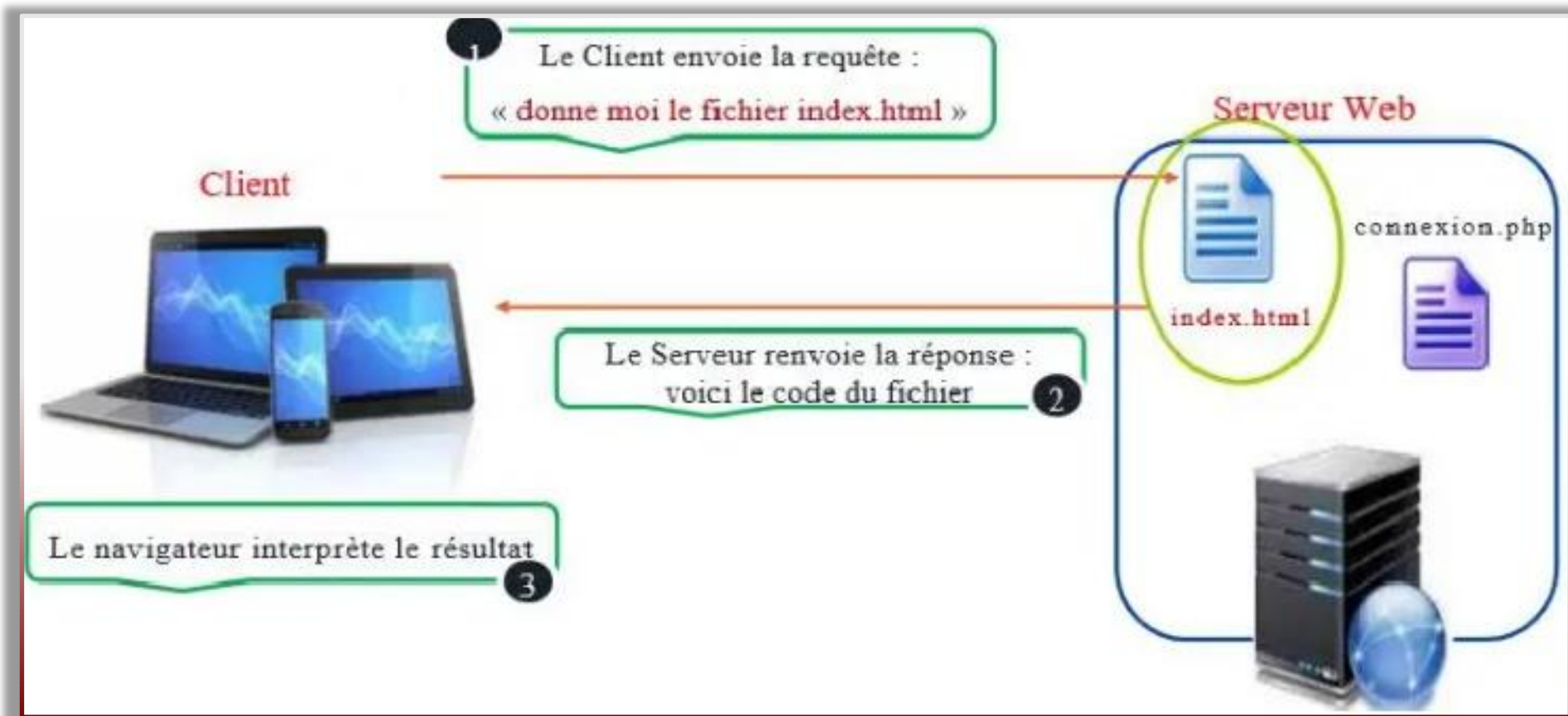
Le contenu de la page peut évoluer dans le temps. Des programmes tournent du côté des serveurs, à l'arrière plan (back-end) pour générer les pages du site

Page

Page statique

- Une page Web statique est une page web dont le serveur sur laquelle est stockée, ne fait aucun effort pour la créer lorsqu'elle est demandée par la machine.
- Une page web statique est simple. Elle a une mise en page uniforme.
- Certains l'appellent la page par défaut d'Internet.
- Les technologies pour créer un site statique sont **HTML**, **CSS**, et parfois JavaScript.
- Porte l'extension **.html**
- Il ne nous faudra pas beaucoup de temps, d'efforts et d'argent pour créer un site statique.

Page statique



Page dynamique

- Une page dynamique est une page dont le contenu n'existe pas en tant qu'il est sur le serveur mais qui est créée seulement lors de sa demande.
- Elle vise à interagir avec les utilisateurs pour répondre à ce qu'ils recherchent en consultant un site.
- Elle est construite en utilisant des scripts côté client et côté serveur. Elle est écrite dans des langages tels que **ASP.NET, PHP et JavaScript**.
- Il s'agit de langages de programmation plus compliqués qui peuvent nécessiter l'intervention de développeurs.
- Ces langages complexes, eux, permettent à un site dynamique d'exécuter des fonctions en temps réel.

Page dynamique

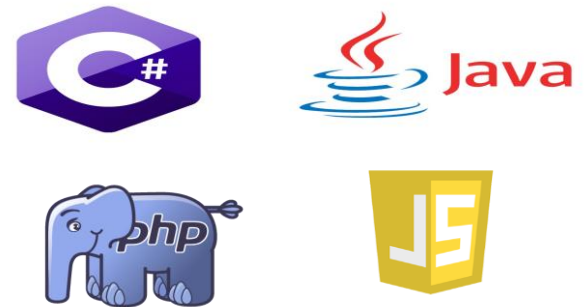


Page dynamique

Langages qui
s'exécutent côté Client



Langages qui
s'exécutent côté Serveur



HTML et CSS

- **HTML** et **CSS** sont deux langages de description de pages.
- Pour créer un site Web il ne suffit pas de taper du texte il faut également positionner le texte, l'enrichir de contenus : multimédia, de liens hypertexte ...
- Pour cela on utilise deux langages, **HTML** et **CSS** qui se complètent :
 - **HTML (HyperText Markup Language, 1991)** : gestion et organisation du contenu.
 - **CSS (Cascading Style Sheets ou Feuilles de style, 1996)** : gestion de l'apparence de la page web (positionnement, couleurs, polices de caractère ...).
- En gros HTML gère le fond et CSS la forme !

HTML histoire

- HTML 1 : créée en 1991 par Tim Berners-Lee.
- HTML 2 : 1994-1996 créée par le W3C qui gère dorénavant les versions suivantes.
- HTML 3 : à partir de 1996 ajoute de nouvelles fonctionnalités (tableaux, applets, scripts, habillage des images par du texte ...).
- HTML 4 : à partir de 1998. Introduction des frames, permet d'exploiter les feuilles de style CSS.
- HTML 5 : the last one. Amélioration de la version précédente pour une mise en œuvre enrichie et facilitée.

CSS histoire

- CSS 1 : 1996. Les premiers pas pour la présentation de pages web : couleurs, marges, polices de caractères ...
- CSS 2 : 1999. Ajout de nombreuses options permettant de positionner avec précision les différents éléments d'une page.
- CSS 3 : the last one. Encore plus de design : bordures arrondies, dégradés, ombres ...

Éditeur de texte

- Pour créer son site Web il faut un logiciel. Il en existe deux grandes catégories :
- **Les WYSIWYG (What You See Is What You Get)** : Ils se comportent comme des logiciels de traitement de texte : pas besoin de connaître HTML ni CSS pour créer son site Web. Les plus connus sont Mozilla Composer, Microsoft Expression Web, Dreamweaver, KompoZer ...
 - Problèmes : génération de code HTML et CSS de mauvaise qualité, aucune maîtrise du code.
- **Les éditeurs de texte** : Non spécifiques à HTML et CSS mais facilitent le développement : Sublime Text, Notepad++, Kwrite, gedit, Visual Studio Code, ...

Navigateur

- Un navigateur permet de voir les sites Web.
- Il en existe de nombreux : Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari ...
- Attention les pages n'apparaissent pas de la même façon selon le navigateur !

HTML

- **HyperText Markup Language**

- **HyperText** : fonction qui permet de passer d'un document à l'autre grâce à un système de liens. Ce système est à la base du fonctionnement d'Internet.
- **Markup Language** : un langage à balises.
- HTML est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS)

HTML

- **Balises en paires**

- balise d'ouverture + texte + balise de fermeture
- Exemple : `<title>` Ceci est le titre de ma 1ère page Web `</title>`

- **Balises orphelines**

- Ne contiennent pas de texte, n'ont pas de balise de fermeture
- Utilisées pour insérer un élément à un endroit précis de la page
- Exemple : `
`, ``

- **Balises avec attributs**

- Une balise peut avoir des attributs avec ou sans valeurs
- Exemple : `<input type="text" id="name" name="name" />`

Balises structurelles

Balises de base :

- `<!DOCTYPE>` : type du document.
- `<html></html>` : balise principale englobant le contenu de la page Web.
- `<head></head>` : en-tête du document.
- `<meta />` : permet de rajouter d'autres informations concernant le document. Balise non obligatoire. Cette balise peut avoir plusieurs attributs tels que :
 - `charset` : type d'encodage de la page
 - `name` : a généralement la valeur author, description ou keywords. S'il existe il est suivi de l'attribut content.
 - `content` : bref description de ce qui est indiqué dans name.
- `<title></title>` : titre de la page Web.
- `<body></body>` : contenu principal de la page Web.

Balises structurales

Cas particulier des commentaires :

- `<!-- ceci un commentaire -->`

Caractères spéciaux :

- `<` , `>` , `&` et `"` : `<` , `>` , `&` , `"` ; ...
- Espace : ` ` ;
- Caractères accentués : `é`(`é`) , `è`(`è`) , `ê`(`ê`) , `à`(`à`) , ...

Balises structurales

Balises de regroupement sans sémantique :

- `<div></div>` : bloc de texte, généralement utilisé pour découper un document en bloc.
- `` : bloc plus petit, généralement utilisé pour découper un paragraphe en sous-bloc.

Balises de regroupement avec sémantique :

- `<article>` : pour indiquer que c'est l'article d'un blog ou d'un journal.
- `<section>` : section de contenu, utilisé si l'on a plusieurs sections d'un document.
- `<header>` : en-tête de la page!
- `<footer>` : pied de page.
- `<details>` : pour donner plus de détails.
- `<summary>` : utilisé pour écrire le résumé de l'article.

Balises textuelles

Balises titres/sous-titres :

- `<h1></h1>` : affiche un en-tête de niveau n et saute une ligne.
- `<h1>`, `<h2>`, `<h3>`, `<h4>` : h1 étant le niveau de titre le plus haut et h4 le sous-titre le plus bas.

Balises pour un bloc de texte :

- `<p></p>` :paragraphe
- `
` ou `</br>` : saut de ligne
- `<dfn></dfn>` : définition
- `<samp></samp>` :exemple

Balises textuelles

Balises pour mise en valeur :

- `` : mettre texte italique
- `` : mettre texte gras
- `<mark></mark>` : surligner texte

Autres balises :

- `<abbr></abbr>` : abréviation
- `<address></address>` : contenu de type adresse email
- `<cite></cite>` : citation

Balises listes

Liste non ordonnée :

- `` : indique que c'est une liste non ordonnée.
- `` : définit les différents éléments de la liste

- **Exemple :**

``

`` 1er élément de la liste ``

`` 2nd élément de la liste ``

``

Balises listes

Liste ordonnée :

- `` : indique que c'est une liste ordonnée

- **Exemple :**

- ``

- `` 1er élément de la liste ``

- `` 2nd élément de la liste ``

- ``

Balises listes

Liste de définitions :

- `<dl></dl>` : indique que c'est une liste de définitions
- `<dt></dt>` : indique le terme à définir
- `<dd></dd>` : donne la définition du terme

- **Exemple :**

`<dl>`

`<dt>` 1er terme à définir `</dt>`

`<dd>` définition du 1er terme `</dd>`

`<dt>` 2ème terme à définir `</dt>`

`<dd>` définition du 2ème terme `</dd>`

`</dl>`

Balises listes

Liste de définitions :

- `<dl></dl>` : indique que c'est une liste de définitions
- `<dt></dt>` : indique le terme à définir
- `<dd></dd>` : donne la définition du terme

- **Exemple :**

`<dl>`

`<dt>` 1er terme à définir `</dt>`

`<dd>` définition du 1er terme `</dd>`

`<dt>` 2ème terme à définir `</dt>`

`<dd>` définition du 2ème terme `</dd>`

`</dl>`

Balises table

Tableau simple :

- Tableau se construit ligne par ligne
- Chaque ligne doit contenir la liste des cellules
 - `<table></table>` : indique que ce qui suit est un tableau
 - `<caption></caption>` : indique titre du tableau
 - `<tr></tr>` : indique que c'est une ligne du tableau
 - `<th></th>` : indique que c'est l'en-tête du tableau (1ère ligne du tableau)
 - `<td></td>` : indique que c'est une cellule/case du tableau

Balises table

Tableau structuré :

- Concerne les tableaux complexes.
- Division en 3 parties : en-tête, corps et pied du tableau.
 - `<thead></thead>` : indique que c'est l'en-tête du tableau
 - `<tbody></tbody>` : indique que c'est le corps tableau
 - `<tfoot></tfoot>` : indique que c'est le pied du tableau
 - `<th></th>` : indique que c'est l'en-tête du tableau (1ère ligne du tableau)
 - `<td></td>` : indique que c'est une cellule/case du tableau

Fusion de cellules

- Fusion Horizontale : fusion de colonnes en utilisant **colspan**
- Fusion Verticale : fusion de lignes en utilisant **rowspan**

Balises figures

Figures :

- Une Figure peut correspondre à une image, un code source, une citation,...
- **<figure></figure>** : indique qu'il y a une figure
- **<figcaption></figcaption>** : titre de la figure
- Exemple pour une image

```
<figure>
```

```
  <img src= "image.png " />
```

```
  <figcaption> Figure 1 </figcaption>
```

```
</figure>
```

Balises formulaires

Objectif :

- Permettre aux utilisateurs d'une page Web de fournir des informations en tapant le texte, sélectionnant des options, validant des boutons, etc.
- Ces balises ne permettent pas de traiter les résultats et les informations saisies. Cela sera fait en utilisant un langage comme PHP.

Balise de base pour création d'un formulaire

- `<form></form>` : indique que l'on va insérer un formulaire.

Balises formulaires

Attributs de base :

- **method** : indique comment envoyer les informations.
 - **post** : méthode la plus utilisée.
 - **get** : moins utilisée + limitée à 255 caractères
- **action** : indique l'adresse de la page ou du programme qui va traiter les informations. La page chargée lorsque l'utilisateur aura appuyé sur le bouton d'envoi du formulaire

Zones de textes

Types de zone de texte :

- Zone de texte courte : pour saisir une ligne
- Zone de texte longue : pour saisir un paragraphe/plusieurs lignes

Balise pour zone de texte monoligne :

- `<input />` : indique que l'on va créer une zone de texte monoligne.
- `<label></label>` : indique le libellé de la zone de texte.

Zones de textes

Exemple :

`<p>` On peut mettre du texte avant le formulaire `</p>`

```
<form method = ''post'' action= ''traitement.php'' >
```

```
  <label for= ''nom'' > Votre Nom </label>
```

```
  <input type= ''text'' name= ''nom'' />
```

```
  <label for= ''prenom'' > Votre prénom </label>
```

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

```
</form>
```

`<p>` On peut mettre du texte après le formulaire `</p>`

Zones de textes

Balise pour zone de texte multilignes :

- `<textarea></textarea>` : indique que l'on va créer une zone de texte multilignes
- `<label></label>` : indique le libellé de la zone de texte

Attributs pour zone de texte multilignes :

- **name** : nom de la zone de texte
- **id** : identifiant de la zone de texte pour lier le label à la zone de texte
- **for** : aura la même valeur que id de la zone de texte pour lier le label
- **rows** : nombre de lignes de texte qui seront affichées
- **cols** : nombre de colonnes

Zones de textes

Exemple :

```
<form method = ''post'' action= ''traitement.php'' >  
  <label for= ''cours'' > Pourquoi voulez vous suivre le cours de Java ? </label>  
  <textarea name=''cours'' id=''cours'' rows=''20'' cols=''8'' >  
    Expliquez en quelques lignes pourquoi vous voulez suivre de cours de Java ?  
  </textarea>  
</form>
```


Zones de textes

Autres valeurs pour l'attribut "type"

- **type="password"** : pour cacher le texte tapé pour un mot de passe
- **type="email"** : pour saisir une adresse email
- **type="url"** / **type="tel"** : pour saisir une URL / un numéro de téléphone
- **type="search"** : pour dire que c'est un champ de recherche
- **type="color"** : pour saisir une couleur
- **type="date"** / **type="time"** : pour saisir une date / un horaire
- **type="datetime"** / **type="datetime-local"** : pour date et heure avec décalage horaire / sans décalage horaire
- **type="week"** / **type="month"** : pour saisir une semaine / un mois

Styles CSS

- un ensemble de règles qui indiquent au navigateur :
 - comment il doit afficher les différents éléments de la page (**design**)
 - comment ces éléments se positionnent les uns par rapport aux autres (**layout**)
- 3 manières de définir des styles :
 - **inline CSS** : directement dans les éléments HTML de la page
 - **CSS internes** : en utilisant des balises
 - **CSS externes** : en liant au document HTML, une feuille de style (fichier.css) externe

Styles CSS

- **Style inline** défini directement dans une balise à l'aide d'un attribut **style**

The diagram illustrates inline CSS styles. On the left, an HTML code editor shows the following code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>O1 inline</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <h1>
      Titre sans style
    </h1>
    <h1 style="color:blue; background:yellow; padding: 4px; border:solid 3px green;">
      Titre avec style
    </h1>
    <h1>
      Titre sans style
    </h1>
  </body>
</html>
```

Annotations on the code:

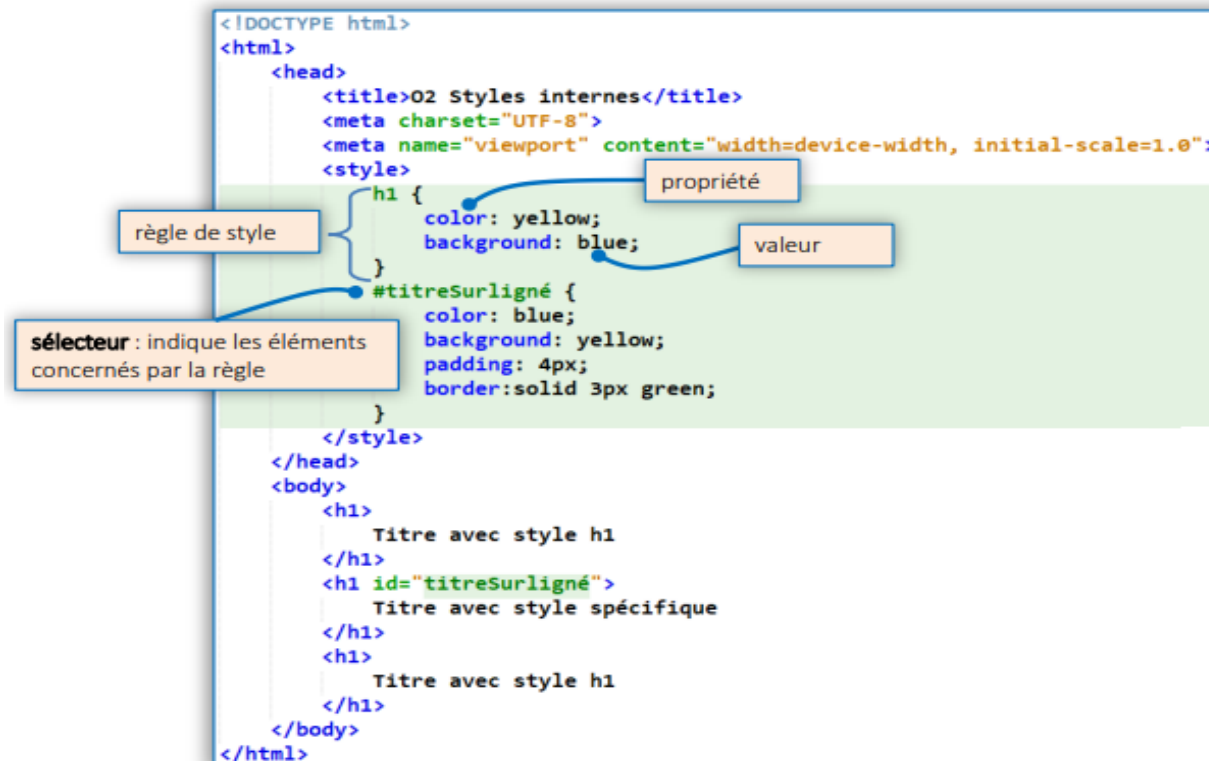
- A bracket labeled "attribut style" points to the `style` attribute in the second `<h1>` tag.
- A bracket labeled "valeur de l'attribut style" points to the entire value of the `style` attribute.
- Two arrows labeled "propriété CSS" and "valeur" point to `background:yellow` and `padding: 4px` respectively within the style value.

On the right, a browser window titled "O1 inline" shows the rendered output:

- The first `<h1>` tag renders as "Titre sans style" in black text.
- The second `<h1>` tag renders as "Titre avec style" in blue text, with a yellow background, a green border, and padding.
- The third `<h1>` tag renders as "Titre sans style" in black text.

Styles CSS

- **Styles internes** définis dans un élément style placé dans l'en tête (**head**) du document



Styles CSS

- **Styles externes** Styles définis dans un élément **style** placé dans l'en tête (**head**) du document

The diagram illustrates the use of external CSS styles. It shows two HTML files, `maPage.html` and `monAutrePage.html`, both linking to a common external CSS file, `meStyles.css`.

maPage.html code snippet:

```
<!DOCTYPE html>
<html>
  <head>
    <title>O3 Styles externes</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="css/mesStyles.css" rel="stylesheet" type="text/css"/>
  </head>
  <body>
    <h1>
      Titre avec style h1
    </h1>
    <h1 id="titreSurligné">
      Titre avec style spécifique
    </h1>
    <h1>
      Titre avec style h1
    </h1>
  </body>
</html>
```

meStyles.css code snippet:

```
h1 {
  color: yellow;
  background: blue;
}
#titreSurligné {
  color: blue;
  background: yellow;
  padding: 4px;
  border: solid 3px green;
}
```

monAutrePage.html code snippet:

```
<!DOCTYPE html>
<html>
  <head>
    <title>O3 Styles externes</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="css/mesStyles.css" rel="stylesheet" type="text/css"/>
  </head>
  <body>
    <h1>
      Titre avec style h1
    </h1>
  </body>
</html>
```

The diagram also shows two browser screenshots. The first screenshot, labeled `maPage.html`, displays three `h1` titles: "Titre avec style h1" (blue background, yellow text), "Titre avec style spécifique" (yellow background, blue text, green border), and "Titre avec style h1" (blue background, yellow text). The second screenshot, labeled `monAutrePage.html`, displays one `h1` title: "Titre avec style h1" (blue background, yellow text).