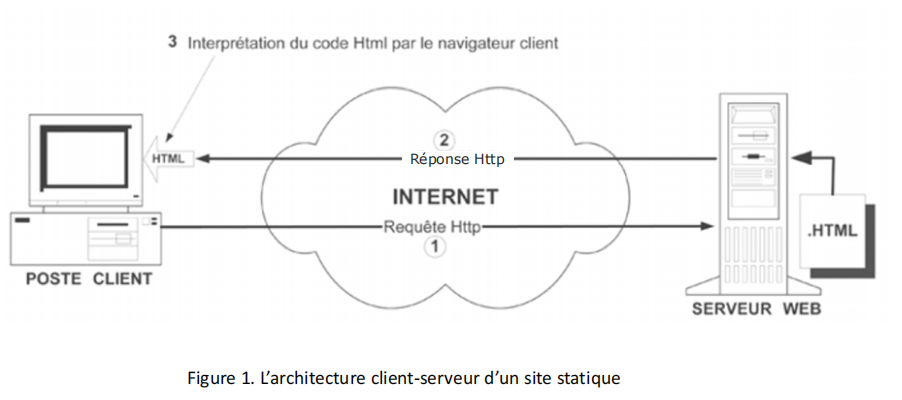
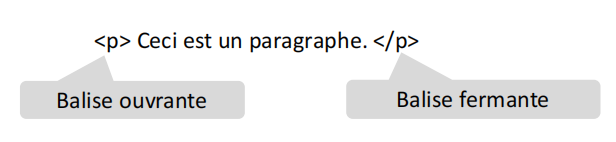
**URL (Universal Resource Locator):** adresse d’une page html situé sur un serveur web.

**Http (HyperText Transfert Protocol) :**



**HTML (HyperText Markup Language) :** langage de balisage permettant de structurer les page web, se base sur des balises permettant aussi d’introduire de texte, des formulaires, des ressource multimédias, et d’autre chose davantage.



Élément paragraphe signifie la balise <p>, ce qui est á l’intérieur de balise ouvrante et fermante est le contenu de l’élément HTML <p> (value).

**Balise auto-fermante (orpheline) :** c’est une balise qui n’a pas besoin d’une balise fermante, et qui généralement ne possède pas de contenu particulier, telle que la balise <br> de retour á la ligne.

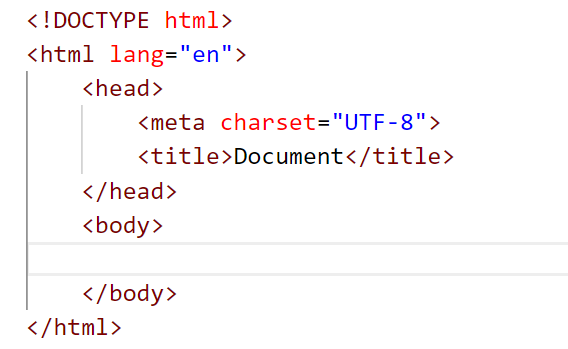
**Les attribues d’une balise :** permet de modifier les propriétés ou bien le comportement d’une balise.

<balise attribute1 = ‘’valeur1’’ attribut2 = ‘’valeur2 ></balise>

Une balise peut se trouver á l’intérieur d’une autre, on dit sur la balise englobante l’élément parent, la balise interne l’élément enfant.

Commentaire en HTML 🡪 < !-- Ceci est un commentaire -->

Le document HTML : se constitue d’un en-tête (header) et un corps (body)



<!DOCTYPE html> permet d’indiquer au navigateur qu’il s’agit d’un document HTML5 á délivrer.

<html></html> est l’élément racine de tout document HTML.

<head></head> : fournit multiple renseignements sur le document, comme le titre de la page, le codage utilisé dans le document, ainsi qu’il permet aussi d’inclure des fichier Css, Js

<meta charset = ‘’utf-8’’> 🡪 permet de définir l’encodage utiliser sur la page HTML, utf-8 support les caractères de tous les langages du monde.

<meta http-equiv = ‘’refresh’’ content = ‘’60’’> 🡪 rafraichir le contenu de la page chaque minute.

**Quelque balise permettant la structuration du texte :**

<b> </b>🡪 mis-en-forme de texte en gras (bold).

<i></i> 🡪mis-en-forme de texte en italique.

<u></u>🡪 souligne le texte.

<mark></mark> mettre en valeur le texte (fond d’écran jeune), utiliser pour maquer de texte pour l’utiliser sous CSS.

<em></em> maquer de texte pour l’utiliser sous CSS (italique par défaut).

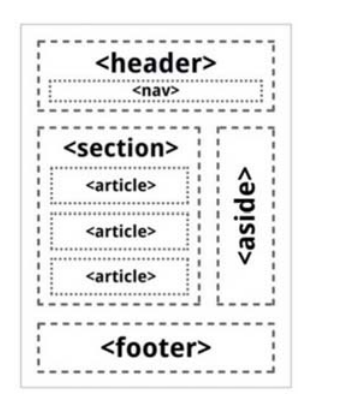
<strong></strong> , <small></small> 🡪 augmenter, diminuer la taille de texte.

<sub></sub> 🡪 O<sub> 2 </sub> 🡪  O2

<sup></sup 🡪  a <sup>2</sup> 🡪 a 2

<hr> 🡪 séparation horizontal.

**Structurer la page Web :**



Toutes ces balises sont introduit dans la balise body.

<div></div> : balise qui n’as pas de valeur sémantique, mais qui est utilisé pour regrouper des éléments pour lesquels on veut appliquer un style commun.

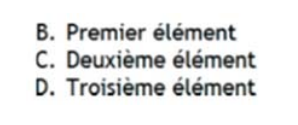
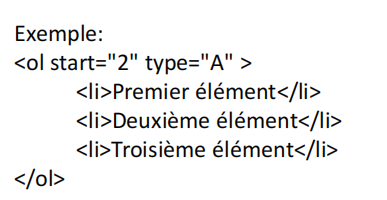
<span></span> : comme <div> mais il ne fait par un retour á la ligne.

**Les listes :**

<ul></ul> : Définir une liste ou l’ordre n’est pas important.

<ol></ol> : Définir une liste ou l’order est importante.

<li></li> : constitue les éléments d’une liste 🡪 enfant de ul ou ol.

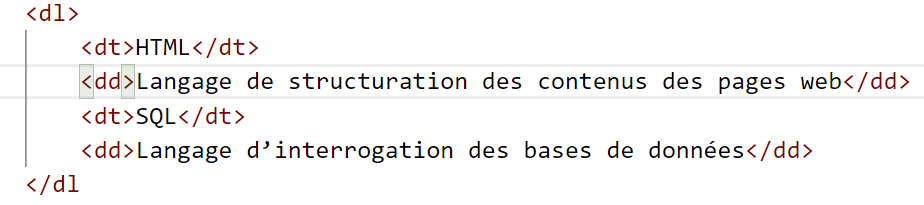


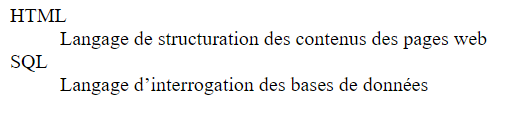
**Liste de définition :**

<dl></dl> liste de définition :

<dt></dt> titre á définir 🡪 élément enfant de dl.

<dd></dd> contenu de la définition 🡪 enfant de dl, mais suivant d’un dt.





Balises de création de tableau :

<table><table> : balise d’ajout d’un tableau sous HTML .

<tr><tr> : permet d’ajouter une ligne dans un tablau HTML <table>.

<td></td> : permet d’ajouter une colonne dans une ligne tr (enfant de tr).

<caption> : associés une légende á un tableau donné.

**Des balises de tableau optionnel** :

<thead></thead> : optionnel, mais qui englobe la première ligne du tableau , first <tr>.

<tbody></tbody> : englobe les ligne de tableau.

<tfoot></tfoot> : englobe la dernière ligne.

Pour fusionner 2 colonnes <td> on utilise un seul mais avec l’attribut colspan = ‘’2’’.

Pour fusionne 2 ou plus linge sur une colonne <td>, on utilise un seul mais avec l’attribut rowspan = ‘’2’’.

**Les lien hypertextes :** permettant par une simple clique de visiter un autre contenu web, que ce soit sur le même serveur ou á l’extérieur dans le réseau.

<a>mon lien</a> 🡪 permet de crée un lien hypertext cliquable.

href : attribut qui indique le ressource cible.

target : attribut qui prend un certaines valeurs, blank ( ouvrir le lien dans une nouvelle fenêtre).

download = ‘’video.mp4’’ 🡪 permet de dire que le lien est un fichier téléchargeable et que video.mp4 son nom.

<a href= ‘’mailto :boussaidmohammed88@gmail.com’’>, le protocole mailto lance l’application de messagerie installer sur la machine en proposant une fenêtre de rédaction.

Les liens ancres : ce sont des liens qui mène vers des sections de la même page, il suffit d’ajouter l’attribut id á la balise qu’on souhaite visiter, ensuite en crée le lien <a> avec le href = ‘’#id’’.

<a href="page2.html#para1"> 🡪 visiter la section de l’id = para1 de la page html page2.html.

**Les balises multimédias :**

<img src= ‘’photo.jpg’’ alt = ‘’fsdmLogo’’ width = ‘’100’’ height = ‘’100’’>

src indique l’emplacement de l’image sur le disque, alt c’est ce que sera afficher au cas où l’image n’est pas chargé,

<audio src="musique.mp3" controls></audio> permet de lire un fichier audio, l’attribut

<video src=" video1.mp4" width="640" height="480" controls></video permet de lire un fichier video.

**Attributs applicable sur les audios et videos :**

controls : afficher les buttons de contrôle

autoplay : active la lecture dès le chargement de la page.

loop : lecture en boucle.

mediagroupe : grouper un ensemble d’éléments multimédias sous une seul liste de lecture.

<track></track> enfant d’élément audio et video, permet d’introduire des subtibles, des description et bien d’autre chose sur le fichier audio/video….

**Les formulaires :** utilisé pour envoyer des informations du client au serveur qui est á l’adresse définit dans l’attribut action, l’envoie ce fait par la méthode get ou post qui doit être spécifier dans l’attribut method, pour envoyer des fichiers on ajouter l’attribut enctype = ‘’multipart/form-data’’.

***Les élément d’un formulaire :***

<fieldset> <legend> Votre identité </legend> <ensemble des input> </fildset> 🡪 permet de grouper plusieurs éléments du formulaire sous le même contexte.

Les types d’input :

1. text : introduire du simple text.
2. password : utiliser pour cacher la saisie de mot de passe.
3. date : choisir une date á l’aide d’un calendrier.
4. color : choisir un couleur á l’aide d’un contrôle de sélection de couleur.
5. number : pour saisir un nombre situer entre 2 valeurs spécifier par l’attribut min et max.
6. range : propose d’entrer d’un nombre mais á l’aide d’un contrôleur glissable entre min et max avec une attribut step.
7. tel : affiche un clavier des nombres pour faciliter la saisie d’un nombre de téléphone.
8. email : pour la saisir d’un email, un message d’erreur s’affiche au cas où l’email n’est pas valide.
9. hidden : zone cacher de l’utilisateur, mais dont la valeur est validé par le formulaire.

10-radio : bouton á cocher, un seul bouton radio peur être cocher, cela est fait par le regroupement de tous les bouton radio sous le même attribut name, mais avec attribut value différentes.

11-checkbox : bouton á cocher, permet de sélectionner des éléments parmi une liste, attribut name présente le nom de la variable, et que value présente sa valeur.

12-reset : bouton qui réinitialise tous les champs du formulaire.

13-submit : bouton de validation du formulaire, il renvoie une requête au serveur contenant les informations saisies comme des variables et ses valeurs.

14-file : pour sélectionner un fichier local á partir de la machine, attribut multiple pour sélectionner plusieurs fichier d’un seul coup, accepte pour définir les types MMIE qui seront accepter lors de traitement.

Les Attributs d’input :

name : présente le nom de la variable avec lequel on récupéré sa valeur au côté serveur.

value : la valeur de champ, il peut la spécifier dans checkbox et radio.

<textarea > ici du texte</textarea> : zone de texte multiligne, attribut rows précise le nombre de ligne, l’attribut cols précise le nombre de colonnes.

<select name = ‘’liste1’’ ></select> 🡪 pour la sélections des éléments dans une liste déroulante, chaque élément de la liste et délimité par la balise <option value = ‘’e1‘’ ></option>, l’attribue multiple sert pour sélectionner plusieurs éléments, et l’attribut selected permet de sélectionner un élément par défaut.

<optgroup label = ‘’ville de maroc’’ ></optgroup> 🡪 sert de regrouper plusieurs élément d’une liste (<option>) sous le même contexte.

<button type="button">OK</button> 🡪 pour ajouter un bouton simple.

<progress id = ‘’indicateur’’ vale = ‘’50’’ max = ‘’100’’></progress> 🡪 un barre qui peut sert á indiquer le progresse d’une certaine opération.

**Attributs commun entre les éléments de formulaire :**

disable : désactive le contrôle de l’élément, état grise.

autocomplete : active ou désactive l’auto complétion.

readonly : l’attribut ne peut pas être modifier par l’utilisateur.

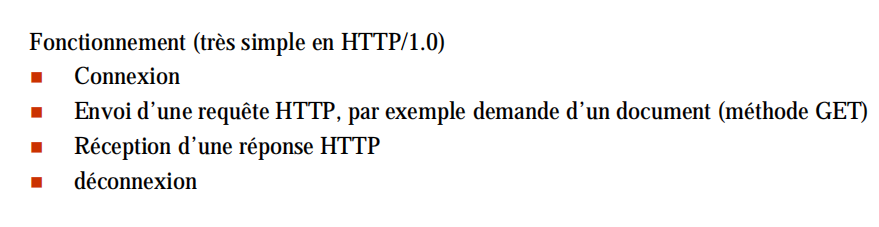
required : indique que la saisie de champ est obligatoire.

pattern : accepte une expression régulier que le champ doit valider, si non un message d’erreur de validation s’affiche.

placeholder : affiche un texte indicatif sur le champ.

autofocus : l’élément reçois le focus dès le chargement de la page.

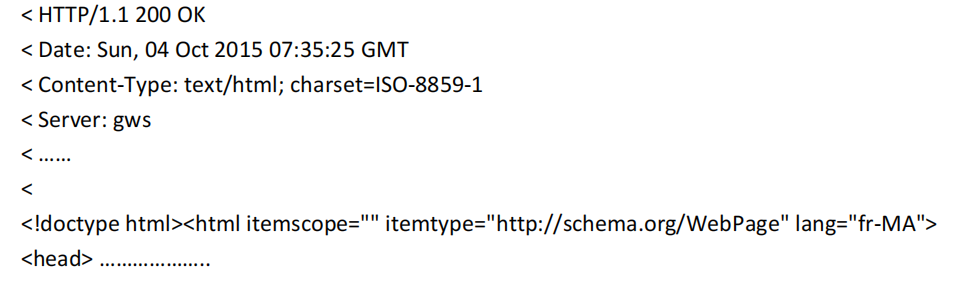
**Le protocole http :** protocole basé sur l’architecure client/serveur, permet á un internaute d’accéder á un contenu situer sur le web en utilisant un navigateur.



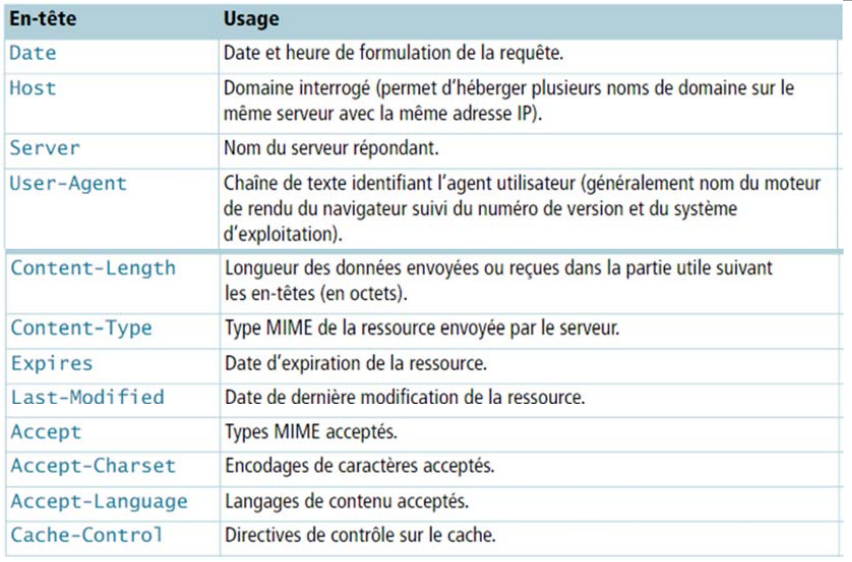
Exemple d’une requête web



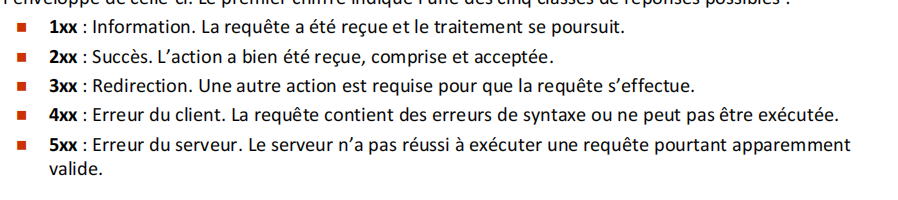
Exemple d’une réponse http :

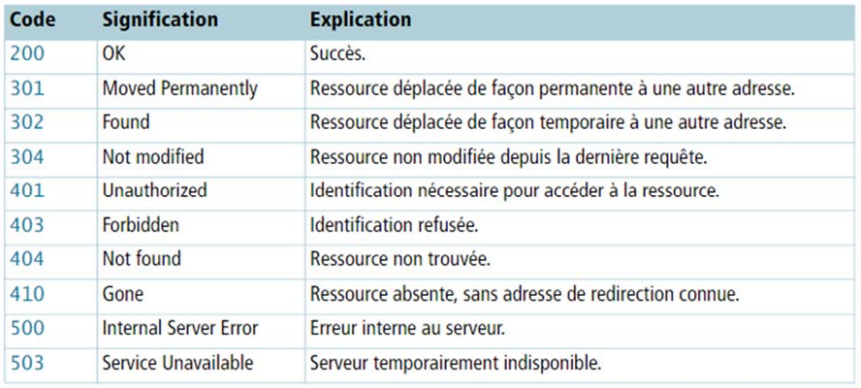


en-tète ou bien header http :



Les code d’état :





CSS

CSS(Cascading Style Sheets) : Langage de mis en forme d’un page html, permet de modifier l’aspect des éléments html.

Les feuilles de styles CSS sont composées d’un sélecteur et un bloc de propriété valeur.

**La séparation du fond et de la forme :** mettre de ce qui est du html dans un fichier, et de ce qui est CSS dans un autre, ce qui permet d’appliquer un style sur plusieurs page html, d’avoir plusieurs styles pour un seul site, limité le téléchargement des styles pour un ensemble de page, ….

Insertion d’un fichier Css ‘a l’entête d’une page html :

<link rel="stylesheet" href="chemin\_ressource/commun.css" />

Ou bien:

<head>

<style type="text/css">

@import url(chemin\_ressource/commun.css);

</style>

</head>

Insertion de CSS ‘a l’entête d’une page html :

<head>

<style type="text/css">

body{

color: navy;

}

</style>

</head>

Insertion de style au sein d’une balise á l’aide de l’attribut style :

<p style = “text-decoration : underline :”> Ceci du texte</p>

Si on a un règle Css vient de contredire une règle précédente, la dernière qui sera appliquer.

Le style appliquer par attribut style á plus de priorité que de ce qui est définit dans un fichier externe, ainsi que ce dernier a plus priorité que de ce qui est déclarer dans la balise <style>.

!important : utiliser pour dire que ce style, va être appliquer même si la propriété a une autre valeur dans un autre sélecteur 🡪 color: black !important;

Les différents sélecteurs CSS :

e1, e2, e3{} 🡪 permet d’applique le style sur tous les éléments e1, e2,e3.

\*{} 🡪 sélecteur universel : appliquer sur tous les éléments html.

.classe1{} 🡪 appliquer sur tous les élément qui contient l’attribut class = ‘’classe1’’ ;

e1.classe1{} 🡪 appliquer sur les élément de e1 qui ont l’attribut class = classe1 ;

Une classe peut être utiliser dans plusieurs éléments html, et un élément html peut appartient á plusieurs classes 🡪 <p class = ‘’classe1 classe2 ‘’>

#id2{} 🡪 appliquer sur l’élément qui a comme attribut id = ‘id2’ ;

2 éléments ne peuvent pas avoir le même id ;

e1[attribut1]{} 🡪 appliquer sur les éléments de type e1 qui ont l’attribut2 quelle que soit sa valeur.

\*[attribut2]{} 🡪 appliquer sur tous les éléments qui ont l’attribut2 quelle que soit sa valeur.

e1[attr1][attr2]{} 🡪 appliquer sur l’élément e1 qui a á la fois les attributs attr1 et attr2.

e1[attr1 = ‘‘val1’’] 🡪 appliquer sur l’élément e1 qui a comme valeur de l’attribut attr1 val1.

e1[attr1 ~= ‘’val1’’] 🡪 appliquer sur l’élément e1 qui contient exactement la chaine val1 de la valeur de l’attribut attr1 ;

e1[attr1 \*= ‘’val1’’] 🡪 appliquer sur l’élément e1 qui contient la chaine val1 de la valeur de l’attribut attr1 🡪 peut être une sous chaine d’une chaine.

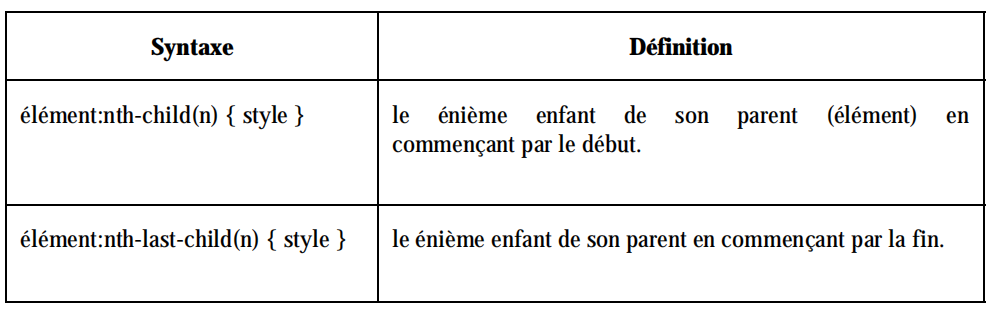
e1[attr1 ^= ‘’val1’’] 🡪 appliquer sur l’élément e1 qui commence exactement avec la chaine val1 de la valeur de l’attribut attr1.

e1[attr1 |= ‘’val1’’] 🡪 appliquer sur l’élément qui commence avec la sous chaine 🡪 peut-être une sous chaine de la première chaine.

e1[attr1 $= ‘’val1’’] 🡪 les élément qui ont la valeur de l’attribut attr1 se termine par la chaine val1.

e1 e2{} 🡪 appliquer le style sur l’élément e2 qui a comme parents ou grand parent l’élément e1.

e1 > e2{} 🡪 appliquer le style sur l’élément e2 qui a comme parent direct l’élément e1 ;



e1 + e2{} 🡪 appliquer le style sur l’élément e2, si est seulement si l’élément précédent direct c’est e1 ;

Les pseudo-classes applicable aux liens:

a :link{} 🡪 appliquer de style sur un lien non encore vue.

a :visted{} 🡪appliquer de style sur un lien déjà visité.

Les pseudo-classe dynamique :

e1:hover{} 🡪 agit lorsque le curseur de la souris pointe sur l’élément e1 ;

e1 :active{} 🡪 agit lors de l’activation de l’élément par un clic de souris ;

e1 :focus{} 🡪 agit lors de la récupération du focus par le clavier ou par la souris.

Il y’a aussi :enabled, :disabled, :checked appliquer sur les input qui veirfier ces attribut ;

Les pseudo-éléments :

e1::first-letter 🡪 agit sur le premier lettre de l’élément e1.

e1 ::first-line 🡪 agit sur le première ligne de l’élément e1 ;

e1 ::before{content :’’<<’’; pro1: val1 …} 🡪 insérer << avant l’élément e1 doté du style précisé.

e1 ::after{content : ‘’>>’’ ; ici du style} 🡪 insérer >> après l’élément e1 doté du style précisé.

Héritage de style :

Si on applique un style sur un div, tous ses élément enfant vont hériter ses propriétés, sauf s’il ont leur propre style.

Il y’a des proprietés qui ne peuvent pas s’hériter, à titre d’exemple, les marges, les bordures, les dimensions, les positions.

Requêtes de médias :

Permettre de crée de styles spécifique adaptés à chaque type de médias.

Une requête media se compose d’un type de média ciblé, des opérateurs logique (AND ONLY NOT), et des conditions.

Pour préciser ses paramètres 3 façon son disponible :

**Requête écrite dans la balise <link> :**

<link href="monstyle.css" **media="type\_de\_média and ( condition )**" rel="stylesheet" /> 🡪 le fichier Css n’est chargé que si l’internaute utilise le media précisé.

Requête utiliser comme un sélecteur de proprietés :

@media type\_de\_média and (condition) {

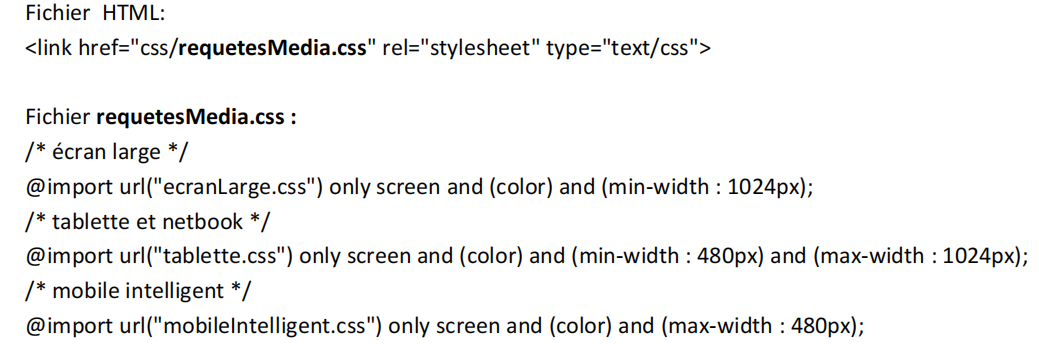
/\* règles CSS \*/

}

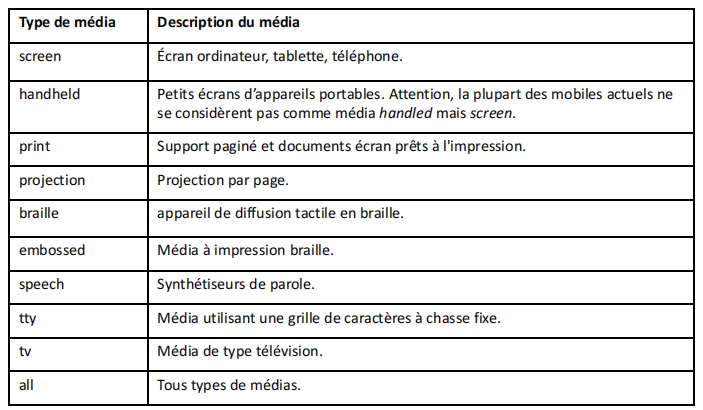
**Requête á l’aide d’import :**

@import url(‘’commun.css’’)type\_de\_media and (condition);

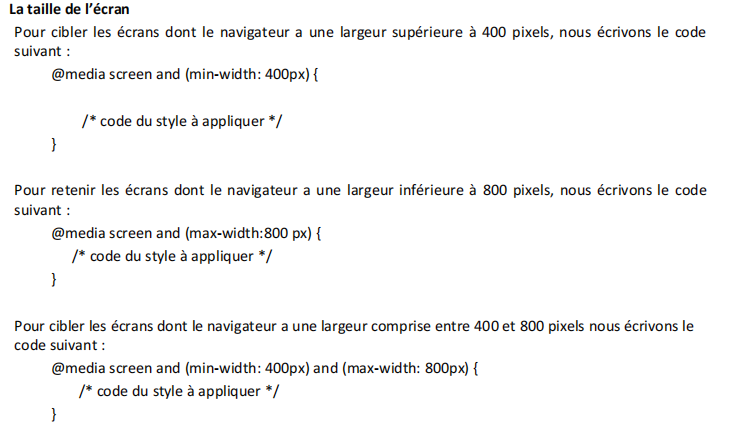
Cette annotation est déclarée dans un fichier css qui est inclut au page html á l’aide de la balise <link>, ce fichier joue le rôle de routeur vers le fichier style convenable.

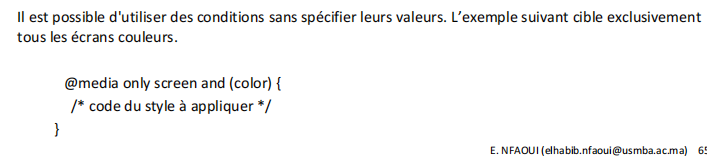


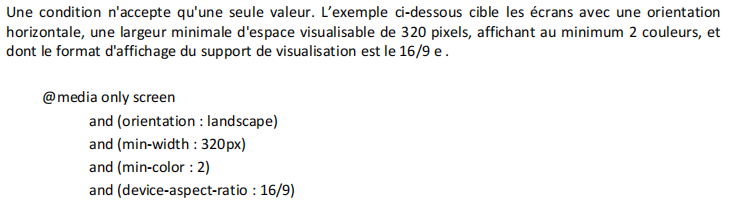
Les différents types de médias :



Exemples des requêtes de médias :

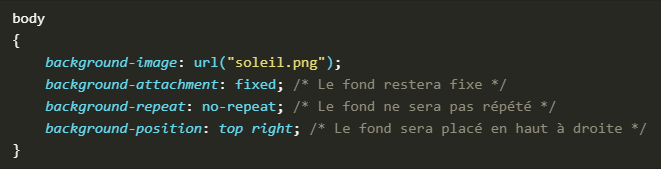






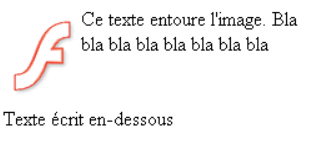
Quelque propriété CSS :

Formatage du texte :

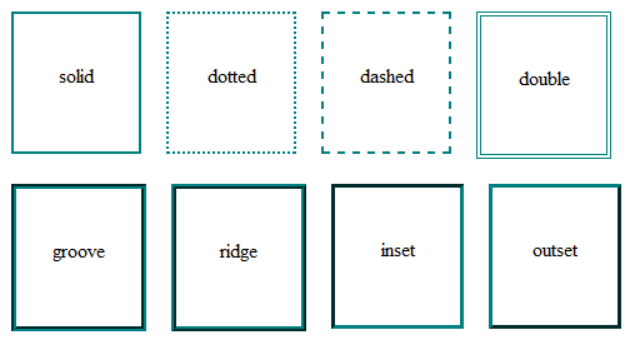
* font-size : 24px ; 🡪 permet de définir la taille du texte.
* font-family : police1, police2 ; 🡪 modifier la police du texte, si police 1 n’existe pas sur la machine cliente le navigateur essaye de charge police2.
* font-style : italic ; 🡪 prend les valeurs : normal, oblique, italic.
* font-weight :bold 🡪 mettre en grand, prend les valeurs : bold, normal.
* text-decoration : underline ; 🡪 applique des soulignement sur le text prend les valeur : line-through, overline, blink.
* text-align : center 🡪 permet d’aligner le texte, prend les valeurs : left, center, right, justify (prend toute la largeur possible, sans laisser d’espace blan á la fin des ligne).
* text-shadow: 2px 2px 4px black; 🡪 ajouter de shadow pour le text.
* color : red ; 🡪 pour colorer un texte.
* background-color : black ;🡪 mettre le fond en un couleur précis.
* background-image : url(‘’nomPic.jpg’’);
* 
* opacity: 0.5; 🡪 pour preciser la transparence de texte.

float : left ; 🡪 permet de faire flotter une image ou bien un block.

clear :both ; 🡪 permet de mettre du texte en dessous du flottant.

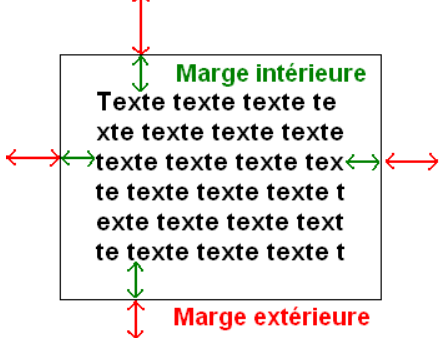


border: 3px blue dashed; 🡪 crée des bordures utiliser souvent dans les tableau.

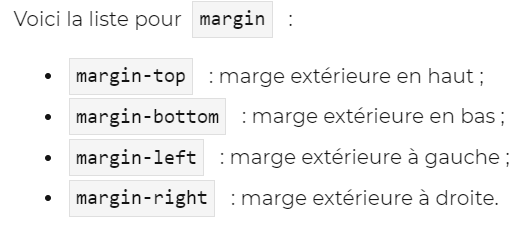


border-radius: 10px, 5px, 10px, 5px; 🡪 permet de crée des bordures arrondies, 1en haut à gauche, 2 : En haut à droite, 3 : En bas à droite, 4 : En bas à gauche.

box-shadow: 6px 6px 0px black; 🡪 permet de crée de l’ombre pour un bordure, 1 le décalage horizontal de l'ombre, 2 le décalage vertical de l'ombre, 3 l'adoucissement du dégradé, 4 La couleur de l'ombre.

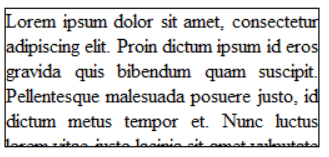


padding c’est la marge intérieure, margin c’est la marge extérieure, les 2 exprimer en px.

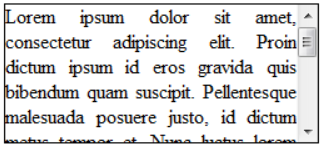


margin : auto ; 🡪permet de center un block donnée.

overflow :hidden 🡪 pour cacher un texte si il dépasse le block ou il a été définit.



overflow :scroll 🡪 permet d’ajouter une barre de défilement pour le block.

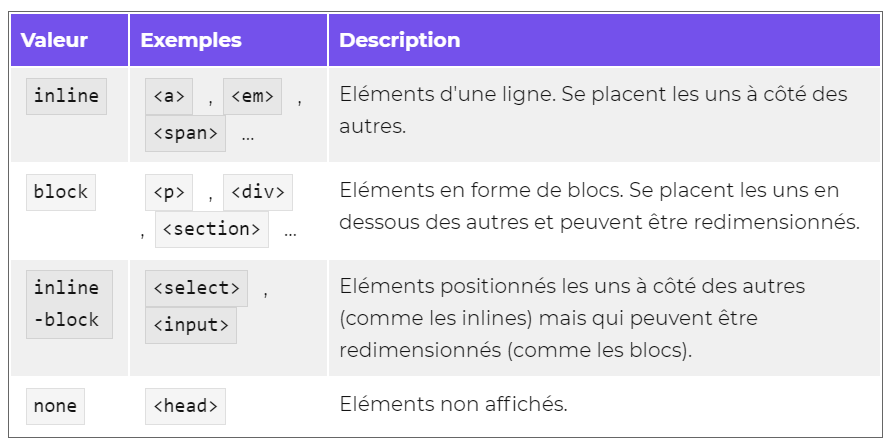


display :flex;

display: flex; flex-direction: column;



display propriétés:



vertical-align : cette propriété permet de modifier l'alignement vertical des éléments. Voici quelques-unes des valeurs possibles pour cette propriété :

* baseline : aligne la base de l'élément avec celle de l'élément parent (par défaut) ;
* top  : aligne en haut ;
* middle  : centre verticalement ;
* bottom  : aligne en bas ;

