

1- formatage texte: HTML et CSS

<!-- -->: commentaire

<p> </p>: paragraphe x

 </br>: Saut à la ligne x

<hr>: Séparation horizontale x

 : Texte à

 : Texte import

 : Texte barré

<ins> </ins>: Texte Surlig

<mark> </mark>: Texte Surlig

 : Texte peu im

<i> </i>: // (italic)

 : Texte barré (Text non correct)

<h1> </h1>: Titre x

<h2> </h2>: Titre x

2 Types de balises:

* balises en paire

(une balise ouvrante

et une fermante)

* balises accolées

(une seule balise)

Attribut: se

situe dans une

balise comme:

src, lang, type

...ect

1- Formattage texte: HTML et CSS

<!-- -->	: Commentaire	<title> ... </title>	X
<p>	</p>	: paragraphe	X
 	</br>	: Saut à la ligne	X
<hr>		: Séparation horizontale	X
		: Texte important (gras)	X
		: Texte important (italique)	X
		: Texte barré (Text à Supp)	X
<ins>	</ins>	: Texte Surligné (insère)	X
<mark>	</mark>	: Texte Surligné	X
		: Texte peu important (gras)	X
<i>	</i>	: // (italique)	X
<s>	</s>	: Texte barré (Text non correct)	X
<h1>	</h1>	: Titre	X
<h2>	</h2>		X

- `<q>` `</q>`: mettre une citation entre " "
- `<cite>` `</cite>`: Titre du document / Film
- `<blockquote>` `</blockquote>`: ~~mettre~~ une citation
- `<details>` `</details>`: cacher un Texte dans une petite
 boîte.
- `<summary>` `</summary>`: modifier le nom de `<details>`
- `` nom du lien `<a>`: lien du site x
- `` nom du lien `<a>` si
 afficher on
 bulle
- `target=" _blank"`: pour avoir le lien dans un autre
- `<abbr>` `</abbr>`: abréviation
- `<address>` `</address>`: adresse
- `` : mettre une Adresse Mail
- `<div>` `</div>`: écrire de droite à gauche
- `` télécharger fichier x ②

- $\langle bdi\ dir="rtl">$ $\langle /bdi>$: right to left
- $\langle bdi\ dir="ltr">$ $\langle /bdi>$: left to right
- $\langle bdi\ dir="">$ $\langle /bdi>$: changer l'orientation Texte.
- $\langle dl>$ $\langle /dl>$: définition
- $\langle dl>$ $\langle /dl>$: pour faire une liste
- $\langle dt>$ $\langle /dt>$: Thème
- $\langle dd>$ $\langle /dd>$: définition
- $\langle progress\ value=""\ max="">$ $\langle /progress>$: barre de progress
- $\langle sub>$ $\langle /sub>$: écrire sous comme C_0^x
- $\langle sup>$ $\langle /sup>$: écrire en haut comme m^x
- $\langle uhh>$ $\langle /uhh>$: couper le mot à un endroit.
- $\langle code>... \langle /code>$: pour écrire un code de langage de programmation ③

- `<kbd> ctrl+s </kbd>`: pour Touche clavier
- `<meter value="Valeur numérique"> ... </meter>`: barre
prog
- `<meter>` prend les attributs :

Value (Valeur)	max (Valeur maximale)
min (Valeur minimale)	high seuil où la valeur devient "Haute"
low (seuil où la valeur devient "basse")	
optimum (seuil où la valeur est optimale)	
- `<output> </output>`: pour résultat
- `<pre> ... </pre>`: génère le Texte comme vous l'avez écrit.
(préformatage)
- `$...$`: Sortie (Résultat d'un calc)
- `$...$`: formatage Texte (Variable Math)
- créer une ancre : `` rajouter l'attribut `id` à une balise qui va servir de repère.

~~~~ x

		} liste à puce (liste non ordonnée)
		
		
		
		} (liste ordonnée)
		

on peut ajouter reversed <ol reversed> il permet d'inverser les numéros (décroissante)

<ol start="nombre"> (nombre que je veux que la liste commence)

<ol type="a"> (numérotation alphabétique)

<ol type="I"> (numérotation romaine)

3. Les Tableaux :

<table>

<caption> titre d'un Tableau </caption>

<thead> entête </thead>

<tbody> corps d'un Tableau </tbody>

<tfoot> pied du Tableau </tfoot>

<tr> (ligne)

<td>

</td> : colonne

<th>

</th> : colonne d'entête

</tr>

</table>

<td rowspan=" " " ">

</td> : 1 case que tu veux quelle prenne les

<td colspan=" " " ">

<td> : colonne première des colonnes lignes

6

4- Formulaire: X

⑦

<form method="post" action="">

.....<input name="" type="text"> (entree du texte)

<label for="">.....<input type="text" name="">

place holder="Texte par défaut"> (dans la balise input)

- type="text" (dans la balise input pour saisir du texte)

- type="password" (dans la balise input pour saisir un mot de passe)

- type="checkbox" (dans la balise input pour cocher une case)

- type="radio" (dans la balise input pour une case obligatoire de la cocher)

- type="color" (dans la balise input pour les couleurs)

- valeur "valeur de la couleur" (sans la balise input pour spécifier la couleur)
- type = "date" (dans la balise input pour le calendrier)
- type = "email" (saisie mail dans la balise input)
- type = "range" (barre glissable)
- type = "number" (pour nombre)
- type = "hidden" (champ caché)
- type = "Submit" (bouton pour envoyer)
- en lui ~~donne~~ donne un nom value = "..."
- type = "reset" (bouton pour sup)
- <fieldset form="id du form"> pour faire des champs à l'intérieur
- <legend> </legend> pour faire une légende - legend ②

- `<textarea rows="nombre de lignes" cols="nombre de colonnes">...</textarea>`

- placeholder = "texte que tu veux qu'il s'affiche avant la saisie dans le champ" (dans la balise `textarea`)

- `<select name=" " >`

`<option>`

`</option>`

Selectionner

`</select>`

lundi
11.00
...

`<optgroup label="nom de notre sélection">` c'est tout

`<option>...</option>`

`</optgroup>`

de label select
on fait des sous groupes
avec des options

sciences
11.00
FTN

9

- multiple : pour sélectionner plusieurs champs dans la balise select
- pour désactiver des champs <disabled> nom du champ <disabled>.

<button> ... </button>

5- Images :

- (on n'utilise si l'image est dans un même fichier que notre page HTML)
- width=" " " (largeur de l'image dans la balise img)
- height=" " " (hauteur de l'image dans " " conseil)
- alt : donner une description (conseil de le faire en tout cas)
- (on l'utilise quand l'image est dans un dossier) *

(10)

- `<figure>`
 ``
 `<figcaption> ... </figcaption>`
`</figure>`

Pour faire
 une légende
 à notre
 image

insérer une map = "# nom de votre map" dans la balise
 img en créant :

(compléter)
 {
 `<map nam = "nom de la map">`
 `<area href "contenu de l'img que vous voulez zoom"`
 `shape = "rect" coords "0,0 250,250">`
 `</map>`
 }

6- Sons et Vidéos :

Audio : mp3 (Compatible partout)
 Wave (Compatible sauf IE)
 ogg (Compatible sauf IE et Safari)

11

video : mp4 (compatible partout)

webm (compatible sauf IE et Safari)

ogg ("")

- <audio src=" " > (même chose que image ~~source~~...)

- controls=" " (barre de contrôle de l'audio dans la balise audio)

- autoplay (~~lancer~~ autoplay pour l'audio dès que la page est chargée).

- muted (muté ~~le son~~ l'audio)

- loop (lire l'audio en boucle)

dans la balise
Audio

- <video src=" " > (~~même~~ même chose que image et audio pour le dossier)

- (même attributs que l'audio controls...)
générer largeur et hauteur comme l'image.

- poster="URL image" (dans la balise video miniature) (12)

7. Structurer la page d

`<div> ... </div>` : b

` ... ` : x

`<main> ... </main>` : 0

`<section> ... </section>`

`<article> ... </article>`

`<header> ... </header>`

`<footer> ... </footer>`

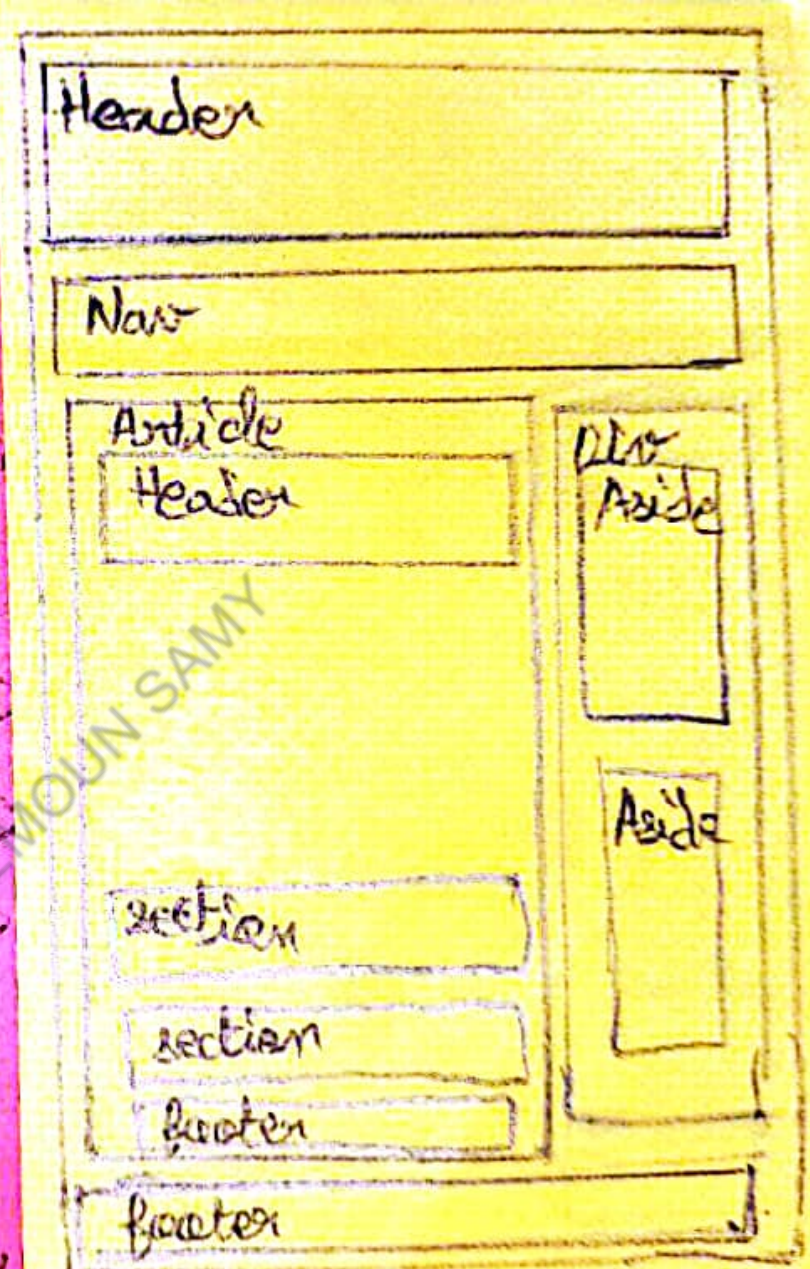
`<nav> ... </nav>` :

liste ``

` ... `

``

`<aside> ... </aside>` : Boite apart.



menus

- `<div>` ... `</div>` : bloc de contenu
- `` `` : * balise que l'on place au sein d'un paragraphe
- `<main>` `</main>` : contenu principale de la page
- `<section>` `</section>` : lui mettre un titre & définir une section
- `<article>` `</article>` : (même chose que section)
- `<header>` `</header>` : entête de la page
- `<footer>` `</footer>` : pied de la page
- `<nav>` `</nav>` :
 - liste ``
 - `` ... ``
 - ``
 } Pour faire des menus
- `<aside>` ... `</aside>` : Boite apart.

8 - introduction design

- lier son fichier HTML à un fichier CSS:
Dans HTML:

`<link rel="style sheet" href="nom du fichier CSS">`
Syntaxe CSS:

```
{  
  <propriete>: <valeur>;  
  ...  
}
```

exemple :

body

```
{ background-color: #DDDD;  
  margin: auto;  
  width: 1000px;  
}
```



- ## 9 - Styliser le Texte : (CSS)
- font-style : ~~mettre~~ changer la police du Texte. X
 - font-variant : styliser les éléments (majuscule ...)
 - font-weight : mettre le Texte en (gras, normal, ~~light~~ ^{extra})
 - font-size : Taille de la Police (avec px, em, %, ~~log~~ ^{rem})
 - line-height : ~~espace~~ entre les lignes (em, ...)
 - font-family : changer la Police des caractères (Arial, ...)
 - font : elle prend le style, variant, ^{weight, sans-serif}, weight, size, height et family.
 - text-decoration : line : ~~la~~ ligne du Texte concerne ^{per} ligne
 - text-decoration-color : couleur du Texte
 - text-decoration-style :
 - text-decoration : commence par line, color, style
 - text-shadow : ombre du Texte (0 0 3px, couleur) (15)

- text-transform: Transformer le texte (Majuscule est une
pas possible que
font-variant.
- text-align: alignement de Texte X
- vertical-align: alignement Vertical.
- text-indent: indentation du Texte (cm, px...)
- text-overflow:
- letter-spacing: espace entre les lettres (px).
- white-space: les espaces.
- word-spacing:
- word-break: coupe des mots.
- direction: direction du Texte.
- text-justify: justifier un Texte (pas conseillé)
- text-align-last: alignement de la dernière ligne.
- /*
- */ : commentaire

10. couleur et fond: (CSS)

color: pour changer la couleur. (nom de la couleur, H..)

- rgb(, ,): red green blue

- rgba(, , ,): a pour ajouter l'opacité de la couleur.

- hsl(, ,): ^{luminosité} Niveau de Saturation.
Niveau de la Teinte

Background-color: le fond

• Background-color: mettre une couleur au fond

• Background-image: url (lien vers l'image): mettre une image au fond.

• Background-position: permet de positionner le fond (Top, ...)

• Background-repeat: pour la répétition du fond (no repeat, repeat-x, repeat-y, space, ...)

(17)

- Background-size: Taille du fond (cover, contain,
- Background-attachment: le comportement du fond (scroll, fixed)

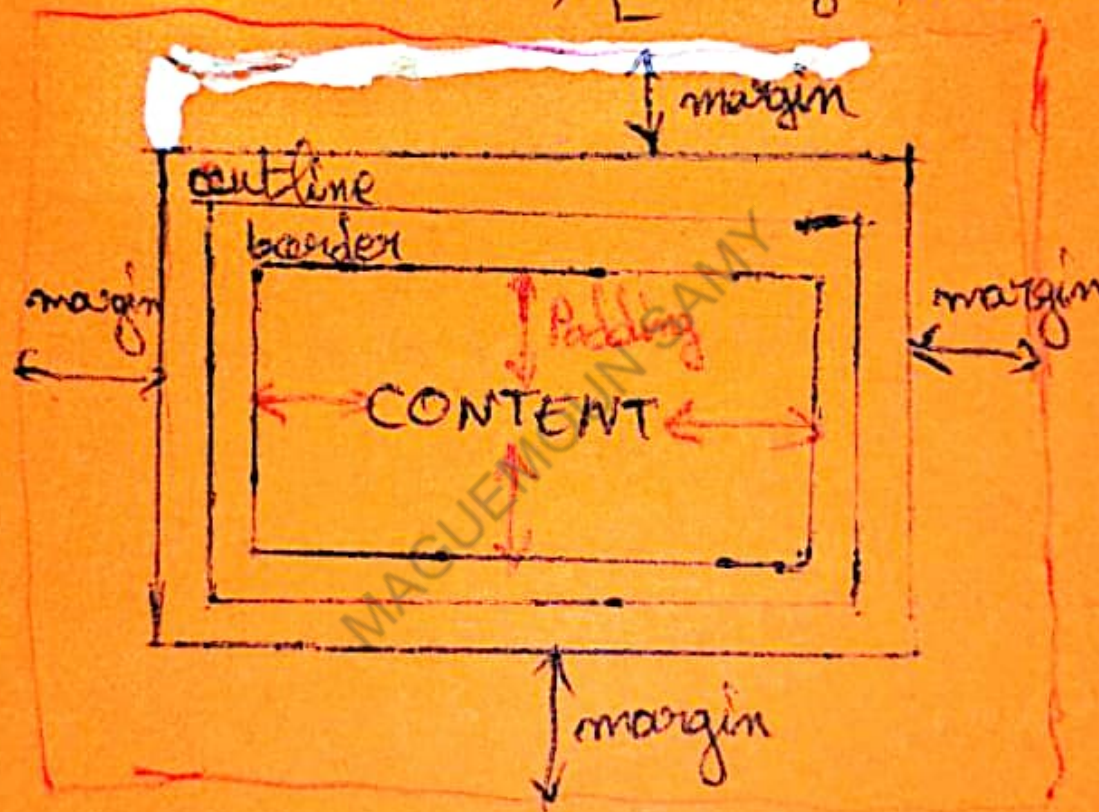
la propriété Background prend tout position...

1d - styliser liste: (css)

- list-style-image: url([lien vers l'img](#)): image appliqué aux listes.
- list-style-position: changer la position de la puce (outside, inside...)
- list-style-type: changer le style de la puce (square, none, disc, circle, armenian, georgian, lower-roman, latin)

12. Modèle de boîte :

- width: largeur (en px, %) [min-width, max-width]
- height: Hauteur (en px, %) [min-height, max-height]



- padding: marge intérieure (padding-top, padding-bottom, padding-left, padding-right, c'est en px)

- margin: marge extérieure (même propriété que padding).
- overflow: comportement en cas de débordement (auto, scroll, hidden).

13 - Bordures et Contours:

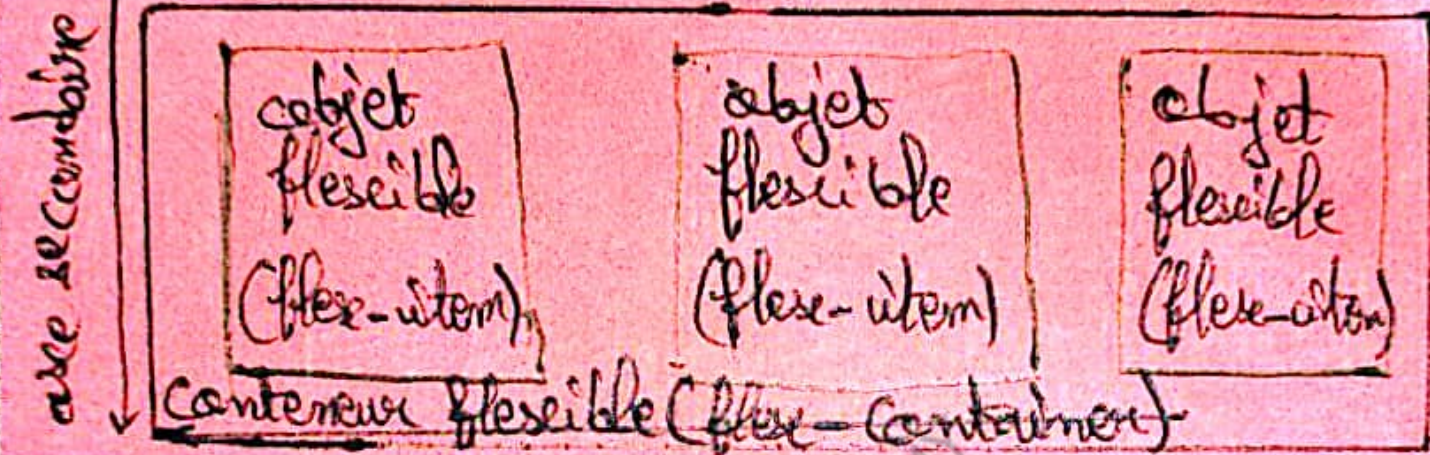
- * Border: pour faire des bordures
- border-width: épaisseur (en px)
- border-style: dotted, dashed, solid, double, groove, ...
- border-color: ~~if~~ mettre une couleur pour la bordure
on peut aussi faire border-top-..., border-left-...
- border-image: ~~se~~ utiliser le site border-image.com
- border-radius: bordure arrondie.
- outline: (même propriété que Border): Contour de la bordure.

14 Affichage et positionnement :

- `display: none;` : masquer l'information.
- `display: inline;` : rendre un élément *inline* (sur la ligne)
- `display: block;` : rendre un élément *block*.
- `display: inline-block;` : les 2 en même temps.
- `display: list-item;` : avoir le comportement d'une liste à puces.
- `visibility:` pour la visibilité des éléments.
- `opacity:` opacité.
- `position:` pour modifier la position (on peut modifier Top, bottom, ...)

14 / Flex box

→ Axe principal



- display: flex; travailler avec des boîtes flexibles.
- flex-direction: row / column / row-reverse / column-reverse; ~~orientation~~ de nos blocs.
- flex-wrap: nowrap / wrap / wrap-reverse; si on a trop de blocs pour qu'ils forment un retour à la ligne.
- flex-flow: valeur direction valeur wrap;
- justify-content: flex-start / flex-end / center / space-between; va nous permettre de répartir nos éléments
- space-around / space-between;

- sur l'axe horizontal (axe principal).
- align-items: flex-start / flex-end / center / stretch / baseline; il va nous permettre de répartir nos éléments sur l'axe vertical (axe secondaire)
 - order: entier; pour faire l'ordre d'objet.
 - flex-grow: entier; pour faire des unités de dimension.
 - flex-shrink: entier; pour faire ~~moins~~ le min de dim.
 - flex-basis: length;
 - flex: grow shrink basis;

* Créer des apparences dynamiques :

* Au survol : ... : hover

* Au clic : ... : active (au moment du clic)

* à la sélection : ... : focus (lorsque l'élément est sélectionné)
(Fonctionne Tab pour déplacer le focus)

* Sélection avancée : ... : hover + ...

* sélecteur : ... : visited (pour un lien)

* mise en page avec flex-box :

- display : flex : aligner les éléments d'un conteneur.

- flex-direction : donner une direction aux

• row : organiser sur une ligne

• column : organiser sur une colonne.

• row-reverse : organiser sur une ligne, mais en ordre inverse.

• column-reverse : organiser sur une colonne, mais en ordre inverse.

- flex-wrap : retourner à la ligne.

• nowrap : pas de retour à la ligne

• wrap : les éléments vont à la ligne lorsqu'il n'y a plus de place.

* Suite mise en page avec flex-box :

• wrap-reverse : les éléments vont à la ligne lorsqu'il n'y a plus de place, en sens inverse.

- justify-content : Aligner les éléments sur l'axe principal.

• flex-start : alignés au début.

• flex-end : alignés à la fin.

• center : alignés au centre.

• space-between : les éléments sont espacés sur tout l'axe (il y a de l'espace entre eux)

• space-around : idem, les éléments sont espacés sur tout l'axe mais ils laissent aussi de l'espace sur les extrémités.

• Align-items : Aligner les éléments sur l'axe secondaire.

• stretch : les éléments sont étirés sur tout l'axe.

* Suite mise en page avec flex-box :

• flex-start : alignés au début.

• flex-end : alignés à la fin.

• center : alignés au centre.

• baseline : alignés sur la ligne de base (semblable à flex-start).

- align-content : répartir les blocs sur plusieurs lignes.

• stretch : les éléments s'étirent pour occuper tout l'espace.

• flex-start : les éléments sont placés au début.

• flex-end : " " " " à la fin.

• center : les éléments sont placés au centre.

• space-between : les éléments sont séparés avec de l'espace entre eux, mais il y a aussi de l'espace (3)

• space-around : idem, mais il y a aussi de l'espace (3)

* les bases de CSS Grids :

- display : grid : Définir une grid avec CSS

• grid-template-columns : pour le nombre de colonnes et la largeur de chacune d'entre elles.

• grid-template-rows : pour le nombre de lignes et la hauteur de chacune d'entre elles.

- grid : gap : avec notre contenu.

fr : fraction unités.

- grid : définir la taille des éléments.

• grid-column-start : indique la ligne verticale de départ de l'élément.

• grid-column-end : indique la ligne verticale d'arrivée de l'élément. (4)

• grid-row-start : indique la ligne horizontale de départ de l'élément.

• grid-row-end : indique la ligne horizontale d'arrivée de l'élément.

* Absence d'autres techniques de mise en page :

• display : permet de transformer vos éléments.

• display : none : cache vos éléments.

• display : inline-block : rend vos éléments inline tout en permettant de les dimensionner.

• position : positionnez vos éléments.

• position : relative : permet de décaler l'élément par rapport à sa position normale (left, top, bottom, right).

• position : absolute : permet de déplacer l'élément n'importe où sur la page. (5)

• position : fixed : rendre un élément fixe.

• position : sticky : rendre un élément adhérent.

* Ajout des Tableaux :

<table> </table> : indique le début et la fin du tableau.

<tr> </tr> : indique le début et la fin d'une ligne du tableau.

<td> </td> : indique le début et la fin du contenu d'une cellule.

- ajouter la propriété CSS : border : 1px solid à td. et ajouter border-collapse : collapse à table pour avoir des bordures collées.

• <th> </th> : l'entête d'un tableau, se trouve

• <caption> </caption> : titre d'un tableau (6)

* Créer des apparences dynamiques :

* Au survol : ... : hover

* Au clic : ... : active (au moment du clic)

* à la sélection : ... : focus (lorsque l'élément est sélectionné)
(Fonctionne Tab pour déplacer le focus)

* Sélection avancée : ... : hover + ...

* sélecteur : ... : visited (pour un lien)

* mise en page avec flex-box :

- display : flex : aligner les éléments d'un conteneur.

- flex-direction : donner une direction aux

• row : organisé sur une ligne

• column : organisé sur une colonne.

• row-reverse : organisé sur une ligne, mais en ordre inverse.

• column-reverse : organisé sur une colonne, mais en ordre inverse.

- flex-wrap : retourner à la ligne.

• nowrap : pas de retour à la ligne

• wrap : les éléments vont à la ligne lorsqu'il n'y a plus de place.

* Suite mise en page avec flex-box :

• wrap-reverse : les éléments vont à la ligne lorsqu'il n'y a plus de place, en sens inverse.

- justify-content : Aligner les éléments sur l'axe principal.

• flex-start : alignés au début.

• flex-end : alignés à la fin.

• center : alignés au centre.

• space-between : les éléments sont espacés sur tout l'axe (il y a de l'espace entre eux)

• space-around : idem, les éléments sont espacés sur tout l'axe mais ils laissent aussi de l'espace sur les extrémités.

• Align-items : Aligner les éléments sur l'axe secondaire.

• stretch : les éléments sont étirés sur tout l'axe.

* Suite mise en page avec flex-box :

• flex-start : alignés au début.

• flex-end : alignés à la fin.

• center : alignés au centre.

• baseline : alignés sur la ligne de base (semblable à flex-start).

- align-content : répartir les blocs sur plusieurs lignes.

• stretch : les éléments s'étirent pour occuper tout l'espace.

• flex-start : les éléments sont placés au début.

• flex-end : " " " " à la fin.

• center : les éléments sont placés au centre.

• space-between : les éléments sont séparés avec de l'espace entre eux, mais il y a aussi de l'espace (3)

• space-around : idem, mais il y a aussi de l'espace (3)

* les bases de CSS Grids :

- display : grid : Définir une grid avec CSS

• grid-template-columns : pour le nombre de colonnes et la largeur de chacune d'entre elles.

• grid-template-rows : pour le nombre de lignes et la hauteur de chacune d'entre elles.

- grid-gap : avec notre contenu.

fr : fraction unités.

- grid : définir la taille des éléments.

• grid-column-start : indique la ligne verticale de départ de l'élément.

• grid-column-end : indique la ligne verticale d'arrivée de l'élément. (4)

• grid-row-start : indique la ligne horizontale de départ de l'élément.

• grid-row-end : indique la ligne horizontale d'arrivée de l'élément.

* Absence d'autres techniques de mise en page :

• display : permet de transformer vos éléments.

• display : none : cache vos éléments.

• display : inline-block : rend vos éléments inline tout en permettant de les dimensionner.

• position : positionnez vos éléments.

• position : relative : permet de décaler l'élément par rapport à sa position normale (left, top, bottom, right).

• position : absolute : permet de déplacer l'élément n'importe où sur la page. (5)

• position : fixed : rendre un élément fixe.

• position : sticky : rendre un élément adhérent.

* Ajout des Tableaux :

<table> </table> : indique le début et la fin du tableau.

<tr> </tr> : indique le début et la fin d'une ligne du tableau.

<td> </td> : indique le début et la fin du contenu d'une cellule.

- ajouter la propriété CSS : border : 1px solid à td. et ajouter border-collapse : collapse à table pour avoir des bordures collées.

• <th> </th> : l'entête d'un tableau, se trouve

• <caption> </caption> : titre d'un tableau (6)

Selecteurs CSS!

Notation	Signification
*	Sélection Universel, Sélectionne toutes les lignes sans exception.
.val	les éléments dont l'attribut class="val".
#id	les éléments dont l'id est id.
E	un élément de type E (l'instance E)
E, F	Sélectionne la ligne E et F
E F	la ligne F descendant de la ligne E.
E + F	la ligne F immédiatement précédé par la ligne E.
E [att]	la ligne E qui contiennent l'attribut att.
E [att="val"]	la ligne E qui contiennent l'attribut att à la valeur val
E.val	sélection de tous les éléments ayant la class "val" qui sont descendants d'un élément E
E #id	la ligne E dont l'id est id.
.class E	sélection de tous les éléments E qui sont des de tous les éléments ayant la classe "class"
E > .class	sélection de tous les éléments ayant la classe "class" qui sont des enfants directs des éléments E
E ~ F	

De Unites CSI

Des Unités CSS:
Unités Absolues: Des Unités de Mesure fixes qui ne dépendent pas d'autres éléments, elles sont généralement utilisées pour des dimensions précises.
Unités Relatives: Des Unités de Mesure qui dépendent d'autres éléments, elles permettent d'obtenir des dimensions plus flexibles et adaptable. em , rem

$$1 \text{ km} = 9,54 \text{ km} = 95,4 \text{ m} = 9$$