Plik : site pour transformer les pages HTML en liens

Deux type d’environnements :

Environnement de développement (ce qu’on utilise : pc, logiciel)

Environnement production (Linux, windows, mac, pour tester les langages)

Le CSS : C’est l’aspect visuel d’une page, l’esthétique … (Fichier a part du HTML)

Quelque terme :

= margin l’extérieur de la page ( Margin on peut l’integrer sous 4 valeurs : px )

= Padding c’est les contours

=background c’est le fond (couleurs d’arrière-plan)

Color = couleur (# 25 36 96 = code couleurs )

**Exemple :**

P{

Margin : 20px ;

Padding : 10 px ;

Background : red ;

Color : white ; <<< couleur du texte

}

<link rel= "stylesheet" href=’’notre page’’> C’est la balise a mettre pour relié le CSS au HTML ( à mettre sur le fichier HTML )

Pour modifier un seul paragraphe ou un seul truc dans notre html on le modifie dans la balise

On renomme un des paragraphe (balise p) <p id=’’p3’’> et dans le fichier CSS on créer cette règle suivante :

#p3 {

background: green;

}

### Pour un id on met des # et pour appliquer à plusieurs valeurs on met l’attribue class=

Exemple : <p class="pbleu"> (toutes les balises avec le ‘’pbleu’’ aurons la même modifications)

Dans le fichier CSS on met la formule :

.pbleu {

background: silver;

}

Pour les bordure :

Px : épaisseur de la bordure

Exemple :

{border : 1px solid gray ;

}

Width : largeur

Heisht : hauteur

Padding : la décale selon la bordure

Margin : la décale selon le coté de l’écran

* Pour insérer une image ou un texte avec une seule attribue on favorise l’id mais si c’est plusieurs, on les regroupe avec class= et dans le CSS on ne fait qu’une requête
* Pour faire des personnalisations communes, on fait :

Exemple : .p1,.p2,.p3{ .p1 { background : red ;}

Margin : \*\*\*\*{ .p2 { background : blue ;}

Pedding :\*\*\*\* { .p3 { background : yellow ;}

Color : \*\*\*\*{

(limité les lignes de code)

Faire une balise commune et une balise seule, on peut comme dans l’exemple en haut

Genre avoir un background propre à chacun et des bordures, couleurs et marges en communs.

* Il y a aussi le groupe de composant avec les container et items

Container c’est ce qui regroupe un bloc d’image par exemple et les items c’est les images une par une.

Pour les couleurs :

Le fonctionnement :

Valeurs numérique >>>>

RGB (255,99,71) RGB = red green blue ( Le max c’est 255 pour la première )

Ou RGBA : Le A c’est pour régler l’opacité, ça se règle entre 0 et 1

Ou Nomination Exa >>>>>> #FF6347 : De O à 9 et de A à F

Pour les arrière-plan on modifie le background de la sorte :

Exemple :

body {

background-image: url("gradient\_bg.png");}

On peut aussi contrôler l’endroit (inutile) avec background position.

background-attachment: fixed; == pour que l’image suit quand on descend

background-attachment: scroll; == pour que l’image descende en même temps que l’on scroll la page

Pour mettre une photo ou un truc sur la bordure avec une marge négative avec la position

Ou avec la position relative

Exemple : position : relative

Top : -20px

https://www.w3schools.com/css/tryit.asp?filename=trycss\_position\_relative

(Pour cet exemple on ne met pas bottom : 30px comme on pourrait le croire, mais top :30 px.)

Parce que on sépare de 30px par rapport au top)

Et de droite a gauche on met left ou right. Et le tableau sera décalé depuis ce coté

Exemple : left : 30px ; il sera décalé de 30px à partir de gauche

Position absolute : c’est le positionnement à partir de la fenêtre

<https://fr.makingcss.com/transition> : diffèrent préfixe pour générer la transition CSS

-webkit-transition: all 0.5s 0s ease;

-moz-transition: all 0.5s 0s ease;

-o-transition: all 0.5s 0s ease;

transition: all 0.5s 0s ease;

webkit : chrome

moz : firefox

o : opéra

z-index: -1; = pour que l’image soit derrière le texte et si on enlève cette ligne c’est le texte qui sera derrière .

<https://www.w3schools.com/css/tryit.asp?filename=trycss_zindex>

(Formule = z-index = la priorité)

CSS float

img {

float: left ; } C’est pour décalé l’image à gauche

<https://www.w3schools.com/css/tryit.asp?filename=trycss_layout_float>

Pour arrêter l’affichage flottant on créer une div avec une class=’’clear’’ et dans le CSS

.class {clear : both ;}

Pour que les images arrêtent de se mettre l’une à côté de l’autre par exemple.

Maintenant on utilise Flexbox

Si on veut afficher un truc qui n’a pas de style par default, il faut utilisé span

<span class="a">Aliquam</span>

On mettra dans le CSS span.a {

display: inline; pour qui prenne la largeur d’un mot

display: inline-block; qu’il prenne une taille un peu plus grosse

display: block; il prendra la forme d’un gros block superposé

Exemple sur le lien : https://www.w3schools.com/css/tryit.asp?filename=trycss\_inline-block\_span1

Pour centrer une image, définissez les marges gauche et droite sur auto et en faire un block élément :

Exemple :

img {

display: block;

margin-left: auto;

margin-right: auto;

width: 40%;

}

<https://www.w3schools.com/css/css_align.asp>

Pour les sélecteurs :

Avec les parents …

Il y a différents types de sélecteur

Avec un espace, (on a déjà vu)

https://www.w3schools.com/css/tryit.asp?filename=trycss\_sel\_element\_gt

Quand on met le > pour désigner l’enfant, il applique sur l’enfant direct

Alors que si on met :

Div p {

<div>

<p>Paragraph 1 in the div.</p>

<p>Paragraph 2 in the div.</p>

<section><p>Paragraph 3 in the div.</p></section> <!-- not Child but Descendant -->

<p>Paragraph 4 in the div.</p>

</div>

Il prendra toute la descendance de div, même si c’est l’enfant de section (dans l’exemple).

Si on veut rajouter le <p> de section tout en gardant la même règle on rajoute une virgule

Le sélecteur avec un + : On va cibler les p à côté d’un div (dans le sens descendant)

(le voisin)

Exemple :

Div + p {

background-color: yellow;

}

<div>

<p>Paragraph 1 in the div.</p>

<p>Paragraph 2 in the div.</p>

</div>

<p>Paragraph 3. After a div.</p>

<p>Paragraph 4. After a div.</p>

<div>

<p>Paragraph 5 in the div.</p>

<p>Paragraph 6 in the div.</p>

</div>

<p>Paragraph 7. After a div.</p>

<p>Paragraph 8. After a div.</p>

C’est le paragraphe 3 et 7 qui seront en jaune

https://www.w3schools.com/css/tryit.asp?filename=trycss\_sel\_element\_pluss

Sélecteur général : (~) (on appel ca tilde) = Ca cible tous les éléments à coté peu importe leurs place (toujours dans le sens descendant)

C’est pour ça que dans l’exemple ci-dessous le paragraphe 1 n’est pas sélectionné.

div ~ p {

background-color: yellow;

}

<h2>General Sibling Selector</h2>

<p>The general sibling selector (~) selects all elements that are siblings of a specified element.</p>

<p>Paragraph 1.</p>

<div>

<p>Paragraph 2.</p>

</div>

<p>Paragraph 3.</p>

<code>Some code.</code>

<p>Paragraph 4.</p>

Le paragraphe 3 et 4 sont en jaune.

Une autre regle :

a[target=\_blank] {

background-color: yellow;

}

Ça veut dire que tout dans la balise a qui a l’attribue Target qui est égale a blank qui est en jaune

Exemple :

<h2>CSS [attribute="value"] Selector</h2>

<p>The link with target="\_blank" gets a yellow background:</p>

<a href="https://www.w3schools.com">w3schools.com</a>

<a href="http://www.disney.com" target="\_blank">disney.com</a>

<a href="http://www.wikipedia.org" target="\_top">wikipedia.org</a>

Psoeudo element CSS :

First line : première ligne (exemple : p::first-line)

first-letter : première lettre (exemple : p::first-letter {)

before : C’est de dire juste avant tel chose il y a tel chose

exemple :

h1::before {

content: url(smiley.gif);

}

Avant le h1 on met une icône smiley.

after : après tel ou tel chose

Pour le reste des pseudo éléments, tout est dans ce lien :

<https://www.w3schools.com/css/css_pseudo_elements.asp>

Pour les Unités :

Longueurs absolues :

cm centimeters

mm millimeters

in inches (1in = 96px = 2.54cm)

px \* pixels (1px = 1/96th of 1in)

pt points (1pt = 1/72 of 1in)

pc picas (1pc = 12 pt)

Longueurs relatives :

Unit Description

em Relative to the font-size of the element (2em means 2 times the size of the current font)

ex Relative to the x-height of the current font (rarely used)

ch Relative to width of the "0" (zero)

rem Relative to font-size of the root element

vw Relative to 1% of the width of the viewport\*

vh Relative to 1% of the height of the viewport\*

vmin Relative to 1% of viewport's\* smaller dimension

vmax Relative to 1% of viewport's\* larger dimension

% Relative to the parent element

Em : c’est selon la taille de base .

Exemple :

p {

font-size: 16px;

line-height: 2em;

}

2\* 16px c’est-à-dire la ligne fait 32 px

Rem : C’est par rapport à l’élément HTML

html {

font-size:16px;

}

On applique le changement à la balise html

div {

font-size: 3rem;

border: 1px solid black;

}

#top-div {

font-size: 2rem;

border: 1px solid red;

}

Et selon le html on change en rem

<p>The font-size of this document is 16px.</p>

<div id="top-div">

The font-size of this div element is 2rem, which translates to 2 x the browser's font size.

<div>The font-size of this div element is 3rem. It also shows that it does not inherit from its parent element font size.</div>

</div>

<p>The rem unit sets the font-size relative to the browsers base font-size, and will not inherit from its parents.</p>

Le texte dans la div sera plus grand et dans l’id top div il sera un peu plus petit mais toujours grand

Le lien de l’exemple :

<https://www.w3schools.com/cssref/tryit.asp?filename=trycss_unit_rem>

(ON DOIT TOUJOURS METTRE HTML ET PAS BODY)

Vw : viewwith : La largeur de l’écran ou fenêtre. S’adapte à l’écran

Exemple :

h1 {

font-size: 20vw;

}

Le résultat : https://www.w3schools.com/cssref/tryit.asp?filename=trycss\_unit\_vw

V min : c’est la taille maximum :

<https://www.w3schools.com/cssref/tryit.asp?filename=trycss_unit_vmin>

V max ; c’est la taille minimum :

<https://www.w3schools.com/cssref/tryit.asp?filename=trycss_unit_vmax>

CSS icons :

Icone qu’on intègre sur notre page :

On peut aussi les relier à Boots rap ou Google

On voit des petites images mais ils sont codés comme des lettres

Format SVG

Deux bibliothéque les plus connu : fontawesome et glyphicon et material-icons

@media : Utiliser différent équipement (pc, tablette , mobile..)

Only screen : seulement les écrans

Avec la résolution ou largeur de

min-width: 600px

et a partir de :

min-width: 768px

il change un peut la configuration de la page

L’exemple :

@media only screen and (min-width: 600px) {

/\* For tablets: \*/

.col-s-1 {width: 8.33%;}

.col-s-2 {width: 16.66%;}

.col-s-3 {width: 25%;}

.col-s-4 {width: 33.33%;}

.col-s-5 {width: 41.66%;}

.col-s-6 {width: 50%;}

.col-s-7 {width: 58.33%;}

.col-s-8 {width: 66.66%;}

.col-s-9 {width: 75%;}

.col-s-10 {width: 83.33%;}

.col-s-11 {width: 91.66%;}

.col-s-12 {width: 100%;}

}

@media only screen and (min-width: 768px) {

/\* For desktop: \*/

.col-1 {width: 8.33%;}

.col-2 {width: 16.66%;}

.col-3 {width: 25%;}

.col-4 {width: 33.33%;}

.col-5 {width: 41.66%;}

.col-6 {width: 50%;}

.col-7 {width: 58.33%;}

.col-8 {width: 66.66%;}

.col-9 {width: 75%;}

.col-10 {width: 83.33%;}

.col-11 {width: 91.66%;}

.col-12 {width: 100%;}

}

Portrait ou longscap : pour savoir si on tourne l’écran ou pas

@media (min with :600 px) and (max with :750 px)

Largeur min de 600 px et max 750 px. Donc tout ce qui est dans l’intervalle des deux valeurs.

(Différente taille de l’écran du téléphone ou tablette)

Clique droit/inspecter/responsive et tu choisis ton téléphone

Pour la résolution on la met a la fin du fichier CSS

Exemple :

320 >>> 420 (un certains ajustement) >>>> Tel

421 >>> 720 (un certains ajustement) >>>>> Tel

721 >>>> 1080 (un certains ajustement) >>>> Tablette

1080 >>>> peu importe (un certain ajustement>>> Pc

Toujours commencer par les plus petites mesures

FLEXBOX :::: IMPORTANT

(((((( Containers )))))

Flex-direction : (pour savoir si les photos se mettent de bas en haut ou droite à gauche

row = droite à gauche

row-reverse = gauche à droite

Column = de haut en bas

Column-reverse = bas en haut

Flex start : s’alligne a gauche

Flex end : s’alligne a droite

Center : centrer

(alignement horizontal)

(Sa correspond à comment le parent va définir L’espace entre les enfants)

Space between : centrer en utilisant toute la largeur de l’image

Space around : entre center et space between (il va mettre de l’espace un peu partout, bien répartie)

Exempe :

; 50 ! image ! 50 ; 50 ! image ! 50 ; 50 ! image ! 50 ;

Space evenly : il ne va pas le diviser comme around mais comme l’exemple ci-dessous :

Exemple :

;75 ! image ! 75 ! image ! 75 ! image ! 75 ;

(Alignement vertical)

Flex start : aligné depuis le haut

Flex end : depuis le bas

Stretch : sur toute la verticalité du truc

Center : symétrique avec le bas et le haut (utilisé plus souvent que baseline)

Baseline : symétrique selon le texte

Lien de tous les exemples :

https://css-tricks.com/snippets/css/a-guide-to-flexbox/