

Intégration Web

2020 - MMI 1 – TP#2 S1



Danielo JEAN-LOUIS
Développeur front-end

But du cours

- Voir les bases de la programmation d'un site web
 - Langage HTML
 - Langage CSS
- Sensibilisation au web : design et programmation
 - Accessibilité
 - Bonnes pratiques
- Avoir les connaissances pour développer un site web

Récapitulatif du tp précédent

- Le HTML et le CSS ont été inventés par Tim Berners-Lee dans les années 90
- La structure d'un site web est développée en html, langage de balisage / description
- Le css est un langage qui décrit la présentation d'un document html
- Document html : ensemble de blocs appelés "balise"
- Site web : ensemble de pages web, documents au format .html (ou .htm)

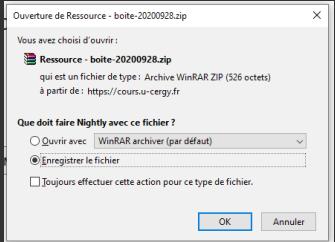
Source :

- <https://github.com/DanYellow/presentations/blob/master/mmi/html/travaux-pratiques/numero-1/presentation.pdf>

Ressource du cours sur ENT

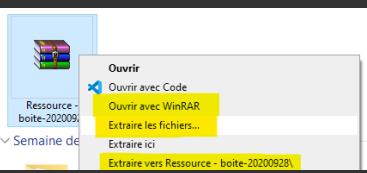
Travaux Pratiques #2 > "Ressource - Boite" sur ENT

Petit point sur les ressources de cours



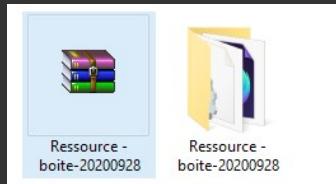
- Les ressources se présentent sous la forme d'une archive une fois téléchargées

- L'archive doit être impérativement extraite sinon vous ne serez pas capables de réaliser correctement les tp



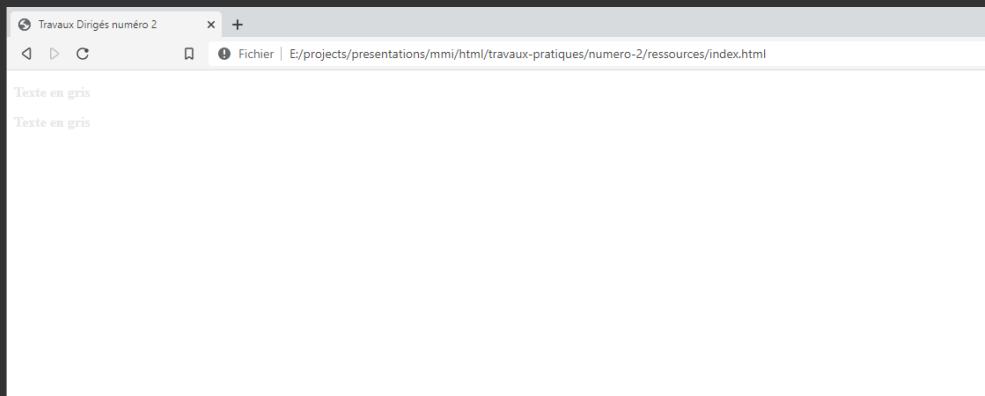
Menu contextuel sur l'archive (clic droit). Choisir un des choix surligné.

Petit point sur les ressources de cours



- Une fois extrait, vous devez obtenir un dossier avec le contenu de l'archive
- Libre à vous de travailler dans le dossier OU en copier le contenu pour le coller ailleurs
- **En aucun vous devez travailler dans l'archive**
- Une fois fait, glissez le dossier dans VS Code

Page générée



Ouvrez le fichier index.html avec votre navigateur préféré
(sauf Internet Explorer)



Pas très lisible, non ?

Point accessibilité – les couleurs

- Beau n'est pas forcément égal à fonctionnel
- Certaines personnes ne perçoivent pas tout le spectre colorimétrique (Daltonien par exemple)
- Pensez aux contrastes
- Comme la sémantique des balises html, les normes d'accessibilité ne sont que des recommandations
- Les gens n'ont pas forcément des écrans super bien calibrés

Sources :

- <https://webaim.org/resources/contrastchecker/>
- <http://www.caf.fr/allocataires/vies-de-famille/vivre-avec-un-handicap/vos-droits/80-des-handicaps-sont-invisibles-le-saviez-vous>

- Si l'accessibilité (ou a11y en anglais) est un ensemble de recommandations, avoir un site accessible rend l'expérience utilisateur bien meilleure

Propriété "background-color"

- Permet de mettre une couleur d'arrière-plan sur (presque) toutes les balises html
- Accepte les valeurs suivantes :
 - Nom de couleur en anglais : red, pink, aliceblue...
 - Couleur au format rgb(a)
 - Couleur au format hexadécimal
 - Et autres

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/color>
- https://www.w3schools.com/css/css_colors.asp
- <https://developer.mozilla.org/fr/docs/Web/CSS/background-color>
- https://developer.mozilla.org/fr/docs/Web/CSS/type_color

- Les trois types de valeurs listés sont ceux les plus utilisés. Il en existe d'autres, à vous de les découvrir

Couleur au format rgb(a)

- Représentation de la couleur sous la forme de quatre composantes :
 - Red (Rouge)
 - Green (Vert)
 - Blue (Bleu)
 - Alpha (Alpha) – Utilisation de la fonction rgba()
- Chaque valeur (sauf la composante alpha 0 – 1 valeur décimale) peut prendre une valeur entre 0 et 255 inclus

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/color>
- https://www.w3schools.com/css/css_colors.asp
- <https://developer.mozilla.org/fr/docs/Web/CSS/background-color>
- https://developer.mozilla.org/fr/docs/Web/CSS/type_color

Couleur au format hexadécimal

- Représentation de la couleur sous la forme de quatre (ou trois) paires de valeurs hexadécimales précédées d'un croisillon (#):

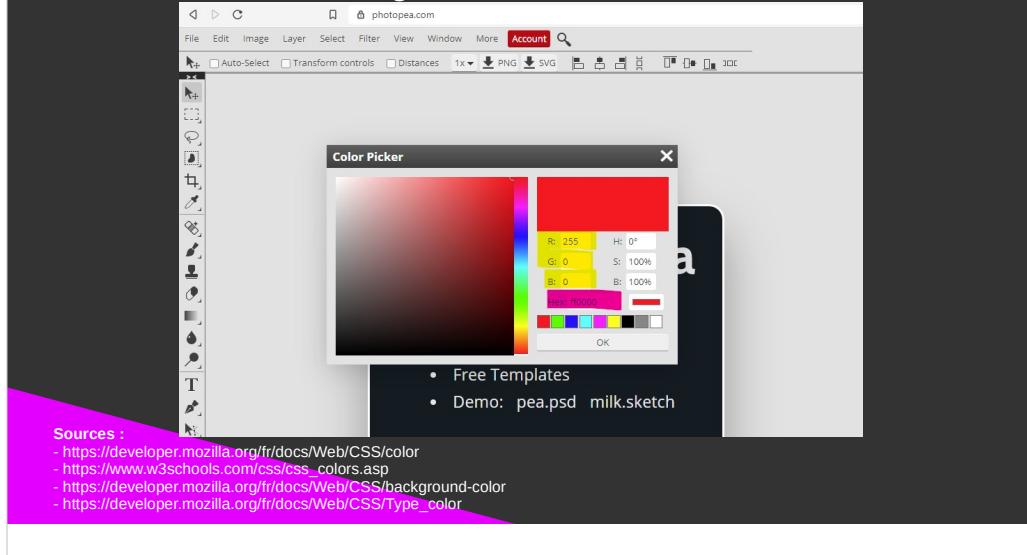


Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/color>
- https://www.w3schools.com/css/css_colors.asp
- <https://developer.mozilla.org/fr/docs/Web/CSS/background-color>
- https://developer.mozilla.org/fr/docs/Web/CSS/type_color

Hexadécimal ou rgb(a) ?

- Il n'y a pas un système moins bon que l'autre
- L'hexadécimal permet de récupérer plus facilement la couleur dans un logiciel



Pratiquons ! - Ajoutons une couleur de fond

Pré-requis :

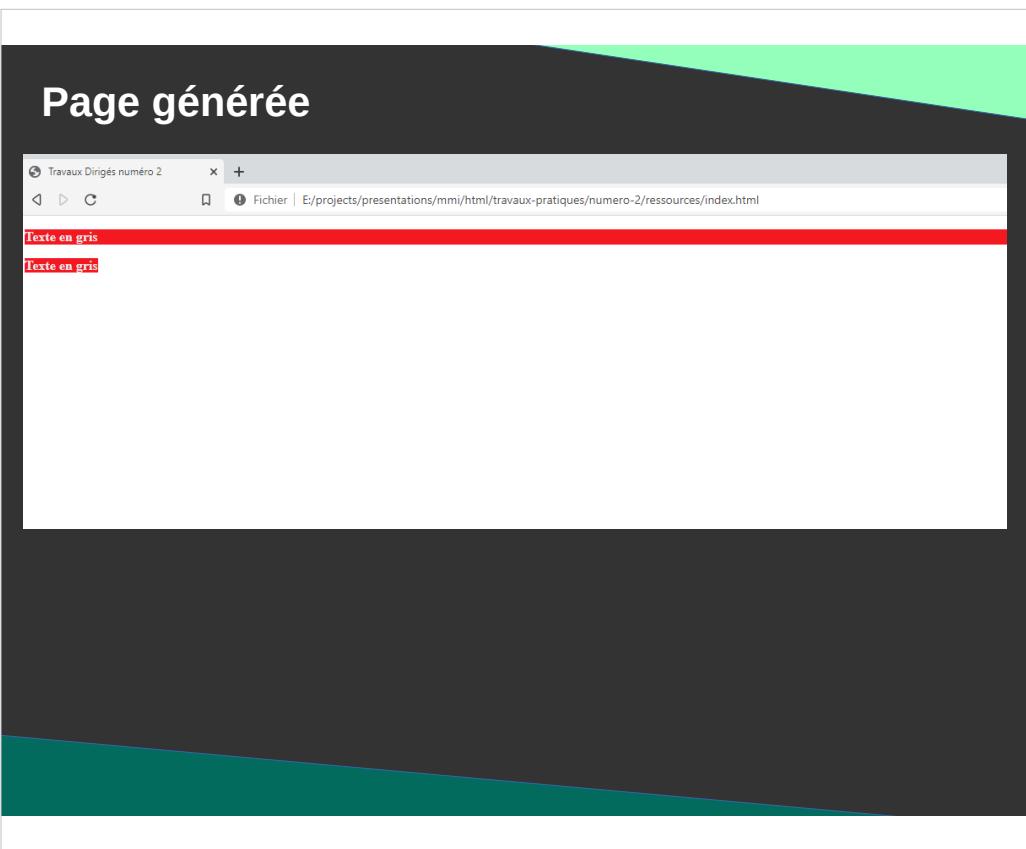
- Avoir la ressource "Boite"
- Ajouter la ligne suivante dans le fichier css style.css

background-color: couleur que vous voulez;

Pour obtenir la classe suivante

```
.informations {  
    color: rgb(230, 230, 230);  
    font-weight: bold;  
    background-color: couleur que vous avez choisi;  
}
```

Rajouter également une déclaration css
background-color à la classe "classe2"



Propriété "background-color"

- La couleur d'arrière-plan peut être aussi définie dans la propriété "background"
 - Utilisation d'image (background-image)
 - Utilisation de dégradés (background-image)
 - Permet l'utilisation de multiples background
 - Et plein d'autres propriétés du background sont raccourcies sous la propriété "background"

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/background>
- <https://developer.mozilla.org/fr/docs/Web/CSS/background-image>

- L'utilisation des multiples arrière-plan fonctionne de la façon suivante :
 - On liste les arrière-plans que l'on souhaite utiliser (images ou dégradés)
 - Le premier élément de la liste est le premier qui s'affiche le plus haut
- Notes sur les multiples arrière-plans :
 - Impossible d'y mettre des couleurs (au pire mettez un dégradé avec deux couleurs identiques)
 - Il n'y a pas de limites sur le nombre de plans

Modèle de boîte - Introduction

- Chaque balise occupe un espace dans la page
- Espace sous forme de rectangle / carré
- *Schématisable* par le schéma suivant

Source :

• https://developer.mozilla.org/fr/docs/Web/CSS/Mod%C3%A8le_de_bo%C3%A8te_CSS

Modèle de boîte

Margin – Marge extérieure

Padding – Marge intérieure

Contenu

largeur x hauteur

Modèle de boite - Contenu

- Par défaut, les balises de type "block" occupent 100 % de la largeur disponible
- Pour la hauteur, les balises occupent la hauteur de leurs contenus → Pas de contenu = hauteur nulle
- On peut modifier la taille de la boite avec les propriétés :
 - width : largeur de boite
 - height : hauteur de boite

Modèle de boîte – Propriétés "height" et "width"

- Accepte les valeurs suivantes :
 - Longueur positive. Par exemple "42px"
 - Un pourcentage. Ex : 20 %
 - "auto" : le navigateur calcule la taille
 - Autres qui sont encore expérimentales

Sources :

- https://developer.mozilla.org/fr/docs/Apprendre/CSS/Building_blocks/Le_modele_de_boite
- <https://developer.mozilla.org/fr/docs/Web/CSS/height>
- <https://developer.mozilla.org/fr/docs/Web/CSS/width>

- Le pourcentage est calculé sur la taille du parent.
Par exemple si une balise possède une largeur de 500px, si son enfant à un largeur de 20 %, alors sa taille calculée est de 100 px

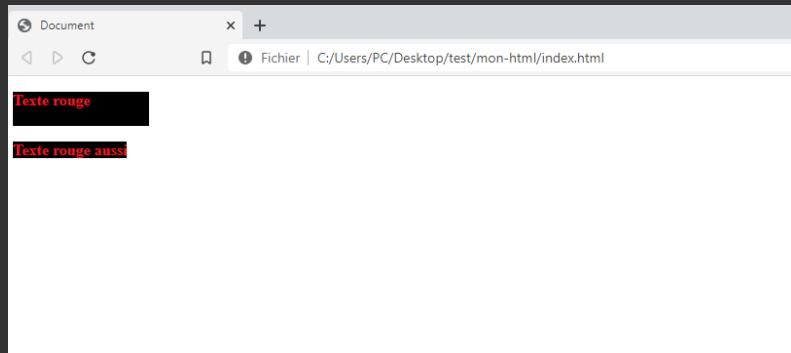
Pratiquons ! - Dimensionnons nos balises

Pré-requis :

- Avoir le contenu du "Pratiquons ! - Ajoutons une couleur de fond"
- Rajouter une "height" et une "width" fixe à la classe css "informations"

```
.informations {  
    color: rgb(230, 230, 230);  
    font-weight: bold;  
    background-color: #000000;  
    width: 140px;  
    height: 35px;  
}
```

Page générée



Modèle de boîte – Propriétés "height" et "width"

- La dimension de la balise "span" n'a pas changé :
 - Ces propriétés ne s'appliquent pas sur les balises de type "inline"
- Les propriétés "width" et "height" peuvent également être limitées par les propriétés suivantes :
 - min-height / max-height
 - min-width / max-width

Sources :

- https://developer.mozilla.org/fr/docs/Apprendre/CSS/Building_blocks/Le_modele_de_boite
- <https://developer.mozilla.org/fr/docs/Web/CSS/height>
- <https://developer.mozilla.org/fr/docs/Web/CSS/width>
- https://developer.mozilla.org/fr/docs/Web/CSS/Mod%C3%A8le_de_bo%C3%A8te_CSS

Modèle de boite - Padding

- Marge intérieure d'une balise
 - Idéal pour mettre de l'espace entre le contenu et le conteneur
- Représenté par la propriété "padding"
- Accepte les valeurs suivantes :
 - Longueur. Exemple : "197px"
 - Pourcentage. Exemple : "25 %"
 - "auto", la navigateur choisit lui-même la dimension

Source :

• <https://developer.mozilla.org/fr/docs/Web/CSS/padding>

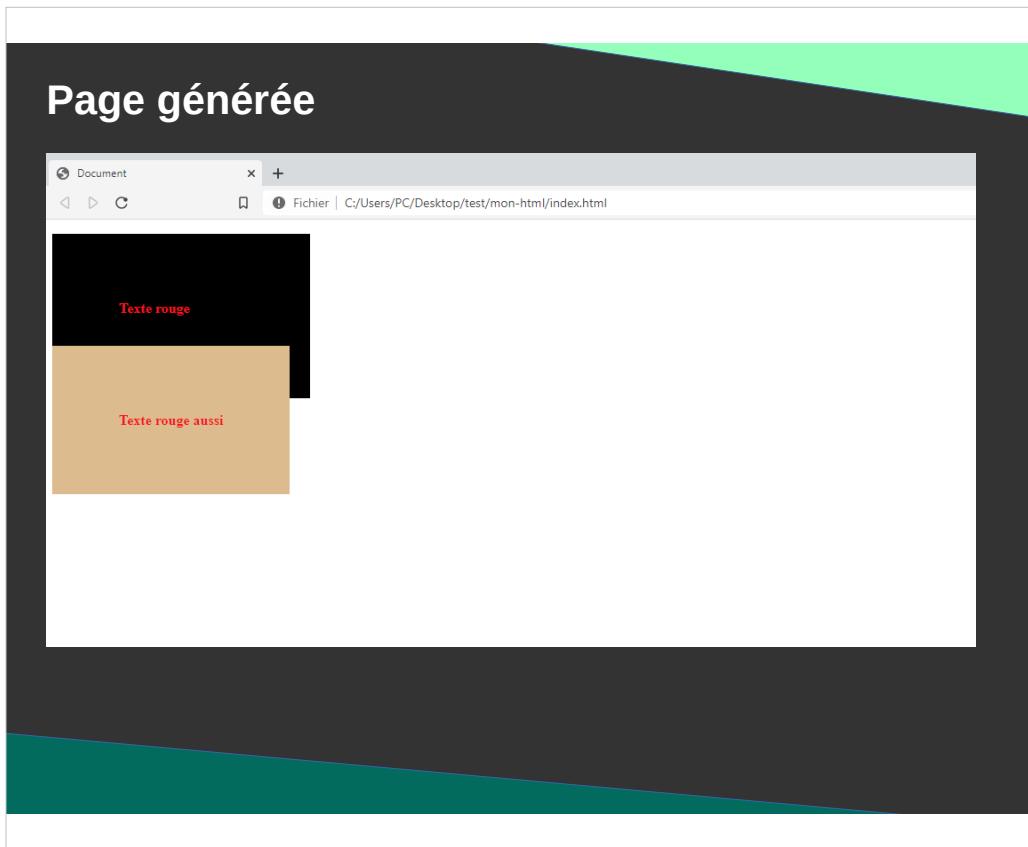
Pratiquons ! - Ajoutons un padding

Pré-requis :

- Avoir le contenu du "Pratiquons ! - Ajoutons une couleur de fond"
- Rajouter la propriété "padding" avec une valeur numérique (suffixée de "px") supérieure à 60.

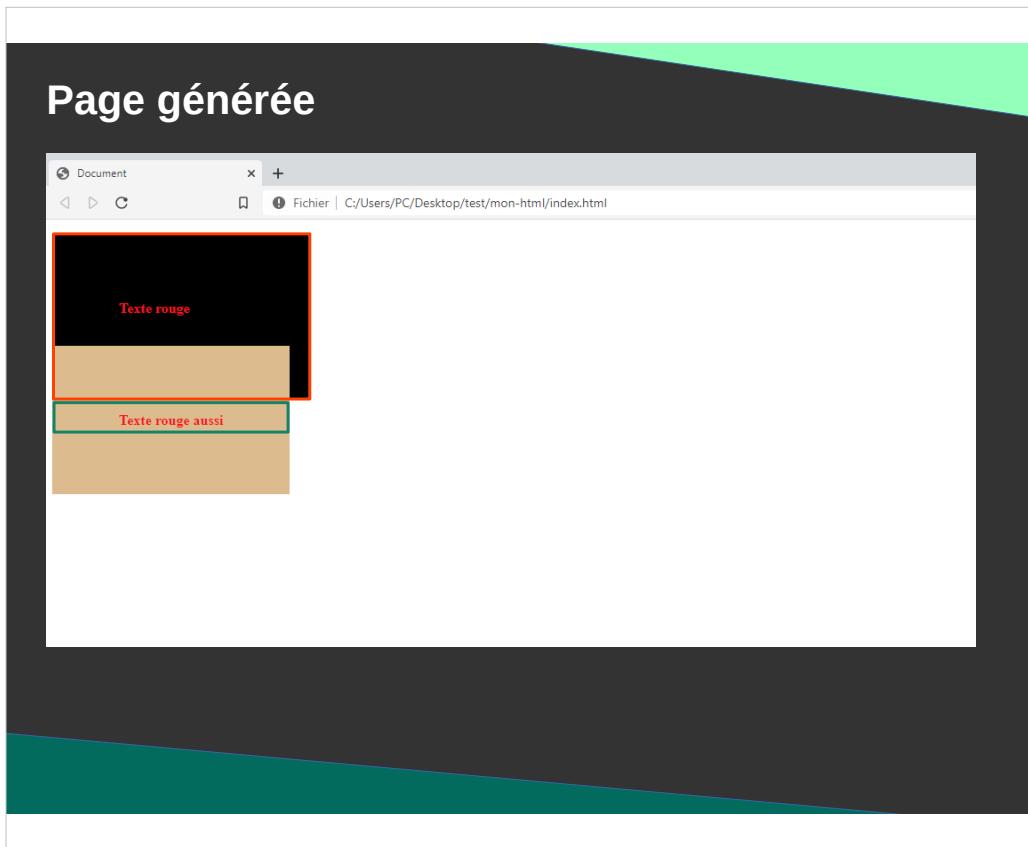
```
.informations {  
    [...]  
    width: 140px;  
    height: 35px;  
    padding: 75px;  
}
```

Source :
• <https://developer.mozilla.org/fr/docs/Web/CSS/padding>



Oui, le résultat est identique à notre essai précédent, mais le style de nos balises est externalisé. C'est bien meilleur pour la lisibilité.

Mais voyons les classes ensemble



Si on dessine des bordures sur la taille réelle des boites dans la page, on remarque que la boite du chevauche le <p> au-dessus

Modèle de boite - Padding

- Ce comportement est lié au fait que le est une balise de type "inline". De fait si le padding est bien visible, l'espace occupé en hauteur par le reste son contenu
- Pour une balise de type "inline", il est préférable d'appliquer la propriété "padding" sur les côtés

Source :

• <https://developer.mozilla.org/fr/docs/Web/CSS/padding>

Modèle de boite - Padding

padding: 20px;

- La propriété "padding" s'applique à tous les côtés. Toutefois, il est possible de séparer chaque côté grâce aux propriétés :
 - padding-top
 - padding-right
 - padding-bottom
 - padding-left

Source :

• <https://developer.mozilla.org/fr/docs/Web/CSS/padding>

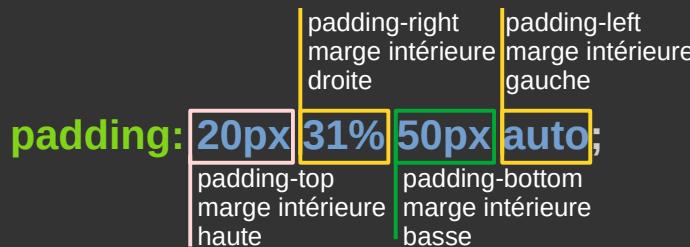
La propriété padding accepte plusieurs valeurs comme valeur. Ainsi il est possible de cibler le ou les côtés voulus en utilisant padding. Par exemple la déclaration "padding: 15px 20px" équivaut à

padding-top: 15px;
padding-right: 20px;
padding-bottom: 15px;
padding-left: 20px;

Modèle de boîte - Padding

padding: 20px;

- La séparation des côtés peut également se faire directement sur la propriété padding, il faut espacer les composantes. Exemple :



- Ceci fonctionne également pour la propriété "margin"

Source :

• <https://developer.mozilla.org/fr/docs/Web/CSS/padding>

La propriété padding accepte plusieurs valeurs comme valeur. Ainsi il est possible de cibler le ou les côtés voulus en utilisant padding. Par exemple la déclaration "padding: 15px 20px" équivaut à

padding-top: 15px;
padding-right: 20px;
padding-bottom: 15px;
padding-left: 20px;

Pratiquons ! - Corrigeons le padding

Pré-requis :

- Avoir le contenu du "Ajoutons un padding"
- Rajouter les propriétés "padding-left", "padding-right", "padding-top" et "padding-bottom" directement dans la balise .

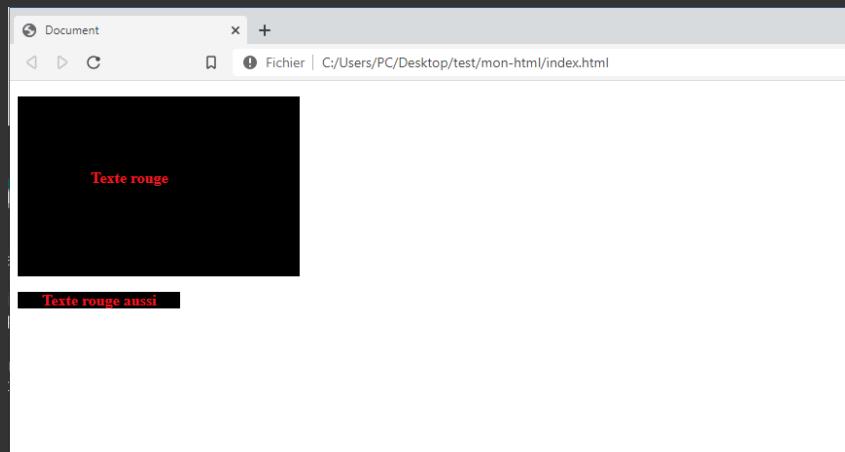
```
<span  
    class="informations classe2"  
    style="padding-left: 20px; padding-right: 20px;  
    padding-top: 0; padding-bottom: 0;">  
    Texte aussi rouge  
</span>
```

Source :

• <https://developer.mozilla.org/fr/docs/Web/CSS/padding>

Pensez à sauvegarder après
chaque modification

Page générée





Et ça a fonctionné ?!

On va introduire une nouveau concept en css : la spécificité ou poids du css

CSS – Spécificité ou poids du css

Règle de base : la dernière déclaration css s'applique à un élément html... mais les règles suivantes surpassent la règle :

- Utilisation d'un id (#) comme sélecteur css
- Utilisation de l'attribut "style" pour le style
- Sélecteur css plus précis
 - Le sélecteur ".infos" est plus léger que "div .infos"
Les déclarations css de "div .infos" seront donc appliquées en premier même si placé plus haut dans le code.

Sources :

- https://developer.mozilla.org/fr/docs/Apprendre/CSS/Building_blocks/Cascade_et_inheritance#Spécificité
- <https://www.zonecss.fr/cours-css/poids-des-selecteurs-css.html>
- https://developer.mozilla.org/fr/docs/Apprendre/CSS/Building_blocks/Cascade_et_inheritance#important

Par "appliquées en premier", il faut comprendre que les déclarations liées sont fusionnées.

CSS – Spécificité ou poids du css - Fusion

```
.infos p {  
    color: blue;  
}  
  
.infos {  
    color: red;  
}
```



```
<div class="infos">  
    <p>Mon texte</p>  
</div>
```

Le sélecteur ".infos p" est plus précis que ".infos" même si ce dernier est plus bas dans le code, le texte est donc bleu

Sources :

- https://developer.mozilla.org/fr/docs/Apprendre/CSS/Building_blocks/Cascade_et_héritage#Spécificité
- <https://www.zonecss.fr/cours-css/poids-des-selecteurs-css.html>
- https://developer.mozilla.org/fr/docs/Apprendre/CSS/Building_blocks/Cascade_et_héritage#important

CSS – Spécificité ou poids du css

- Utilisation de la valeur spéciale "!important"
 - **A utiliser avec extrême parcimonie**
 - S'applique après la valeur d'une propriété css :
padding-left: 20px!important;
 - Et lui permet de passer devant toutes les autres règles
 - Se situe en haut de toutes les règles de la spécificité du css
 - **A utiliser qu'en dernier cours**

Sources :

- https://developer.mozilla.org/fr/docs/Apprendre/CSS/Building_blocks/Cascade_et_héritage#Spécificité
- <https://www.zonecss.fr/cours-css/poids-des-selecteurs-css.html>
- https://developer.mozilla.org/fr/docs/Apprendre/CSS/Building_blocks/Cascade_et_héritage#!important

- Vous travaillerez peut être sur des projets avec des fichiers CSS que vous ne pourrez pas éditer. Ceci est un dernier recours.

Modèle de boite - Margin

- Marge extérieure d'une balise
 - Idéal pour mettre de l'espace en entre deux balises juxtaposées (verticalement ou horizontalement)
- Représenté par la propriété "margin"
- Accepte les valeurs suivantes :
 - Longueur (qui peut être négative). Ex: "197px"
 - Pourcentage (qui peut être négatif). Ex: "25 %"
 - "auto", la navigateur choisit lui-même la dimension.
Permet également de centrer horizontalement,
uniquement si la balise a une taille fixe

Source :

• <https://developer.mozilla.org/fr/docs/Web/CSS/margin>

Pratiquons ! - Ajoutons un margin

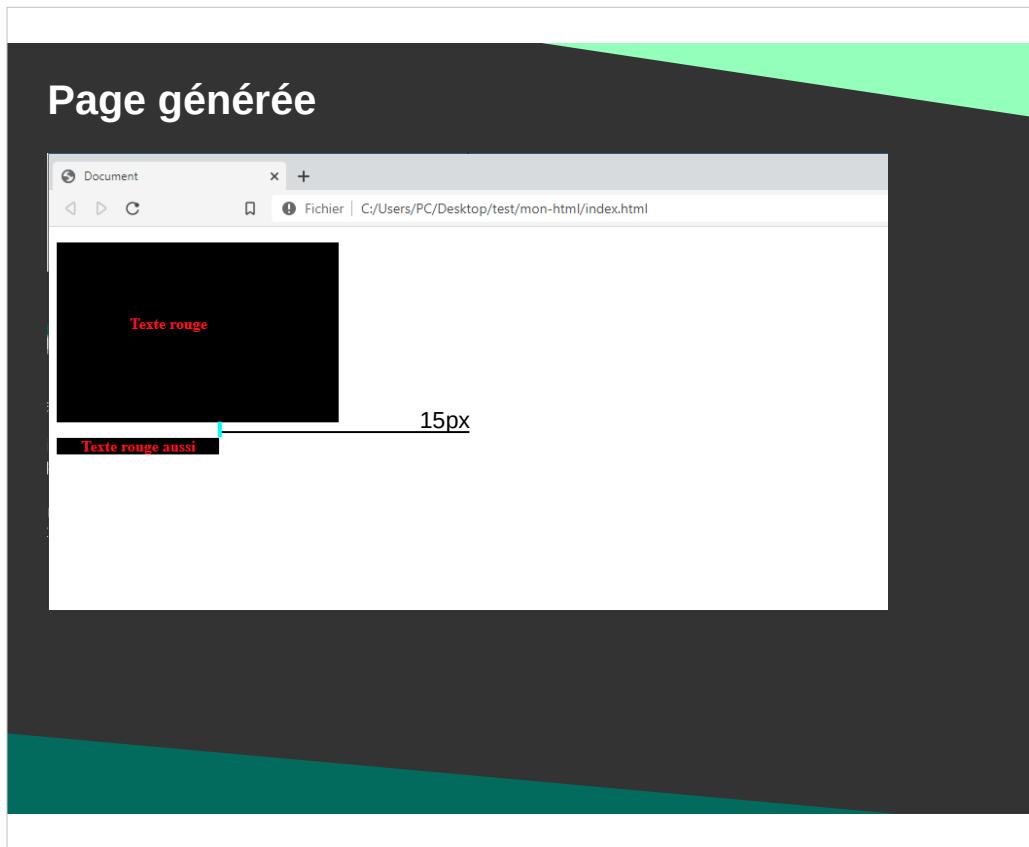
Pré-requis :

- Avoir le contenu du "Pratiquons ! - Ajoutons un margin"
- Rajouter la propriété "margin" avec une valeur numérique

```
.informations {  
    [...]  
    margin: 15px;  
}
```

Source :

• <https://developer.mozilla.org/fr/docs/Web/CSS/margin>



- Malgré la présence d'un margin haut et bas de 15 px sur les deux balises, seul une hauteur s'est appliquée, les valeurs ont été fusionnées

Modèle de boite - Margin

- Les marges basses et hautes de balises juxtaposées sont toujours fusionnées (sauf cas spéciaux voir sources), la marge la plus grande des deux est utilisée
- Les marges basses et hautes de balises de type "inline" ne s'appliquent pas

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/margin>
- https://developer.mozilla.org/fr/docs/Web/CSS/Mod%C3%A8le_de_bo%C3%A8te_CSS/Fusion_des_marges

Modèle de boite - Margin

margin: 42px;

- La propriété "margin" s'applique à tous les côtés. Toutefois, il est possible de séparer chaque côté grâce aux propriétés :
 - margin-top
 - margin-right
 - margin-bottom
 - margin-left

Source :

• <https://developer.mozilla.org/fr/docs/Web/CSS/margin>

La propriété padding accepte plusieurs valeurs comme valeur. Ainsi il est possible de cibler le ou les côtés voulus en utilisant padding. Par exemple la déclaration "padding: 15px 20px" équivaut à

padding-top: 15px;
padding-right: 20px;
padding-bottom: 15px;
padding-left: 20px;

Modèle de boite – Rappel

Margin – Marge extérieure

Padding – Marge intérieure

Contenu

largeur x hauteur

Quiz

"Test - Modèle de boîte et couleurs" sur ENT

Conditions du test :

- ~5 minutes
- Tentatives illimitées
- Meilleure des notes prise en compte

Modèle de boîte complet

Margin – Marge extérieure

Bordure

Padding – Marge intérieure

Contenu

largeur x hauteur

Bordures

- Propriété css définie par le terme "border"
- Permet d'afficher une bordure et d'en définir :
 - Couleur
 - Style
 - Épaisseur
- Applique le style sur les quatre côtés de la balise

border: 2px solid blue;

Source :
• <https://developer.mozilla.org/fr/docs/Web/CSS/border>

- L'ordre des propriétés ne compte pas, le parseur css des navigateurs sont suffisamment malins pour comprendre que "2px" est l'épaisseur de la bordure
- Si la propriété "border" manque de certaines valeurs, ce sont celles par défaut qui s'appliquent

Bordures - Couleur

- Valeur représentant la couleur de la bordure
- Accepte comme valeur, les valeurs d'une couleur habituel (vu précédemment)
- Représente la couleur des quatre côtés de la balise
- Peut-être décorrélé de la propriété "border" avec la propriété "border-color"

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/border>
- <https://developer.mozilla.org/fr/docs/Web/CSS/border-color>

Bordures - Style

- Valeur représentant la style de la bordure
- Accepte comme valeur des valeurs définies (voir sources)
- Représente le style de bordure des quatre côtés de la balise
- Peut-être décorrélé de la propriété "border" avec la propriété "border-style"

Source :

- <https://developer.mozilla.org/fr/docs/Web/CSS/border>
- <https://developer.mozilla.org/fr/docs/Web/CSS/border-style>

Bordures – Style – Exemples de style de bordure

dotted		Affiche une série de points ronds. L'espace entre ces points n'est pas défini par la spécification et est donc laissé à l'implémentation. Le rayon des points correspond à la moitié de <code>border-width</code> .
dashed		Affiche une série de tirets. La taille de ces tirets n'est pas définie par la spécification et est laissée à l'implémentation.
solid		Affiche une ligne droite continue.
double		Deux lignes droites sont affichées dont la somme des épaisseurs correspond à la valeur fournie par <code>border-width</code> .
groove		La bordure crée un effet 3D donnant l'impression qu'elle a été gravée dans le document. On obtient un effet opposé à <code>ridge</code> .
ridge		La bordure crée un effet 3D donnant l'impression que la bordure ressort du document. L'effet obtenu est opposé à celui obtenu avec <code>groove</code> .

<https://developer.mozilla.org/fr/docs/Web/CSS/border-style>

Source :

- <https://developer.mozilla.org/fr/docs/Web/CSS/border>
- <https://developer.mozilla.org/fr/docs/Web/CSS/border-style>

Bordures - Width

- Valeur représentant l'épaisseur de la bordure
- Accepte comme valeur des valeurs définies (voir source) ainsi qu'une longueur
- Représente l'épaisseur des quatre côtés de la balise
- Peut-être décorrélé de la propriété "border" avec la propriété "border-width"

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/border>
- <https://developer.mozilla.org/fr/docs/Web/CSS/border-width>

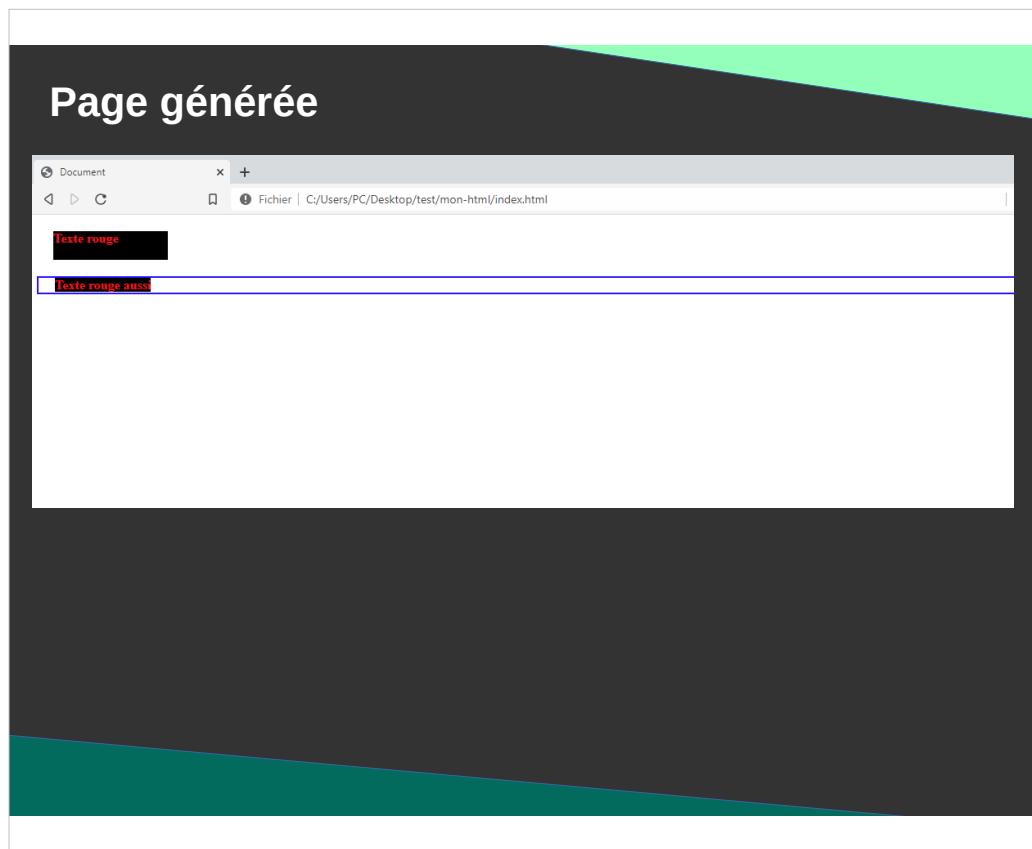
Pratiquons ! - Essayons les bordures

Pré-requis :

- Avoir le contenu du "Pratiquons ! - Corrigeons le padding"
- Rajouter une classe css contenant une déclaration css permettant d'afficher une bordure et l'ajouter à une balise

```
.ma-bordure {  
    border: 2px solid blue;  
}
```

Vous pouvez mettre cette déclaration dans une classe spécifique si vous le souhaitez



Bordures

- La propriété "border" peut être décomposée par côté :
 - border-top
 - border-right
 - border-bottom
 - border-left

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/border>

Bordures

- Les propriétés "border-width/color/style" peuvent être décomposées par côté :
 - border-top-{propriété}
 - border-right-{propriété}
 - border-bottom-{propriété}
 - border-left-{propriété}

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/border>

Bordures

- La propriété "border" a une propriété "similaire" : outline
- L'outline se place au-dessus de la bordure
- La propriété "outline" diffère de "border" en deux points :
 - Ne prend pas de place dans le flux
 - Ne permet pas d'avoir un design différent sur chaque côté

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/border>
- <https://stackoverflow.com/questions/1158515/difference-between-outline-and-border> - en
- <https://developer.mozilla.org/fr/docs/Web/CSS/outline>

Bordures

- Il est possible de mettre une image en bordure avec la propriété "border-image"
 - L'image sera la même sur tous les côtés.
La propriété "border-image-left" (par exemple) n'existe pas
- Il est possible de faire des bordures arrondies grâce à la propriété "border-radius"

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/border-image>
- <https://developer.mozilla.org/fr/docs/Web/CSS/border-radius>

Modèle de boîte vraiment complet

Margin – Marge extérieure

Bordure / Outline

Padding – Marge intérieure

Contenu

largeur x hauteur

Modèle de boîte en vrac

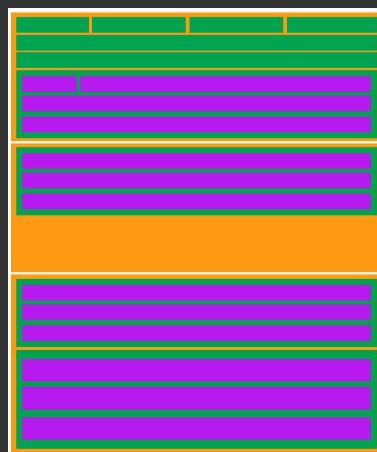
- La propriété "visibility" permet de cacher ou non un élément html tout en conservant sa place dans la page
- La propriété "overflow" permet de gérer le dépassement de contenu

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/overflow>
- <https://developer.mozilla.org/fr/docs/Web/CSS/visibility>

Disposition des balises

- Par défaut, les balises ("inline" ou "block") se suivent et s'empilent respectivement
 - On dit qu'elles sont dans le flux



Disposition des balises – Propriété position

- Modifiable grâce à la propriété "position"
- Cinq valeurs possibles :
 - static (valeur par défaut)
 - relative
 - absolute
 - fixed
 - sticky
- Propriété fonctionnant aussi bien sur les balises de types "block" que "inline"

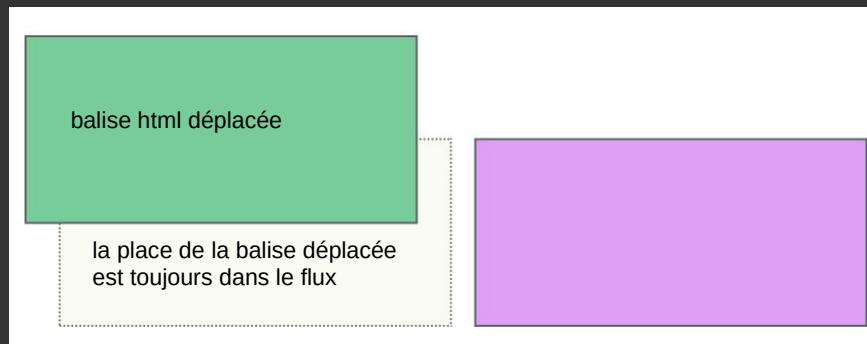
- Tim Berners-Lee a également inventé :
 - le www (world wide web) ou web, système incluant :
 - adresses web
 - protocole HTTP (communication client-serveur)
 - Le HTML
 - Il invente également le W3C
- Le HTML n'est pas un langage de programmation.
Il ne permet pas de mettre en place des conditions (if, else, boucle while...)

Disposition des balises – Propriété position

- Couplée aux propriétés
 - top
 - left
 - right
 - bottom
 - z-index
- Propriétés utilisables uniquement si la propriété "position" a une valeur différente de "static"

Propriété position – valeur "relative"

- Permet de sortir d'une balise html du flux tout en gardant sa place dans le flux



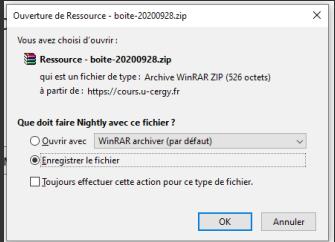
Propriété position – valeur "relative"

- Le point d'origine est l'emplacement d'origine de la balise dans le flux de la page

Ressource du cours sur ENT

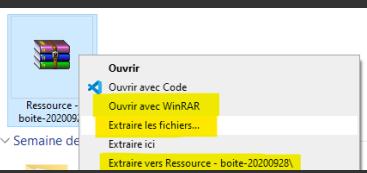
Travaux pratiques #2 > "Ressource position"

Petit point sur les ressources de cours



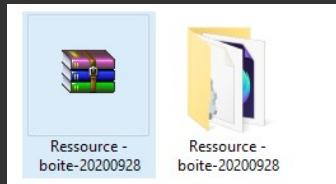
- Les ressources se présentent sous la forme d'une archive une fois téléchargées

- L'archive doit être impérativement extraite sinon vous ne serez pas capables de réaliser correctement les tp



Menu contextuel sur l'archive (clic droit). Choisir un des choix surligné.

Petit point sur les ressources de cours



- Une fois extrait, vous devez obtenir un dossier avec le contenu de l'archive
- Libre à vous de travailler dans le dossier OU en copier le contenu pour le coller ailleurs
- **En aucun vous devez travailler dans l'archive**

Pratiquons ! - Découvrons la position "relative"

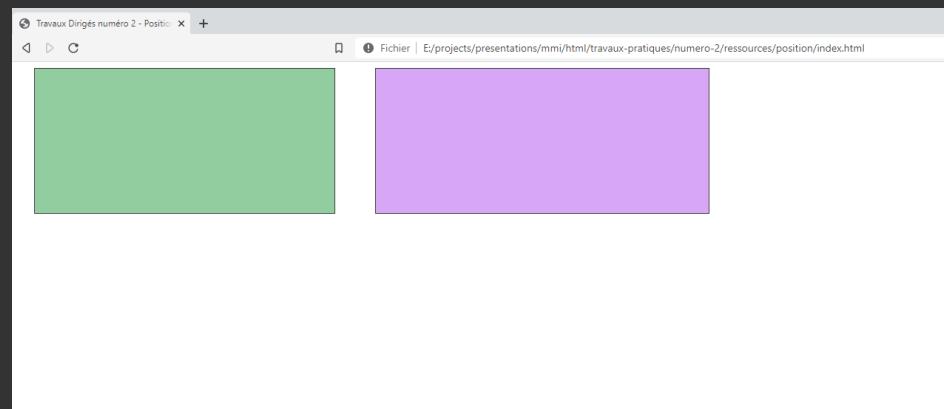
Pré-requis :

- Avoir le contenu de la ressource "ressource position" sur ENT
- Rajouter la classe "position-relative" à la balise ayant la classe "is-green"

Source :

• <https://github.com/DanYellow/presentations/tree/master/mmi/html/travaux-pratiques/numero-2/ressources/position>

Page générée



Pratiquons ! - Découvrons la position "relative"

Pré-requis :

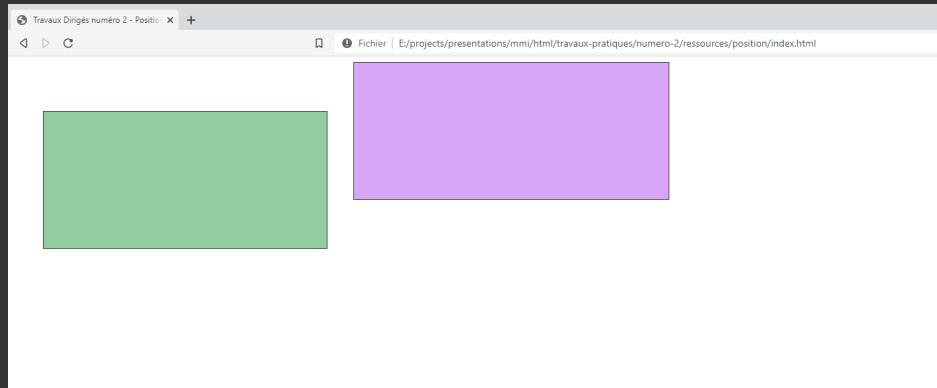
- Avoir le contenu de la ressource "ressource position" sur ENT
- Changer la ou les valeurs "auto" dans la classe "position-relative" par une longueur

Source :

• <https://github.com/DanYellow/presentations/tree/master/mmi/html/travaux-pratiques/numero-2/ressources/position>

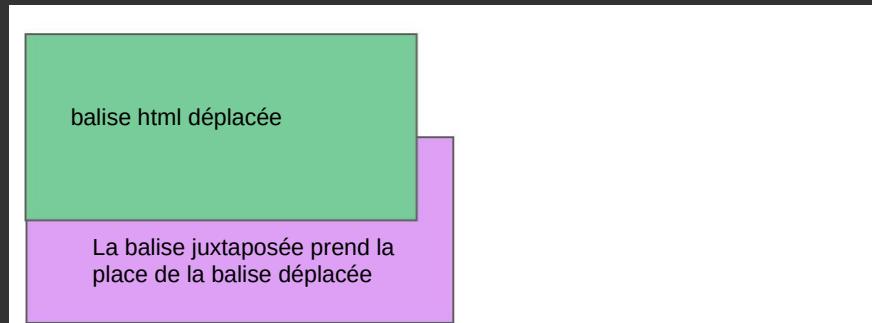
Une longueur est un nombre décimal suivi d'une unité, il en existe plusieurs mais les plus utilisées sont le pixel, l'em (un peu moins de nos jours), le rem, le vh, le vw, le vmin et le vmax

Page générée



Propriété position – valeur "absolute"

- Permet de sortir totalement une balise html du flux. Les balises juxtaposées (et dans le flux) prennent sa place



Propriété position – valeur "absolute"

- Contrairement à la valeur "relative", l'origine du repère est le coin en haut à gauche de la page

Pratiquons ! - Découvrons la position "absolute"

Pré-requis :

- Avoir le contenu de la ressource "ressource position" sur ENT
- Remplacer la classe "position-relative" par la classe "position-absolute" à la balise ayant la classe "is-green"
- Changer la ou les valeurs "auto" dans la classe "position-absolute" par une longueur

Sources :

- <https://github.com/DanYellow/presentations/tree/master/mmi/html/travaux-pratiques/numero-2/ressources/position>
- <https://developer.mozilla.org/fr/docs/Web/CSS/position>



Propriété position – valeur "absolute"

- Par défaut une balise en position "absolute" a pour origine du repère le coin en haut à gauche de la page
- Si son parent à la position "relative", l'origine change et devient celle du parent
- Utiliser les propriétés "right" et "bottom" calcule la position de la balise par rapport à la droite et le bas du repère

Propriété position – valeur "fixed"

- Fixe (fixed) la position de la balise sur la fenêtre → Même quand la page défile la balise reste visible
- La balise sort et perd sa place dans le flux. Les balises juxtaposées prennent sa place

Pratiquons ! - Découvrons la position "fixed"

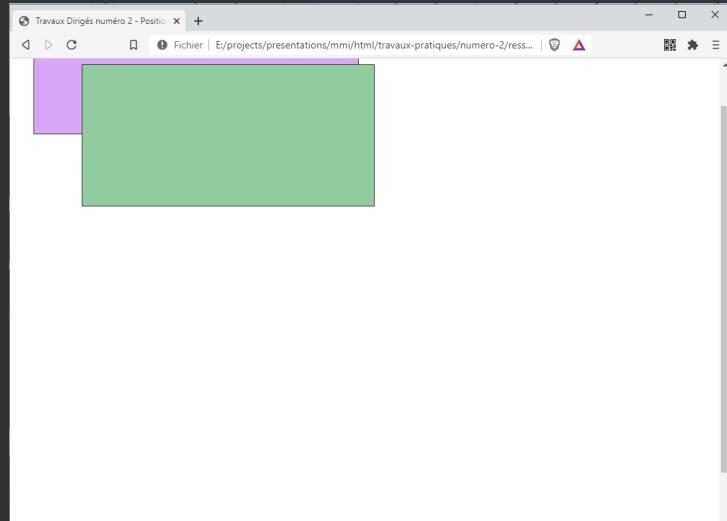
Pré-requis :

- Avoir le contenu de la ressource "ressource position" sur ENT
- Remplacer la classe "position-absolute" par la classe "position-fixed" à la balise ayant la classe "is-green"
- Ajouter la classe "body-scroll" à la balise <body>
- Changer la ou les valeurs "auto" dans la classe "position-fixed" par une longueur

Sources :

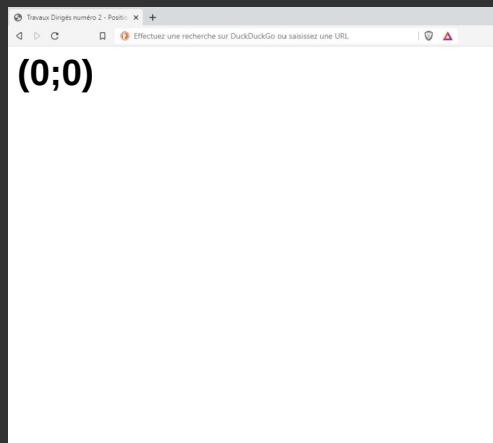
- <https://github.com/DanYellow/presentations/tree/master/mmi/html/travaux-pratiques/numero-2/ressources/position>
- <https://developer.mozilla.org/fr/docs/Web/CSS/position>

Page générée



Page un peu défilée

Propriété position – position "fixed"



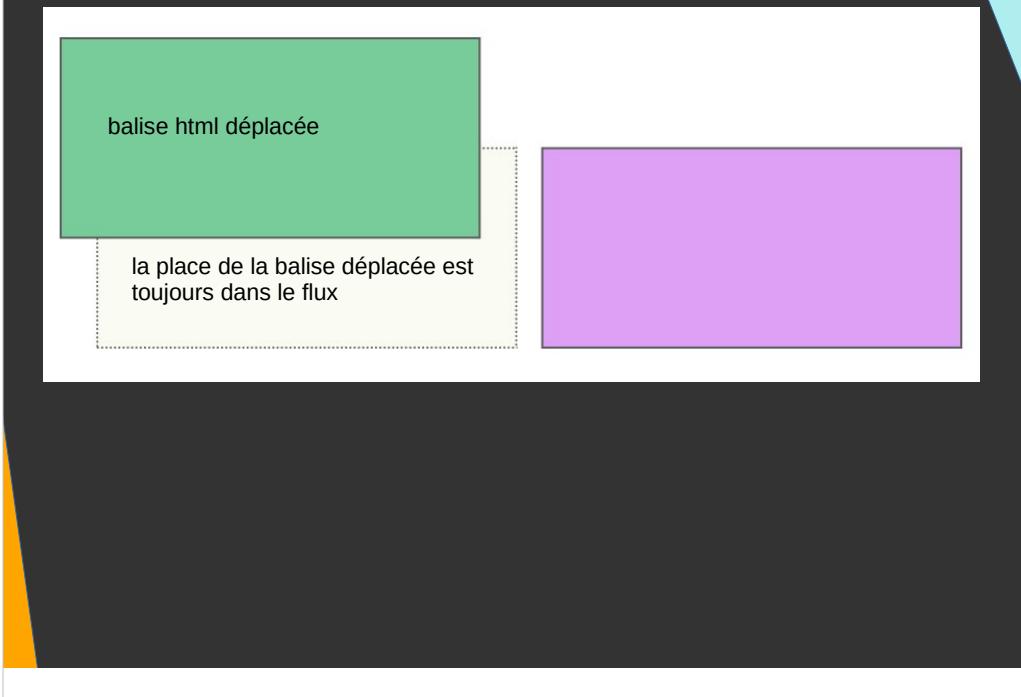
- En haut à gauche se trouve l'origine du repère d'un site web

Propriété position – valeur "sticky"

- Colle (sticky) la balise sur la fenêtre quand sa position (dans son ancêtre de défilement le plus proche) est dépassé
 - Si la balise parente n'est pas assez haute alors l'effet de la position sticky ne s'appliquera pas
- Fonctionne aussi bien avec le défilement (scroll) horizontal que vertical
- Comme la position "relative" la position "sticky" sort la balise du flux mais la balise garde l'espace dans le flux

- Tim Berners-Lee a également inventé :
 - le www (world wide web) ou web, système incluant :
 - adresses web
 - protocole HTTP (communication client-serveur)
 - Le HTML
- Il invente également le W3C
- Le HTML n'est pas un langage de programmation.
Il ne permet pas de mettre en place des conditions (if, else, boucle while...)

Propriété position – valeur "sticky"



- Tim Berners-Lee a également inventé :
 - le www (world wide web) ou web, système incluant :
 - adresses web
 - protocole HTTP (communication client-serveur)
 - Le HTML
 - Il invente également le W3C
- Le HTML n'est pas un langage de programmation.
Il ne permet pas de mettre en place des conditions (if, else, boucle while...)

Pratiquons ! - Découvrons la position "sticky"

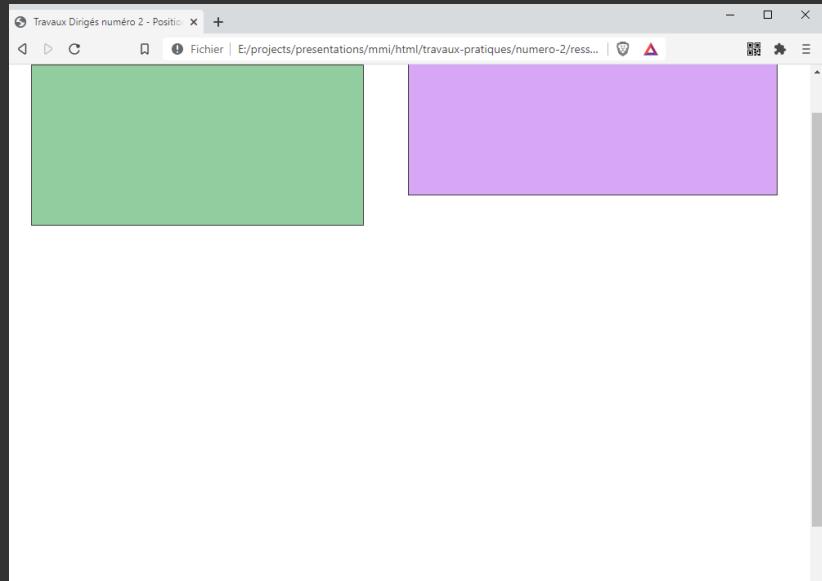
Pré-requis :

- Avoir le contenu de la ressource "ressource position" sur ENT
- Remplacez la classe "position-fixed" par la classe "position-sticky" à la balise ayant la classe "is-green"
- Ajoutez la classe "body-scroll" à la balise <body>
- Changez la valeur "auto" dans la classe "position-sticky" par une longueur
- Défilez la page

Sources :

- <https://github.com/DanYellow/presentations/tree/master/mmi/html/travaux-pratiques/numero-2/ressources/position>
- <https://developer.mozilla.org/fr/docs/Web/CSS/position>

Page générée



Propriété z-index

- Active le "mode 3D" d'une balise html en permettant de se déplacer sur l'axe z
- Accepte un entier en guise de valeur
 - Plus la valeur est haute plus la balise est haute dans la "tour" de balise
- Fonctionne comme une pile d'assiettes :
 - Deux balises ne peuvent pas être au même niveau
 - Si deux z-index, dans le même contexte, ont la même valeur, le navigateur tranchera

Source :

• <https://developer.mozilla.org/fr/docs/Web/CSS/z-index>

- Si deux balises, possédant la propriété "z-index" ont la même valeur, le navigateur tranchera

Propriété z-index

- Chaque balise crée son propre contexte "z-index"
 - Les enfants de la balise ne sont pas comparés aux enfants d'une autre balise
- Valeur sans unité
- Fonctionne avec une balise dont la propriété "position" est différente de "static"

Source :

• <https://developer.mozilla.org/fr/docs/Web/CSS/z-index>

Récapitulatif

- La propriété "position" permet de changer le comportement d'une balise vis à vis du flux de la page
- La propriété "background-color" permet d'ajouter une couleur d'arrière-plan à une balise
- Il est possible d'outrepasser la spécificité naturelle des règles via plusieurs techniques

Quiz

"Test - Position" sur ENT

Conditions du test :

- ~5 minutes
- Tentatives illimitées
- Meilleure des notes prise en compte

Jusqu'à présent, notre mise en page n'a été composée que de colonnes, mais il est possible de faire des mises en page plus complexes





**Notamment juxtaposer des balises tout
en les gardant dans le flux**

-

Mise en page - Historique

- Au début du web, les développeurs utilisaient des tableaux pour réaliser les mise en page
- Ensuite la propriété "float"
- Puis la valeur "inline-block" pour la propriété "display"
- Et puis le CSS3 apporta flexbox

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/float>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/table>
- <https://www.alsacreations.com/article/lire/1209-display-inline-block.html>
- <https://putaindecode.io/podcasts/s02e04-css-layout> – podcast en français

- L'utilisation de tableaux pour faire de la mise en page est à bannir. Elle doit être utilisée que pour faire des e-mail et encore, il existe des outils pour faire des e-mails sans casse-tête
- La propriété "float" est également à éviter, elle crée plein d'effets de bords qui nécessitent énormément d'astuces pour les contre-carrer
- "display : inline-block" fonctionne mais montre ses limites. Il y a notamment des astuces à utiliser pour avoir le comportement attendu. L'idée du "display: inline-block" étant de mixer les types d'affichage "inline" et "block"

Mise en page - Flexbox

- Propriété apparue avec CSS3
- Compatible avec tous les navigateurs modernes
- Permet des mises en page complexes **sur une seule dimension** (colonne ou ligne)

Sources :

- https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Flexible_Box_Layout
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/> - anglais
- <https://putaindecode.io/podcasts/s02e04-css-layout> – podcast en français
- <https://philipwalton.github.io/solved-by-flexbox/> - anglais

Mise en page - Flexbox

- Propriété s'applique sur le parent
 - La balise parente doit avoir la valeur "flex" pour la propriété "display"

display: flex;

```
<body>
  <div class="est-flexbox"> <!-- cette balise possède la déclaration css "display: flex" -->
    <select name="" id="">[...]</select>
    <div class="enfant">[...]</div>
    <ul></ul>
  </div>
</body>
```

Les enfants sont maintenant des éléments flexbox

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/flex>
- <https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/3298561-faites-votre-mise-en-page-avec-flexbox>
- Liste bugs flexbox : <https://github.com/philipwalton/flexbugs> - anglais

La déclaration "display : flex"
s'applique sur le parent, mais c'est
sur les enfants que l'on voit le
changement de mise en page

Mise en page - Flexbox

- Possibilité de faire une mise en page en colonne ou en ligne (propriété "flex-direction")
- Gestion des espaces et du positionnement dans l'espace
- Changement de l'ordre des éléments
- Les enfants d'une balise flexbox peuvent eux-même être flexbox

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/flex>
- <https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/3298561-faites-votre-mise-en-page-avec-flexbox>

Mise en page - Flexbox

- La propriété "flex-direction" définit l'axe principal de l'alignement
 - propriété "justify-content" définit le comportement de l'axe primaire
 - propriété "align-items" définit le comportement de l'axe secondaire

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/flex>
- <https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/3298561-faites-votre-mise-en-page-avec-flexbox>

- Si la valeur de la propriété "flex-direction" est "row" alors la valeur de la propriété "justify-content" s'appliquera au niveau de l'axe horizontal. Conséquemment la valeur de la propriété "align-items" s'appliquera au niveau de l'axe vertical
- Les propriétés "justify-content" et "align-items" permettent notamment de définir l'alignement des éléments

Pratiquons ! - Découvrons flexbox

Pré-requis :

- Avoir le contenu de la ressource "ressource flexbox" sur ENT – **Bien penser à extraire le contenu de l'archive pour travailler**
- Rajouter la classe "affichage-flexbox" dans le fichier style.css
- Ajouter la déclaration "display: flex;" à la classe suscitée

Pratiquons ! - Découvrons flexbox



On remarque deux choses une fois le parent passé en mode flexbox (`flex-direction: row` – valeur par défaut):

- Les enfants occupent la taille de leur contenu en largeur
- Les enfants occupent en hauteur, celle du voisin le plus haut

Pratiquons ! - Découvrons flexbox

Pré-requis :

- Avoir le contenu de la ressource "ressource flexbox" sur ENT et effectué les modifications demandées dans le premier "découvrons flexbox"
- Ajouter la déclaration "flex-direction: column;" à la classe "affichage-flexbox" dans le fichier CSS

Pratiquons ! - Découvrons flexbox



Mise en page - Flexbox

- En ajoutant "-reverse", il est possible d'inverser l'ordre des enfants d'un conteneur flexbox
 - Exemple "flex-direction : row-reverse"
- En mettant la déclaration "margin:auto" aux enfants, ils se centrent tous seuls verticalement et horizontalement
- Un enfant d'un élément flexbox peut être lui-même flexbox

Sources :

• <https://developer.mozilla.org/fr/docs/Web/CSS/flex>

- Le suffixe "reverse" est très utile pour certaines mise en page, mais surtout pour les langues qui se lisent de droite à gauche.

Pratiquons ! - Découvrons flexbox, mise en page

Pré-requis :

- Avoir le contenu de la ressource "ressource mise en page" sur ENT
- Ouvrez le fichier index.html dans votre navigateur favori (pas IE) et vous devriez obtenir
- Ouvrez le fichier jpg "design-mise-en-page-flexbox" contenu dans le même dossier

Pratiquons ! - Découvrons flexbox, mise en page

Accueil
Galerie
Contact

MMI

GEII

TC

TSI

600 × 480

600 × 480

600 × 480

600 × 480

Pratiquons ! - Découvrons flexbox, mise en page

Accueil Galerie Contact

MMI

GEII

TC

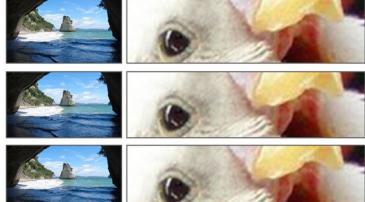
TSI

Lore ipsum dolor sit amet, consecetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lore ipsum dolor sit amet, consecetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lore ipsum dolor sit amet, consecetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lore ipsum dolor sit amet, consecetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.



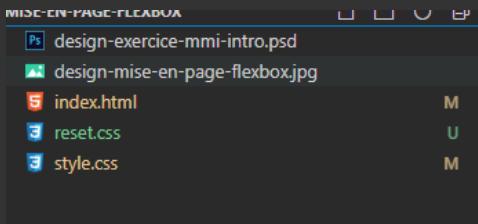
Pratiquons ! - Découvrons flexbox, mise en page

Buts de la pratique :

- Partir du code fourni et le transformer pour qu'il ressemble au fichier "design-mise-en-page-flexbox.jpg" fourni (slide précédente)
- Réaliser son "premier" site
- Voir plus en détails la propriété flexbox et d'autres vues en cours
- Apprendre les bonnes pratiques de développement de site web

Pratiquons ! - Découvrons flexbox, mise en page

Contenu de la ressource



Reset css

- Permet de réinitialiser les styles par défaut des navigateurs, toutes les balises ont un style "nul"
 - Par exemple les balises h1 et p ont la même taille
- Attention tous les resets ne valent pas
 - Eric Meyer Reset le plus populaire
- S'oppose à la philosophie du normalize.css
 - Les balises conservent leur style par défaut mais il est uniforme entre les navigateurs

Sources :

- <https://www.alsacreations.com/astuce/lire/36-reset-css.html>
- <https://meyerweb.com/eric/tools/css/reset/> - anglais
- <https://stackoverflow.com/questions/6887336/what-is-the-difference-between-normalize-css-and-reset-css> - anglais
- <https://necolas.github.io/normalize.css/>

- Certains reset ont tendance à trop en faire. Les navigateurs ont des styles par défaut pour notamment permettre aux sites de s'afficher de façon convenable si le css a été désactivé notamment en terme d'accessibilité

Pratiquons ! - Découvrons flexbox, mise en page

Accueil

Galerie

Contact

- Ajout de la valeur "flex" pour la propriété "display" de la navigation
- Utilisation de la déclaration "margin-left : auto" pour ferrer à droite l'élément de navigation "contact"

Pratiquons ! - Découvrons flexbox, mise en page

Accueil

Galerie

Contact

- Utilisation des propriétés :

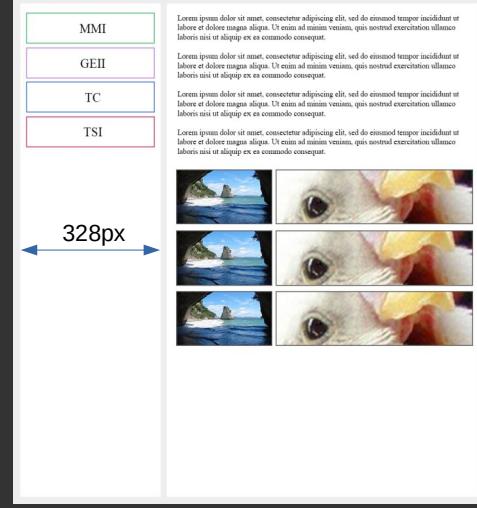
- padding

- 16 px en haut et en bas / 30 px à gauche et à droite

- border

- 1px d'épaisseur
 - style : solid
 - couleur : 5d5d5d

Pratiquons ! - Découvrons flexbox, mise en page



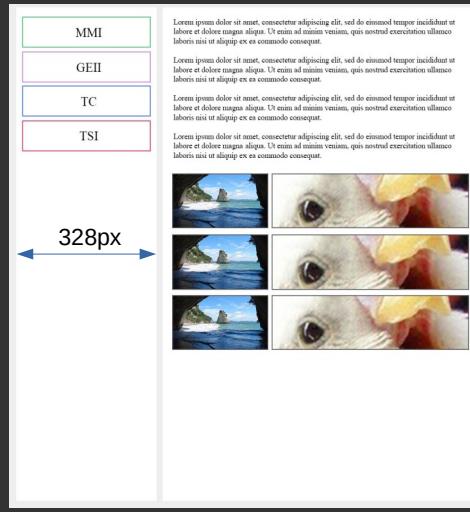
Sources :

- <https://www.alsacreations.com/tuto/lire/1493-css3-flexbox-layout-module.html>
- <https://developer.mozilla.org/fr/docs/Web/CSS/flex-shrink>
- <https://developer.mozilla.org/fr/docs/Web/CSS/flex-grow>
- https://developer.mozilla.org/fr/docs/Web/CSS/Flexible_Box_Layout/Concepts_de_base_flexbox

- Par défaut, les enfants des éléments flex occupent l'espace de leur contenu. Le navigateur s'arrange pour tout faire rentrer
- Mettre une taille fixe ne fonctionne pas comme attendu

Essayez de mettre une déclaration comme "width: 500px" au niveau de la balise <aside> vous verrez qu'elle n'occupe pas 500px. Ceci est dû au fait que flexbox permet, par défaut, à chaque enfant d'avoir un espace optimal quitte à compresser (shrink) les éléments trop volumineux

Pratiquons ! - Découvrons flexbox - flex-basis



- La propriété "flex-basis" permet de définir la taille d'un élément avant que l'espace soit distribué
- "flex-basis" écrase la valeur de la propriété "width"
- Accepte une longueur comme valeur
- Représente la troisième valeur de la propriété "flex"

Sources :

- <https://www.alsacreations.com/tuto/lire/1493-css3-flexbox-layout-module.html>
- <https://developer.mozilla.org/fr/docs/Web/CSS/flex-shrink>
- <https://developer.mozilla.org/fr/docs/Web/CSS/flex-grow>
- https://developer.mozilla.org/fr/docs/Web/CSS/Flexible_Box_Layout/Concepts_de_base_flexbox

Il faut comprendre que même si vous mettez des tailles précises sur vos éléments flex, il peut arriver qu'il y ait des incohérences au final, De fait, le navigateur choisit après coup l'affichage optimal

Pratiquons ! - Découvrons flexbox - flex

- Propriété à appliquer sur les enfants direct d'éléments flex
- Flex propriété raccourcie contenant :
 - flex-grow : capacité à occuper l'espace restant
 - flex-shrink : rétrécissement d'un élément si nécessaire
 - flex-basis : taille de l'élément avant répartition de l'espace
- "flex-grow" et "flex-shrink" acceptent un nombre décimal positif (si valeur incorrecte, valeur = 1)
- "flex-shrink: 0" indique que ma balise ne doit pas être rétrécie

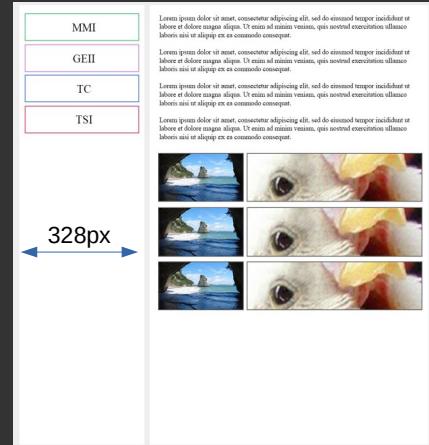
Sources :

- <https://www.alsacreations.com/tuto/lire/1493-css3-flexbox-layout-module.html>
- <https://developer.mozilla.org/fr/docs/Web/CSS/flex-shrink>
- <https://developer.mozilla.org/fr/docs/Web/CSS/flex-grow>
- https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Flexible_Box_Layout/Concepts_de_base_flexbox

La propriété **flex** est une propriété qui s'applique sur les enfants d'une balise "flexbox". C'est une propriété raccourcie, elle présente (dans cet ordre) :

- **flex-grow**
- **flex-shrink**
- **flex-basis**

Pratiquons ! - Découvrons flexbox, mise en page



- Avoir la navigation de gauche avec une largeur de 328px
- ```
.liste-cursus {
 flex-basis: 328px;
 flex-shrink: 0;
 flex-grow: 0;
}
```

### Sources :

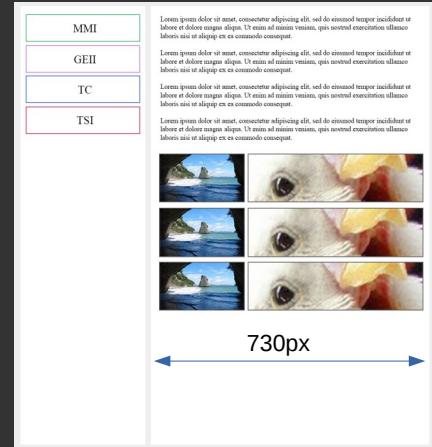
- <https://www.alsacreations.com/tuto/lire/1493-css3-flexbox-layout-module.html>
- <https://developer.mozilla.org/fr/docs/Web/CSS/flex-shrink>
- <https://developer.mozilla.org/fr/docs/Web/CSS/flex-grow>
- [https://developer.mozilla.org/fr/docs/Web/CSS/Flexible\\_Box\\_Layout/Concepts\\_de\\_base\\_flexbox](https://developer.mozilla.org/fr/docs/Web/CSS/Flexible_Box_Layout/Concepts_de_base_flexbox)

Dans notre classe "cursus", on veut que notre largeur soit exactement de 328px. Vu que nous voulons cette largeur quoi qu'il arrive, on la force avec la déclaration "flex-shrink : 0"

## Pratiquons ! - Découvrons flexbox, mise en page

- Et la partie de droite avec une largeur de 730 px

```
.description {
 flex: 0 0 730px;
}
```



### Sources :

- <https://www.alsacreations.com/tuto/lire/1493-css3-flexbox-layout-module.html>
- <https://developer.mozilla.org/fr/docs/Web/CSS/flex-shrink>
- <https://developer.mozilla.org/fr/docs/Web/CSS/flex-grow>
- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Concepts\\_de\\_base\\_flexbox](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Flexible_Box_Layout/Concepts_de_base_flexbox)

Dans notre classe "cursus", on veut que notre largeur soit exactement de 328px. Vu que nous voulons cette largeur quoi qu'il arrive, on la force avec la déclaration "flex-shrink : 0"

## Pratiquons ! - Découvrons flexbox, mise en page

The screenshot shows a web page with a dark background. At the top, there's a green header bar with the title "Pratiquons ! - Découvrons flexbox, mise en page". Below the header is a sidebar on the left containing four colored boxes labeled "MMI" (green), "GEII" (purple), "TC" (blue), and "TSI" (light blue). The main content area to the right of the sidebar contains three paragraphs of placeholder text (Lorem ipsum...) in a smaller font. The overall layout uses flexbox principles to organize the content.

MMI

GEII

TC

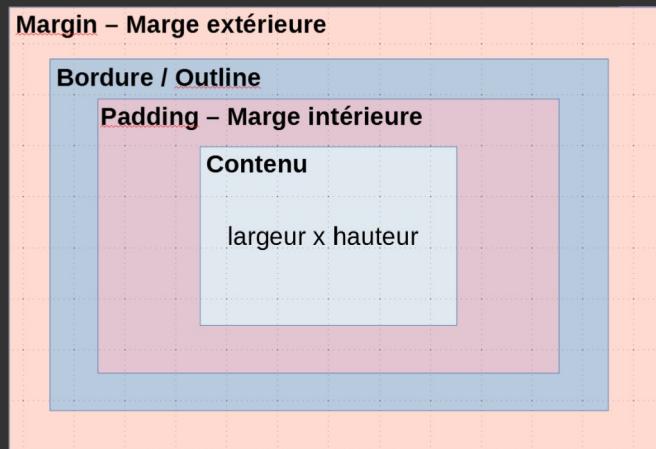
TSI

La mise en page ne ressemble pas vraiment à notre jpg malgré le bon dimensionnement des deux parties

Dans notre classe "cursus", on veut que notre largeur soit exactement de 328px. Vu que nous voulons cette largeur quoi qu'il arrive, on la force avec la déclaration "flex-shrink : 0"

## Calcul de dimension de boîte

### Rappel du modèle de boîte



- Les changements effectués avec la console du navigateur sont locaux, ce n'est pas avec ça que vous allez hacker un site, au mieux vous ferez une bonne blague à un utilisateur non averti

## Calcul de dimension de boite

- Trois composantes du modèle de boîte définissent la taille d'une balise :

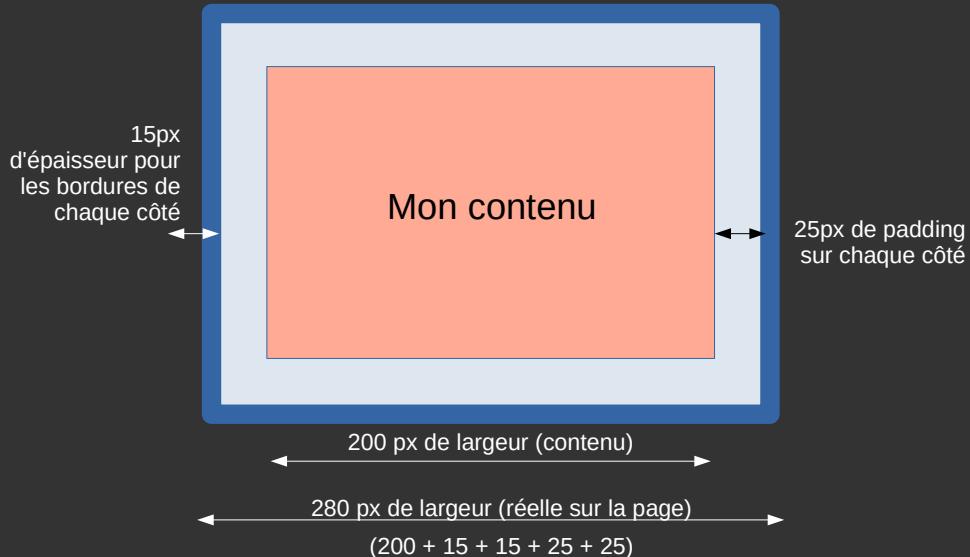
- contenu (width / height)
  - padding
  - border

- Admettons que j'écrive le code suivant :

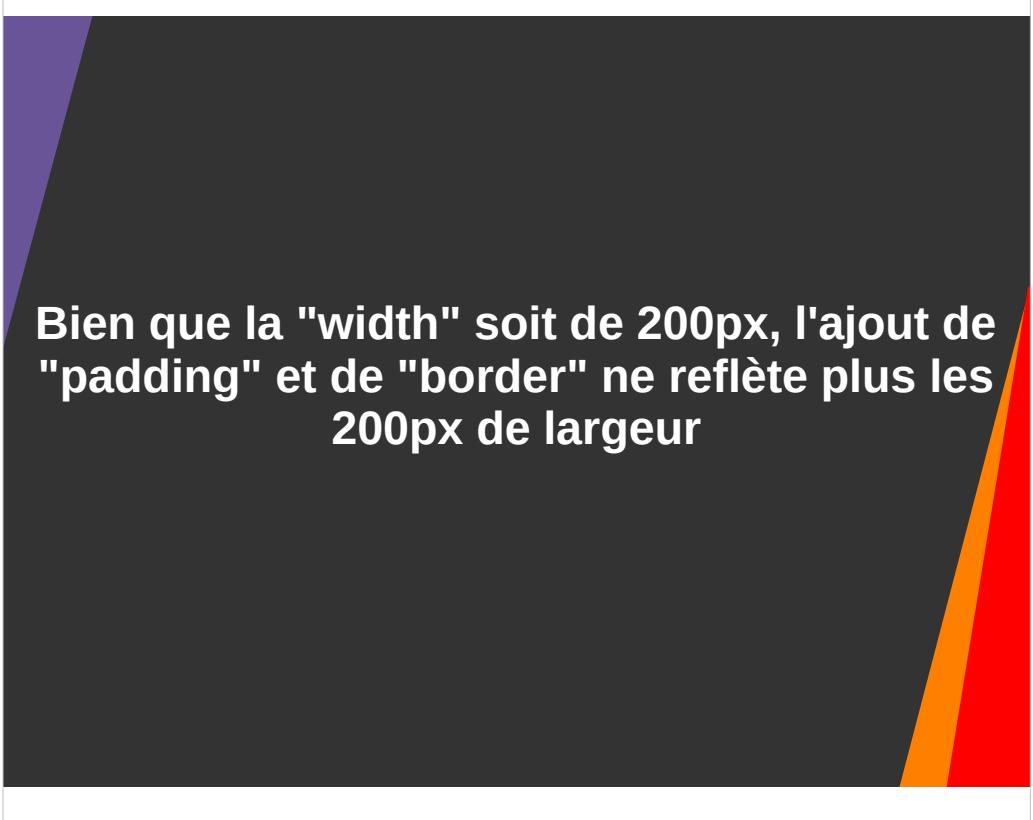
```
.ma-balise {
 width: 200px;
 border: 15px solid blue;
 padding: 25px;
}
```

- Les changements effectués avec la console du navigateur sont locaux, ce n'est pas avec ça que vous allez hacker un site, au mieux vous ferez une bonne blague à un utilisateur non averti

## Calcul de dimension de boite



- De ce fait, il faut prendre en compte la taille de la bordure, le padding et la dimension de mon contenu si on veut la vraie taille
- Si je veux que ma boite ait une largeur de 280px, il faut que je définisse sa largeur (width) à 200px, le padding et les bordures rajouteront les pixels manquant pour obtenir une largeur de 280px



**Bien que la "width" soit de 200px, l'ajout de "padding" et de "border" ne reflète plus les 200px de largeur**

Soustraction de l'épaisseur de la bordure et la marge intérieure pour obtenir la dimension permettant d'avoir une boîte ayant la taille voulue.



**Il faut donc prendre en compte vos éventuelles border et margin quand vous définissez la "width" et la "height" d'une balise, mais...**

Soustraction de l'épaisseur de la bordure et la marge intérieure pour obtenir la dimension permettant d'avoir une boite ayant la taille voulu.

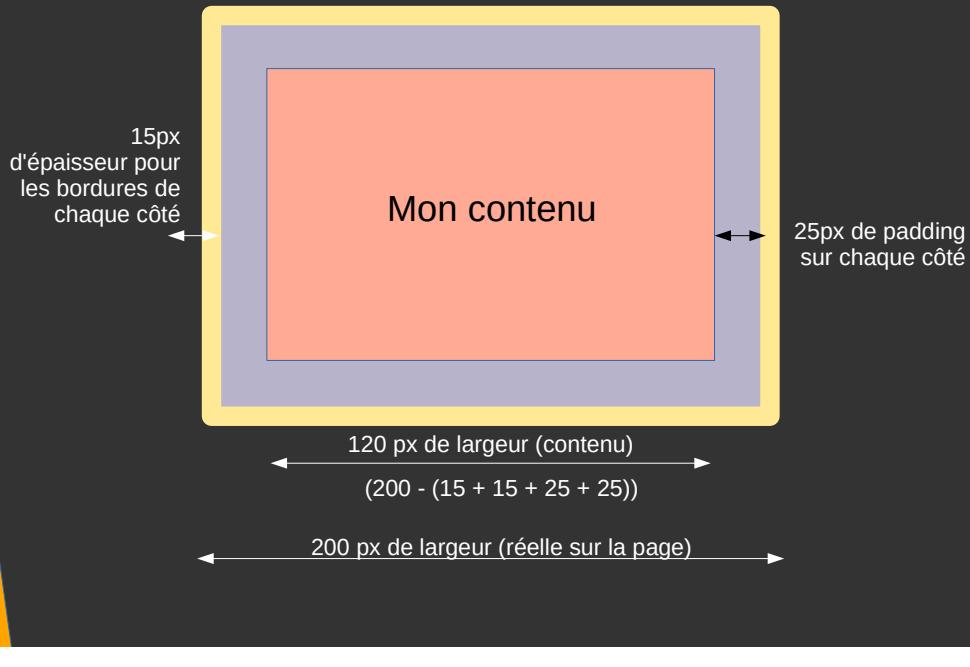
## Propriété box-sizing - Calcul de dimension de boite

- Permet de changer la façon dont est calculé les dimensions d'une boite
  - Moins de prise de tête
- Deux valeurs possibles
  - content-box : valeur par défaut
  - border-box : les propriétés "width" et "height" incluent la marge intérieure et la bordure

**Sources :**

- <http://fr.learnlayout.com/box-sizing.html>

## Propriété box-sizing - Calcul de dimension de boîte



- Avec la valeur "border-box", plus de prise de tête, le navigateur s'arrange pour qu'avec le padding et le border ma balise fasse la taille définie dans la propriété "width"

## Propriété box-sizing - Calcul de dimension de boite

```
*, *::before, *::after {
 box-sizing: border-box;
}
```

Seul cas où utiliser le sélecteur \* est légitime

Sources :

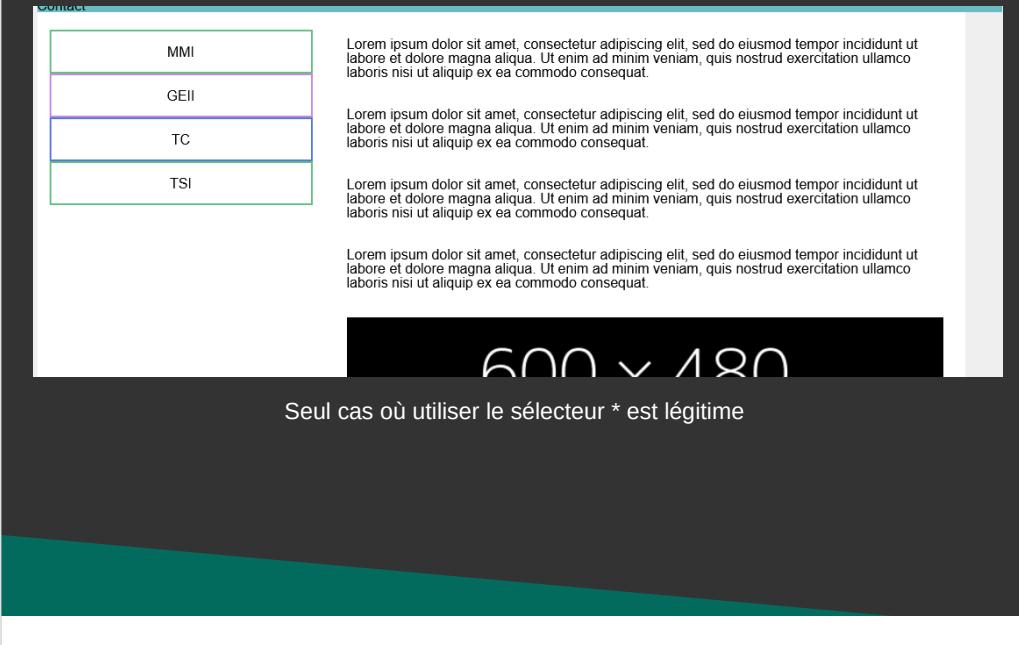
- <http://fr.learnlayout.com/box-sizing.html>

## Pratiquons ! - Découvrons flexbox, mise en page

- Rajoutons la déclaration suivante à notre fichier css
- `* , *::before, *::after {  
 box-sizing: border-box;  
}`

Dans notre classe "cursus", on veut que notre largeur soit exactement de 328px. Vu que nous voulons cette largeur quoi qu'il arrive, on la force avec la déclaration "flex-shrink : 0"

## Pratiquons ! - Découvrons flexbox, mise en page



Dans notre classe "cursus", on veut que notre largeur soit exactement de 328px. Vu que nous voulons cette largeur quoi qu'il arrive, on la force avec la déclaration "flex-shrink : 0"



**Ca avance, il nous faut maintenant les  
espaces**

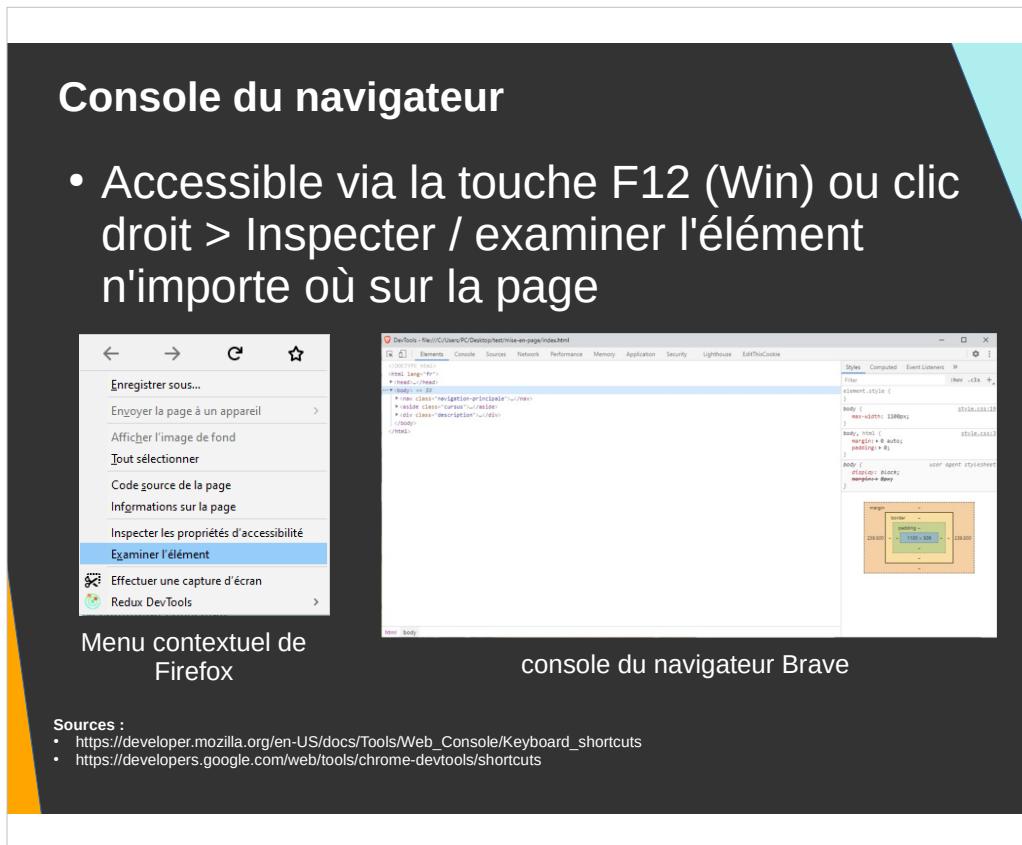
**Pour rappel la propriété "justify-content" permet,  
sur un parent flex, de définir le comportement de  
ses enfants sur l'axe principal (ici x)**

**Testons la propriété... sans toucher à VS Code**

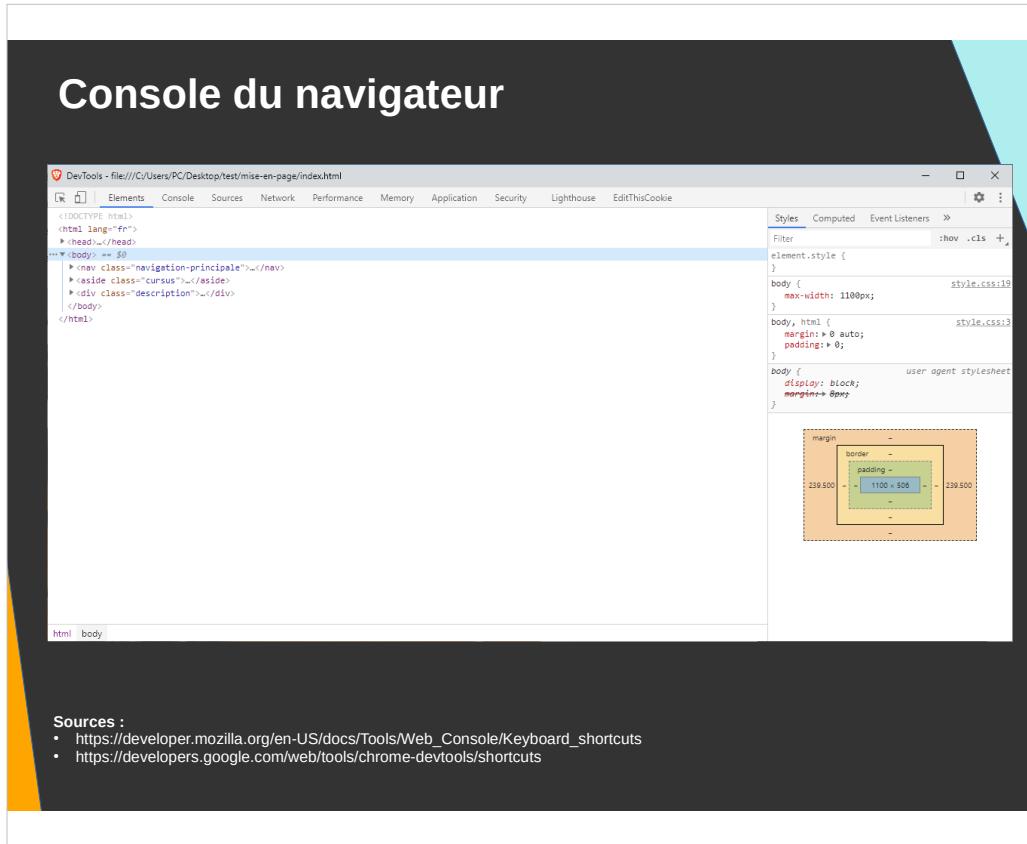
## Console du navigateur

- Outil natif des navigateurs modernes
  - Même sous mobile avec un ordinateur
- Permet d'effectuer rapidement des modifications / tests sur une page web
- Les changements sont locaux
- Indispensable pour debugger son site
- Permet "d'émuler" des appareils mobiles

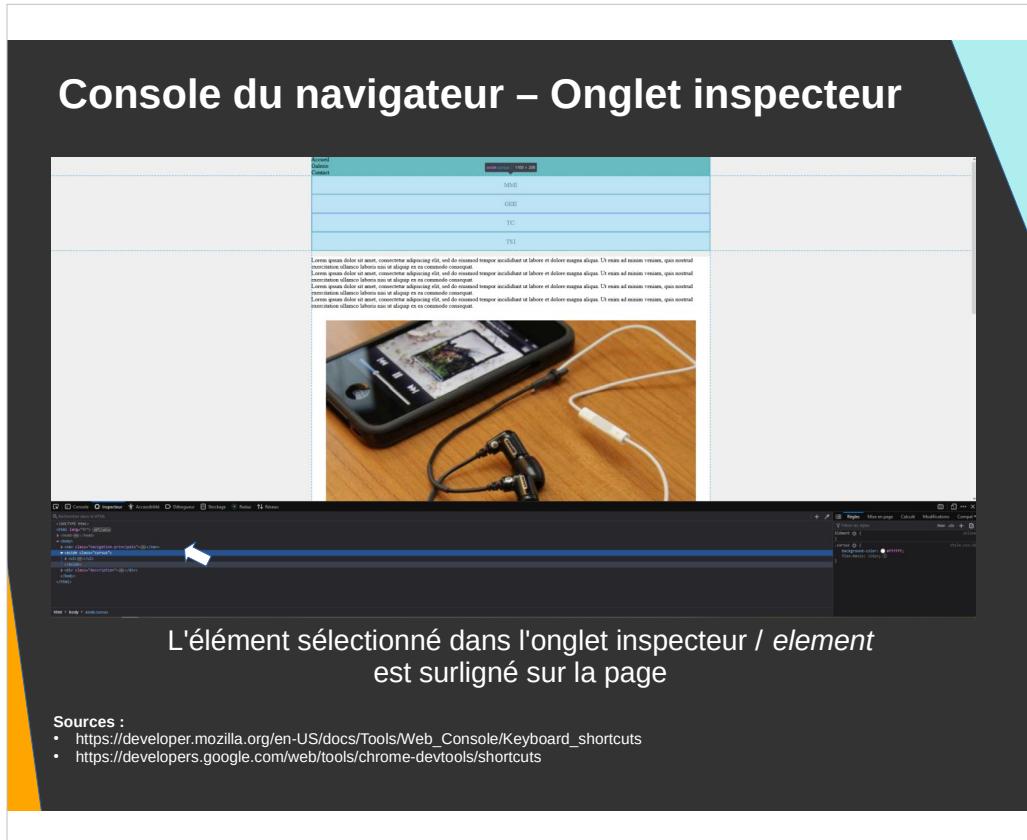
- Les changements effectués avec la console du navigateur sont locaux, ce n'est pas avec ça que vous allez hacker un site, au mieux vous ferez une bonne blague à un utilisateur non averti



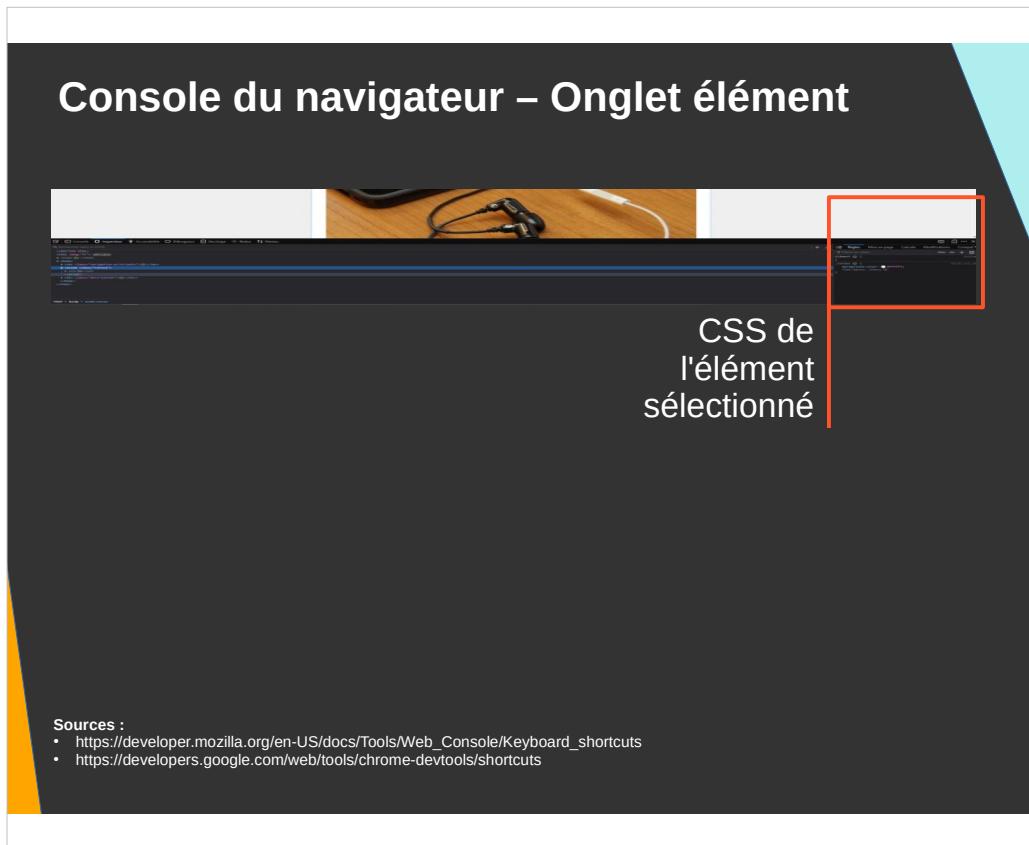
- Accessible via la touche F12 (Win) ou clic droit > Inspecter / examiner l'élément n'importe où sur la page
- Toutes les consoles web de navigateur fonctionnent plus ou moins de la même façon. Ce qui nous intéresse c'est l'onglet "inspecteur" (firefox) ou "Elements" (Chrome / Opera / Brave)
- Sans surprise IE possède un inspecteur très limiter, à éviter sauf si on ne peut pas faire autrement



- Durant ce cours, seul les trois premiers éléments nous intéressent (noms traduits par mes soins):
  - Sélectionner un élément
  - Barre d'outil mobile (à droite sous Firefox)
  - Elements



- L'onglet affiche en temps réel le contenu de la page web, contrairement au code source qui n'affiche le code qu'au premier chargement de la page



- On peut voir les sélecteurs css associées à l'élément sélectionné
- La partie "element" en haut représente l'élément unique, ajouter des déclarations css ici, c'est comme rajouter des déclaration directement sur la balise

## Pratiquons ! - Console web

Pré-requis :

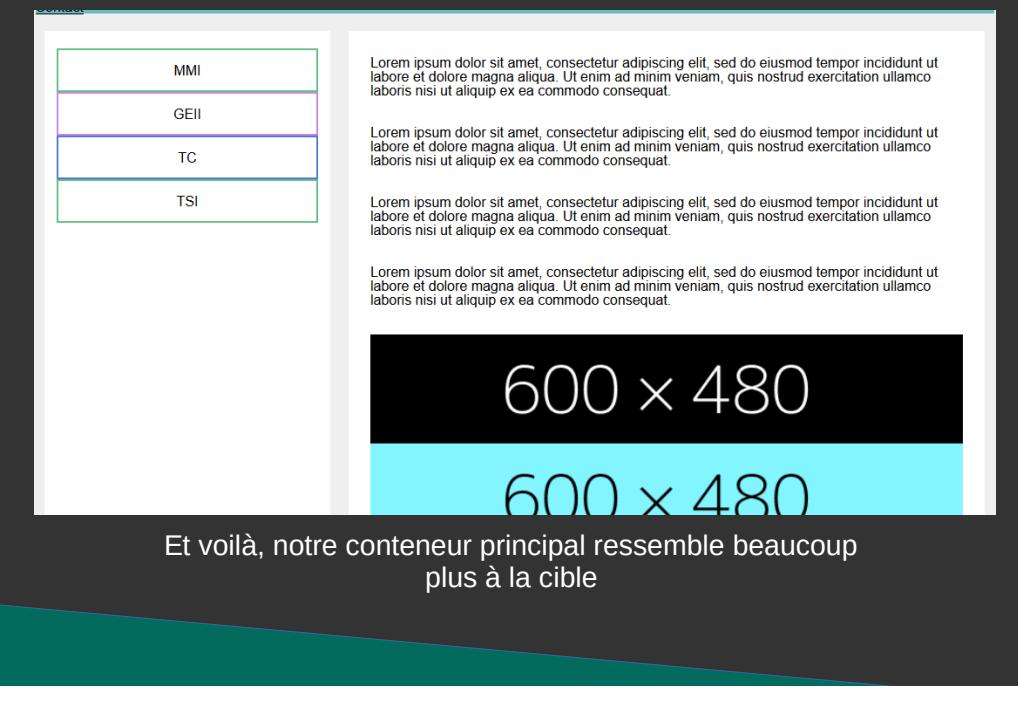
- Avoir le contenu de la ressource "ressource mise en page" sur ENT
- Chercher le conteneur principal de votre site
- Ajouter la propriété "justify-content" et jouer sur les valeurs possibles

Grâce à la console du navigateur, il est possible éditer la page aussi bien au niveau html et css

## Pratiquons ! - Console web

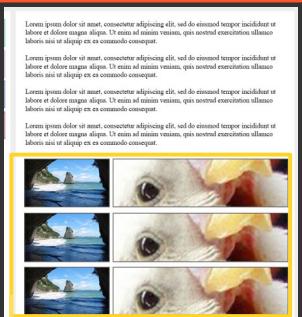
- Ajouter la déclaration  
**"justify-content: space-around;"**  
sur votre conteneur principal

## Pratiquons ! - Console web



## Pratiquons ! - Découvrons flexbox, mise en page

- Toujours penser la segmentation de vos balises en blocs qui vont ensemble



bloc "global" .description

### Sources :

- <https://www.alsacreations.com/tuto/lire/1493-css3-flexbox-layout-module.html>
- <https://developer.mozilla.org/fr/docs/Web/CSS/flex-shrink>
- <https://developer.mozilla.org/fr/docs/Web/CSS/flex-grow>
- [https://developer.mozilla.org/fr/docs/Web/CSS/Flexible\\_Box\\_Layout/Concepts\\_de\\_base\\_flexbox](https://developer.mozilla.org/fr/docs/Web/CSS/Flexible_Box_Layout/Concepts_de_base_flexbox)

Dans notre classe "cursus", on veut que notre largeur soit exactement de 328px. Vu que nous voulons cette largeur quoi qu'il arrive, on la force avec la déclaration "flex-shrink : 0"

## Pourcentages – Premiers pas vers le responsive design

- Utilisation de pourcentages pour gérer les tailles



largeur du parent  
largeur de  
l'élément cible

- L'utilisation de pourcentages permet à vos éléments de s'adapter à la taille des différents écrans

## Pourcentages – Premiers pas vers le responsive design

- Utilisation du produit en croix

$$\text{longueur en \%} = \frac{\text{longueur de l' élément cible} \times 100}{\text{longueur du parent}}$$

- La valeur obtenue servira de valeur pour la longueur que l'on souhaite.
- Il est préférable de mettre toutes les décimale (même s'il y en a 10) pour plus de précision

## Pourcentages – Premiers pas vers le responsive design

- Largeur des petites images : 224px (32.230211583%)
- Largeurs des grandes images: 462px (66.474882014%)

```
.image-complementaire:nth-child(odd) {
 flex: 1 0 32.230211583%;
}
```

} On cible les un élément sur deux en commençant par l'élément impair

```
.image-complementaire:nth-child(even) {
 flex: 1 0 66.474882014%;
}
```

} On cible les un élément sur deux en commençant par l'élément pair

### Sources :

• <https://css-tricks.com/examples/nth-child-tester/> - en

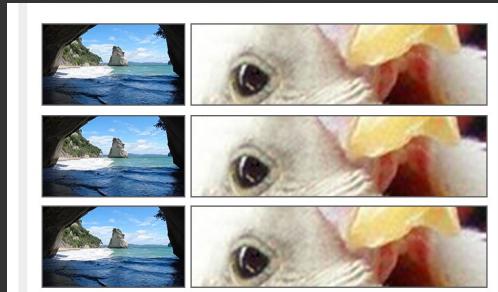
• <https://developer.mozilla.org/en-US/docs/Web/CSS/:nth-child>

- La pseudo-classe nth-child permet de cibler des éléments en fonction de leur position dans un groupe en fonction des éléments voisins
- Il est possible de faire la même chose dans nth-child, mais il faudra utiliser une autre classe qu'on mettra manuellement sur un élément sur deux. Plus facile, mais moins polyvalent que nth-child()
- Une pseudo-classe permet, ajouté à un sélecteur, d'appliquer des déclarations css sur un élément quand il est dans un état spécifique :  
<https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-classes>

## Pourcentages – Premiers pas vers le responsive design



Nos éléments ont bien la "bonne" taille, en revanche ils ne forment pas de colonnes comme sur le design



- Les images

## Pratiquons ! - Découvrons flexbox - flex-wrap

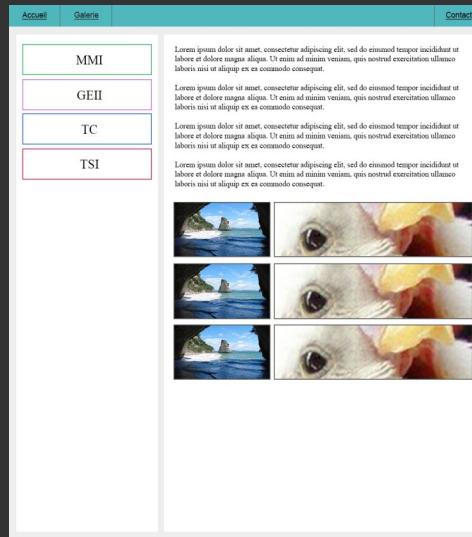


- La propriété `flex-wrap` indique si les enfants d'un élément flexbox doivent retourner à la ligne
- Possibilité de changer la direction de chaque ligne grâce à la valeur "`wrap-reverse`"

### Sources :

- <https://www.alsacreations.com/tuto/lire/1493-css3-flexbox-layout-module.html>
- <https://developer.mozilla.org/fr/docs/Web/CSS/flex-wrap>
- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Concepts\\_de\\_base\\_flexbox](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Flexible_Box_Layout/Concepts_de_base_flexbox)

## Pratiquons ! - Découvrons flexbox



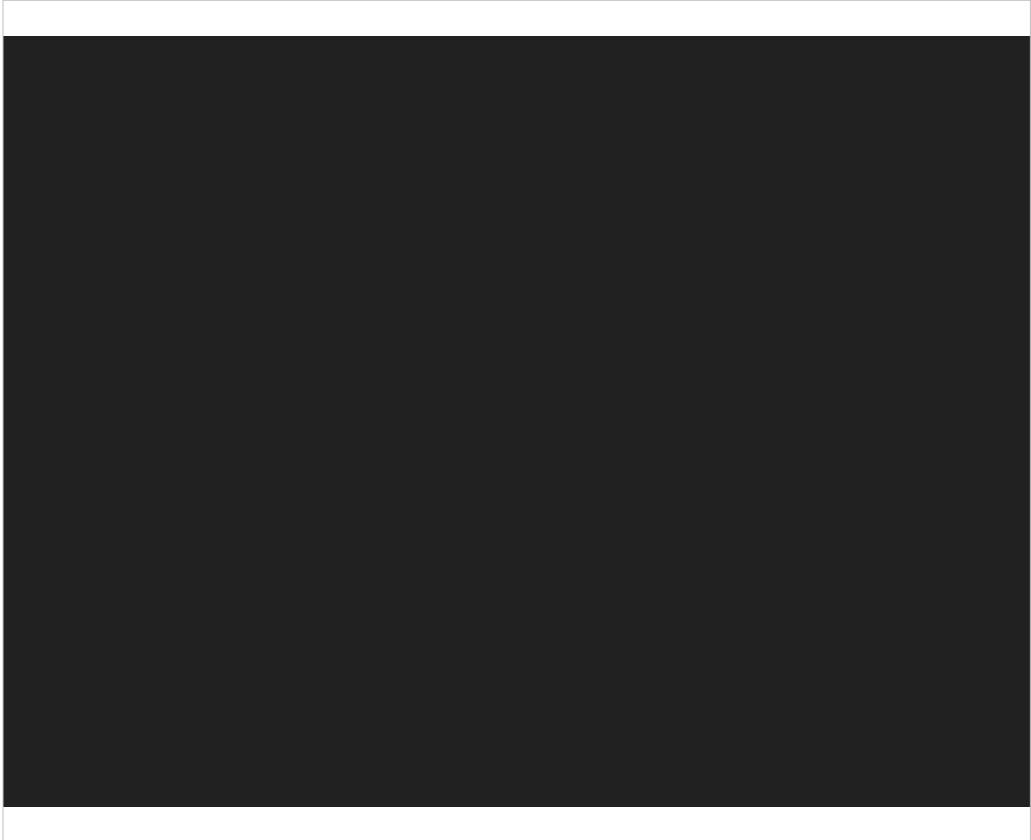
On devrait obtenir peu ou prou ce qu'on a en design

## Mise en page – Flexbox – En vrac

- La propriété "order", sur les enfants d'un élément flexbox, permet d'en définir manuellement l'ordre
- Un enfant d'un élément flexbox peut surcharger l'alignement de la propriété parente "align-items" avec la propriété "align-self"
- Flexbox possède une petite sœur bien plus puissante : "grid". Elle permet de gérer les éléments sur deux axes : horizontal et vertical

### Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/flex>
- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Grid\\_Layout](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Grid_Layout)
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Concepts\\_de\\_base\\_flexbox](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Flexible_Box_Layout/Concepts_de_base_flexbox)



**Questions ?**