

# Intégration Web

2020 - MMI 1 – TP#3 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

- La mise en page flexbox permettant de faire des mise en page plus complexe
- La propriété "position" pour modifier le comportement d'une balise dans le flux
- Le modèle de boite avec les propriétés :
  - width / height
  - margin / padding
  - border / outline

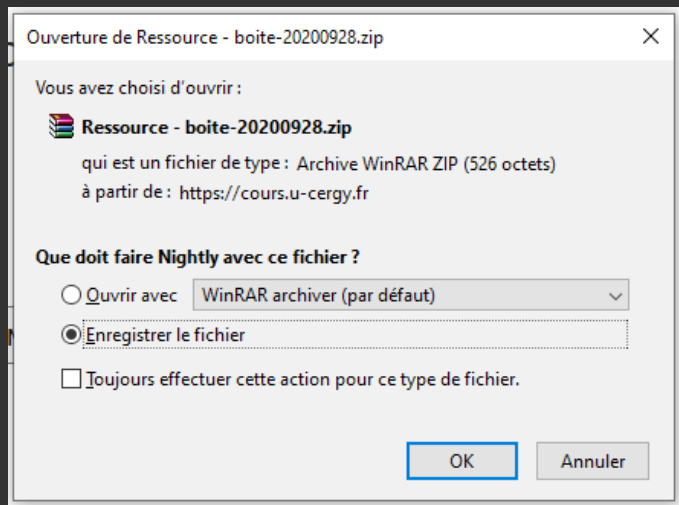
## Source :

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

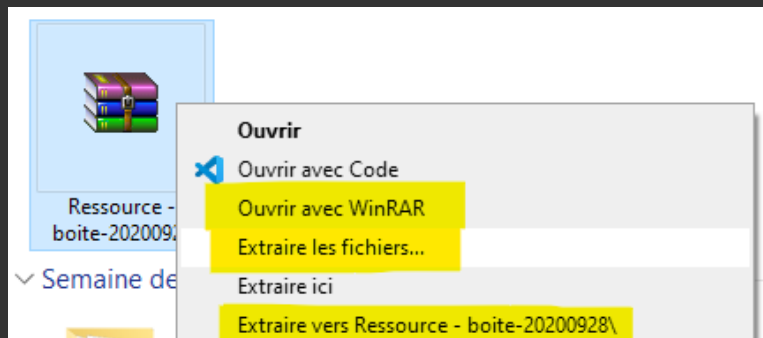
# Ressource du cours

**Travaux Pratiques #3 > "Ressource - Texte à décorer"  
sur ENT**

# Petit point sur les ressources de cours

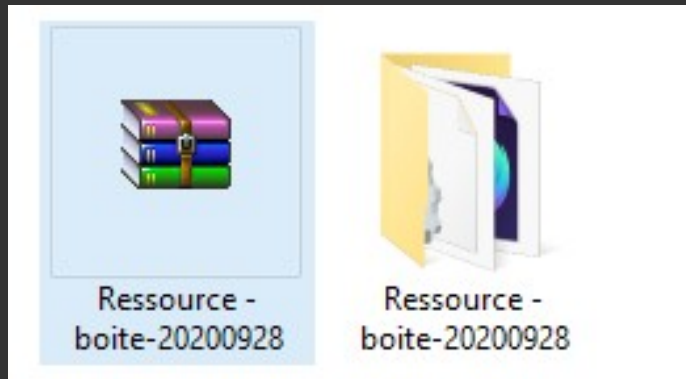


- Les ressources se présentent sous la forme d'une archive une fois téléchargées
- L'archive doit être impérativement extraite sinon vous ne serez pas capables de réaliser correctement les tp



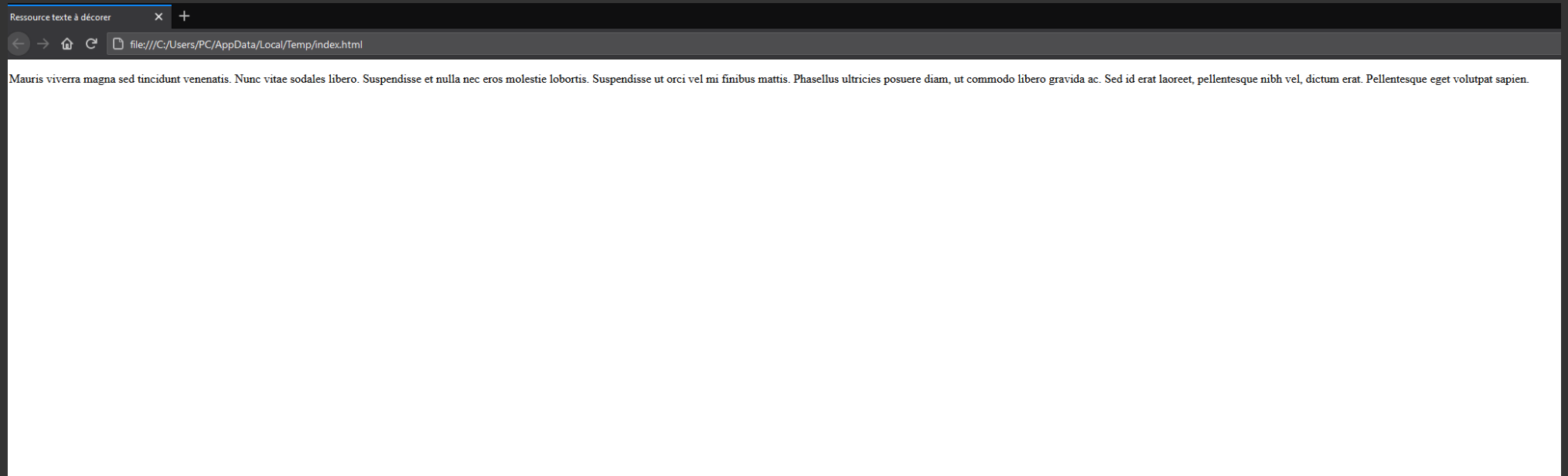
Menu contextuel sur l'archive (clic droit). Choisir un des choix surligné.

# Petit point sur les ressources de cours



- Une fois extraite, 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





# Formatage du texte

- Exclusivement fait grâce aux propriétés css
  - color : couleur du texte
  - font-weight : graisse du texte
  - font-family : police du texte
  - font-size : taille du texte
  - text-align : alignement du texte
  - text-decoration : "décore" le texte
  - [...]

## Sources :

- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Text](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Text)
- <https://developer.mozilla.org/fr/docs/Web/CSS/text-decoration>
- [https://developer.mozilla.org/fr/docs/Learn/CSS/Styling\\_text/initiation-mise-en-forme-du-texte](https://developer.mozilla.org/fr/docs/Learn/CSS/Styling_text/initiation-mise-en-forme-du-texte)
- <https://cssgradient.io/blog/css-gradient-text/> - texte en dégradé - en

# Formatage du texte - font-weight

- Permet de modifier la graisse d'un texte
- Toutes les polices d'écriture ne sont pas éligible à cette propriété (ex : Impact). Testable via un logiciel comme PhotoShop ou Photopea (gratuit)
- Accepte comme valeur :
  - Nombres compris en 100 et 900 définis
    - Changement de la règle avec CSS4 → Possibilité de mettre une valeur décimale
  - normal (400) (valeur par défaut) / bold (700)
  - lighter / bolder, valeurs relatives au parent

## Sources :

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

# Formatage du texte - font-family

- Permet de changer la police du texte
- Accepte comme valeur une liste de polices ordonnée par priorité, la police prioritaire étant la plus à gauche
  - Exemple : Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
- Par défaut, accès uniquement à des polices système
- Utilisation de la règle @font-face pour utiliser des polices non système

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/font-family>
- <https://www.cssfontstack.com/> - liste des polices "sûres"
- <https://developer.mozilla.org/fr/docs/Web/CSS/@font-face>

# Variables / Propriétés personnalisées CSS

- Éviter la redondance des **propriétés**
- Géré par tous les navigateurs modernes
- Syntaxe :
  - Définition de la variable
    - `--nom-de-variable: valeur-propriété;`
  - Utilisation
    - `propriété-css: var(--nom-de-variable);`

## Sources :

- [https://developer.mozilla.org/fr/docs/Web/CSS/Using\\_CSS\\_custom\\_properties](https://developer.mozilla.org/fr/docs/Web/CSS/Using_CSS_custom_properties)

# Variables / Propriétés personnalisées CSS

- Les variables possèdent un contexte (scope) défini par le sélecteur css
- Utilisation de la pseudo-classe ":root" pour rendre la variable globale

```
:root {  
  --couleur-arriere-plan: blue;  
}  
  
body {  
  --couleur-verte: rgb(0, 201, 50);  
  background-color: var(--couleur-arriere-plan, blue);  
  color: var(--couleur-verte);  
  border: 3px solid var(--couleur-verte);  
}
```

La variable est globale car définie dans ":root"

## Sources :

- [https://developer.mozilla.org/fr/docs/Web/CSS/Using\\_CSS\\_custom\\_properties](https://developer.mozilla.org/fr/docs/Web/CSS/Using_CSS_custom_properties)

# Formatage du texte - font-family

```
✓ :root {  
  --liste-polices: Helvetica, Arial, serif;  
}  
  
✓ .news {  
  font-family: var(--liste-polices);  
}  
  
/* ----- */  
  
✓ .news {  
  font-family: Helvetica, Arial, serif;  
}
```

Les deux codes font la même chose mais l'utilisation de variables rend le code plus maintenable

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/font-family>
- <https://www.cssfontstack.com/> - liste des polices "sûres"
- <https://developer.mozilla.org/fr/docs/Web/CSS/@font-face>

# font-family – Bonnes pratiques

- Éviter de mélanger trop de polices d'écriture au sein même d'un site  
**Deux ou trois** polices différentes maximum
- Pensez aux dyslexiques, préférer les polices simples à celles (trop) stylisées

## Sources :

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

# Formatage du texte - font-size

- Définit la taille du texte
- Accepte comme valeurs (toujours positives) :
  - Pourcentage
  - Longueur, unités possibles :
    - px
    - rem / em
    - ...

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/font-size>
- <https://developer.mozilla.org/fr/docs/Web/CSS/length>



# Point accessibilité – Taille de texte

- 14/16 px est considéré comme étant la taille recommandée pour un corps de texte sur le web
- Préférer les tailles de polices relatives comme le rem, le % ou l'em. Mais le rem reste moins prise de tête pour les calculs
- Ne pas oublier que certains utilisateurs peuvent zoomer ou même changer la taille de base de la font-size du navigateur

## Sources :

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

# Formatage du texte – rem / em / %

- Unités relatives
- rem se base sur l'élément racine (r pour root) de la page, souvent la font-size définit sur <html> avec une valeur de 16px
- em sera plus utilisé pour les marges notamment de la balise <p>
  - Se base sur la font-size du parent direct, de même pour l'unité pourcentage

## Sources :

- <https://24ways.org/2006/compose-to-a-vertical-rhythm/> - en

# Formatage du texte – rem

- 1 rem représente 100 % de la valeur de la font-size root
  - Balise <html>
  - Valeur du navigateur
- Exemple : si on définit "font-size: 16px" au niveau de la balise <html>, une balise <h1> qui a la déclaration "font-size : 1.5rem" a la valeur de 24px une fois convertie en pixels (1.5 x 16 = 24)
- Calcul pixel → rem :

$$\text{taille en rem} = \frac{\text{taille de l'élément (px)}}{\text{taille racine (px)}}$$

# Formatage du texte – em / %

- 1 em / % représente 100 % de la valeur de **la font-size du parent direct**
- Exemple :
  - Parent du parent : 20px
  - Parent : 0.5em (20 x 0.5 = 10px)
  - Élément cible : 2em (10 x 2 = 20px)
  - → Taille de police de l'élément cible : 20px
- Calcul pixel → em / % :

$$taille\ en\ em = \frac{taille\ police\ \acute{e}l\acute{e}ment\ (px)}{taille\ police\ parent\ (px)}$$

# Formatage du texte – text-align

- Permet de modifier l'alignement du texte
- Aligné à gauche par défaut (text-align: left;)
- Accepte les valeurs suivantes :
  - left, right, center, justify
- La valeur "justify" donne accès à une nouvelle propriété "text-justify"

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/text-align>
- <https://developer.mozilla.org/fr/docs/Web/CSS/text-justify>

# Formatage du texte – text-decoration

- Décore le texte
- Propriété raccourcie
  - text-decoration-line
  - text-decoration-color
  - text-decoration-style
- Permet par exemple de souligner ou barrer du texte avec une certaine couleur

## Source :

- <https://developer.mozilla.org/fr/docs/Web/CSS/text-decoration>
- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Text\\_Decoration](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Text_Decoration)

# Pratiquons ! - Décorons du texte

Pré-requis :

- Avoir le contenu de la ressource "Ressource texte à décorer" sur ENT
- Changer l'alignement des paragraphes (différent pour chaque)
- Changer la "font-family" d'une partie du texte.  
Exemples de polices sûres :
  - Arial, Impact, "Times New Roman", Verdana, Tahoma (voir lien en-dessous pour plus de polices)
  - Ne pas oublier les "polices de rechange"

Source :

- <https://www.cssfontstack.com/> - liste des polices "sûres"

# Pratiquons ! - Décorons du texte

- Au sein du même paragraphe :
  - Souligner du texte et le rendre gras
  - Augmenter la font-size
  - Diminuer la font-size
  - Ajouter une couleur d'arrière-plan
  - Utiliser une propriété liée à la décoration du texte non vue en cours
- Les décorations citées doivent être appliquées sur un fragment de texte différent
  - Une ligne = un fragment

## Source :

- <https://developer.mozilla.org/fr/docs/Web/CSS/text-decoration>
- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Text\\_Decoration](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Text_Decoration)



# Quiz

## "Test - Texte" sur ENT

Conditions du test :

- ~9 minutes
- Une tentative

# Ressource du cours

**Travaux Pratiques #3 > "Ressource – Site web" sur  
ENT**

# Site web

- Les pages peuvent être statiques ou dynamiques
- Ensemble infini de pages web liées par des liens hypertextes
- Les pages sur le même domaine sont considérées comme faisant partie du même site web.
- Si mon site a un lien vers le site de duckduckgo ça ne veut pas dire que mon site fait partie de duckduckgo et vice-versa

## Sources :

- <https://24ways.org/2006/compose-to-a-vertical-rhythm/> - anglais

# Balise <a>

- a pour "anchor" ou "ancree" en français
- Permet de définir un hyperlien dans la page ou une autre page sur le web
- Utilise l'attribut "href" pour définir l'hypertexte

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/a>

# Balise <a>

- Historiquement les liens s'affichent soulignés
  - Préférable de réserver le soulignement aux liens
- Par défaut un lien s'ouvre dans le même onglet
  - Utilisation de l'attribut "target" pour changer le comportement
    - target="\_blank" : Ouvrir un lien dans un nouvel onglet
- Grâce au système d'imbrication html, une image peut pointer vers une ressource

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/a>
- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a#attr-target>

# Balise <a> - Exemple de code

```
<a href="https://www.u-cergy.fr">
```

Site de Cergy Paris Université

```
</a>
```

Mon lien s'ouvrira dans le même onglet

```
<a href="https://www.u-cergy.fr"
```

```
target="_blank">
```

Site de Cergy Paris Université

```
</a>
```

Mon lien s'ouvrira dans un nouvel onglet

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/a>

# Balise <a> - ancre sur la même page

- Permet d'atteindre un élément dans la page grâce à son paramètre "id"
- Un élément sur la page doit avoir un id avec une valeur
- Une balise <a> doit avoir comme valeur pour l'attribut "href" le même id précédé du signe croisillon (#)

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/a>

# Pratiquons ! - Lien interne

Pré-requis :

- Avoir le contenu de la ressource "Ressource – Site web" sur ENT
- Ajouter l'attribut id sur une balise vers le bas de la page
- Ajouter en haut de page une balise <a> avec l'attribut "href" ayant la valeur "#" + "le-nom-de-votre-id"



# Pratiquons ! - Lien interne

[Hello test](#) Cras molestie diam eu metus malesuada facilisis. Curabitur nec consequat purus. Mauris gravida volutpat tortor, at vestibulum metus egestas sed. Ut non felis in tortor iaculis hendrerit. Suspendisse ornare tincidunt odio, et faucibus eros volutpat a.

Avant clic

[Hello test](#) Cras molestie diam eu metus malesuada facilisis. Curabitur nec consequat purus. Mauris gravida volutpat tortor, at vestibulum metus egestas sed. Ut non felis in tortor iaculis hendrerit. Suspendisse ornare tincidunt odio, et faucibus eros volutpat a.

Après clic

# Pseudo-classe

- Permet d'appliquer des déclarations css sur un élément dans un état spécifique
- Syntaxe :
  - un sélecteur html + ":" + le nom de la pseudo classe
  - exemple : `sélecteur:pseudo-classe {  
border: 2px solid blue;  
}`

## Sources :

- <https://www.creativejuiz.fr/blog/css-css3/difference-entre-pseudo-element-et-pseudo-classe>
- <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-classes>

# Pseudo-classe

- Vous ne pouvez pas inventer vos propres pseudo-classes
- Vous ne pouvez pas associer plusieurs fois la même pseudo-classe

## Sources :

- <https://www.creativejuiz.fr/blog/css-css3/difference-entre-pseudo-element-et-pseudo-classe>
- <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-classes>

# Pseudo-classe – <a> dans tous ses états

- La balise <a> est souvent associée aux pseudo-classes suivantes :
  - :link : désigne les liens non visités
  - :visited : désigne les liens visités
  - :hover : désigne les liens au survol
  - :active : désigne le moment entre le clic et le relâchement d'un lien
- Pour éviter tout problème de comportement, il est préférable de les écrire dans l'ordre suivant :
  - :link — :visited — :hover — :active.

## Sources :

- <https://www.creativejuiz.fr/blog/css-css3/difference-entre-pseudo-element-et-pseudo-classe>
- <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-classes>

# Pratiquons ! - Lien interne

Pré-requis :

- Avoir le contenu de la ressource "Ressource – Site web" sur ENT avec une balise `<a>` dans le code
- Rajouter au sélecteur a les pseudo-classes (:link, :visited, :hover, :active)
  - `a:pseudo-classe {`  
    `color: red;`  
    `}`
- Mettre un style différent (color, text-decoration...) au choix pour chacune de ces pseudo-classes

# Pseudo-classe – <a> dans tous ses états

- Prend également en compte la pseudo-classe "focus"
  - N'est pas influencé par l'ordre vu précédemment
  - Visible uniquement au "focus" clavier du lien
    - Déplacement sur le site avec le clavier

## Sources :

- <https://www.creativejuiz.fr/blog/css-css3/difference-entre-pseudo-element-et-pseudo-classe>
- <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-classes>

# Balise <a> - lien externe

- Utilisation de l'attribut "href"
- Permet de faire des liens absolus ou relatifs
  - Vers page web, fichiers (jpg, pdf, doc)...
- Il est préférable d'ouvrir un lien dans le même onglet
  - Attribut "target" permet de changer ce comportement grâce à la valeur "\_blank"
- Indiquer si on ouvre une nouvelle page ou lien
- Éviter les cta (call to action) trop obscurs comme "cliquer ici", il est vaut mieux mettre l'action que ça va effectuer. Exemple "Accéder à ENT"

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/a>

# Pratiquons ! - Lien externe

Pré-requis :

- Avoir le contenu de la ressource "Ressource – Site web" sur ENT avec une balise `<a>` dans le code
- Faire pointer votre lien (ou un autre) vers l'autre page présente dans la ressource "description.html"
- Dans le fichier "description.html" faire pointer un lien vers la page "index.html"
- Dans le fichier "description.html" faire pointer une image vers une image



# Balise <a> - attribut href

- Permet de faire des liens vers des ancres ou des pages distantes
- Schéma url : permet d'effectuer des actions spécifiques
  - mailto: envoyer une adresse mail
  - tel: composer un numéro de téléphone

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/a>

# Point accessibilité – Les liens

- Les liens ne doivent pas être utilisés pour être des pseudo-boutons, la balise `<button>` est là pour ça
- Éviter les liens trop proches pour limiter les mauvaises manipulation
- Sur mobile, on cible avec les doigts, périphérique plus gros, il faut donc s'arranger pour que les éléments cliquables soient plus gros sur appareils mobiles et/ou plus espacés

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/a#Accessibilit%C3%A9>

# Point accessibilité – Les liens

- Indiquer les liens qui vont vers des sites de langue étrangère
  - Utilisation de l'attribut "title" est suffisant
- Indiquer le libellé du lien
  - Éviter des liens du type "cliquez ici" préférer "accéder à mon site"
  - De même éviter les liens qui sont des liens
  - La balise "title" est une alternative viable

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/a#Accessibilit%C3%A9>

# Quiz

## "Test - Liens" sur ENT

Conditions du test :

- ~10 minutes
- Une tentative

# Pseudo-classe - :hover

- Permet d'appliquer des déclarations css sur un élément au survol
- Applicable sur n'importe quelle balise
- Idéal pour effectuer des transitions d'élément(s)
- Peut de cibler plusieurs balises en même temps grâce au système d'imbrications
- Ne fonctionne pas sur mobile

## Sources :

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

# Pseudo-classe - :hover

- Jouer avec le principe des imbrications de balises html et règles css
- Indiquer le comportement des enfants lors du survol (:hover) du parent

```
.ma-balise:hover p {  
    color: red;  
}
```

```
<div class="ma-balise">  
    <p>Mon texte</p>  
</div>
```

Ici, on cible les <p> quand <div class="ma-balise"> est survolée

## Sources :

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

# Ressource du cours

Travaux Pratiques #3 > "Ressource – hover" sur ENT

# Pratiquons ! - :hover sur sur balise

Pré-requis :

- Avoir le contenu de la ressource "Ressource – hover" sur ENT
- Ajouter un nouveau sélecteur ".conteneur-survol:hover" dans le fichier css
- Changer la couleur du span contenue dans la balise au survol de la classe "conteneur-survol"



**Ça fonctionne... mais ça manque de dynamisme**

# Transitions css

- Permet de faire une transition entre deux états (valeur de propriétés)
- Propriété "transition"
- Souvent lancées au survol (hover), au focus ou autres interactions (nécessite javascript)
- Fonctionne mieux si les transitions sont sur le sélecteur et non le sélecteur + pseudo-classe
- Animations "simples" comparé aux animations
- Rajoute du "dynamisme"

## Sources :

- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Transitions/Utiliser\\_transitions\\_CSS](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Transitions/Utiliser_transitions_CSS)
- <https://developer.mozilla.org/en-US/docs/Web/CSS/transition>
- [https://developer.mozilla.org/fr/docs/Web/CSS/Liste\\_propri%C3%A9t%C3%A9s\\_CSS\\_anim%C3%A9es](https://developer.mozilla.org/fr/docs/Web/CSS/Liste_propri%C3%A9t%C3%A9s_CSS_anim%C3%A9es)

# Transitions css

- Propriété raccourcie :
  - transition-property : propriétés qui vont changer
  - transition-duration : durées des transitions
  - transition-timing-function : description de la courbe d'évolution des valeurs intermédiaires
  - transition-delay : délai avant le lancement de la transition
- Si les valeurs ne sont pas définies, celles par défaut seront utilisées

## Sources :

- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Transitions/Utiliser\\_transitions\\_CSS](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Transitions/Utiliser_transitions_CSS)
- <https://developer.mozilla.org/en-US/docs/Web/CSS/transition>
- [https://developer.mozilla.org/fr/docs/Web/CSS/Liste\\_propri%C3%A9t%C3%A9s\\_CSS\\_anim%C3%A9es](https://developer.mozilla.org/fr/docs/Web/CSS/Liste_propri%C3%A9t%C3%A9s_CSS_anim%C3%A9es)

# Transitions css

**transition:** **margin-right** **2s** **linear** **.5s**;

-duration      -delay

-property      -timing-function

## Sources :

- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Transitions/Utiliser\\_transitions\\_CSS](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Transitions/Utiliser_transitions_CSS)
- <https://developer.mozilla.org/en-US/docs/Web/CSS/transition>
- [https://developer.mozilla.org/fr/docs/Web/CSS/Liste\\_propri%C3%A9t%C3%A9s\\_CSS\\_anim%C3%A9es](https://developer.mozilla.org/fr/docs/Web/CSS/Liste_propri%C3%A9t%C3%A9s_CSS_anim%C3%A9es)

# Pratiquons ! - :hover sur sur balise

Pré-requis :

- Avoir le contenu de la ressource "Ressource – hover" sur ENT avec une pseudo-classe "hover" sur le sélecteur "conteneur-survol"
- Ajoutons une transition sur le sélecteur ".conteneur-survol:hover"

```
.conteneur-survol:hover .titre {  
  [...]  
  transition-property: color;  
  transition-duration: durée choisie;  
  transition-timing-function: linear;  
  transition-delay: durée choisie;  
}
```

# Pratiquons ! - :hover sur sur balise

- Corrigeons notre code :
  - Retirer la transition dans le sélecteur ".conteneur-survol:hover"
  - Mettre la transition sur la classe ".titre"

```
.titre {  
  [...]  
  transition-property: color;  
  transition-duration: durée choisie;  
  transition-timing-function: linear;  
  transition-delay: durée choisie;  
}
```

Maintenant la transition est présente quand on entre et sort de l'élément ".conteneur-survol"

# Transitions css

- Presque toutes les propriétés css sont animables
- Utilisation de la propriété "transform" pour effectuer des transitions plus élaborées
- Un élément modifié par la propriété transform garde sa place dans le flux. Elle ne fait donc pas bouger les éléments juxtaposés

## Sources :

- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Transitions/Utiliser\\_transitions\\_CSS](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Transitions/Utiliser_transitions_CSS)
- <https://developer.mozilla.org/en-US/docs/Web/CSS/transition>
- [https://developer.mozilla.org/fr/docs/Web/CSS/Liste\\_propri%C3%A9t%C3%A9s\\_CSS\\_anim%C3%A9es](https://developer.mozilla.org/fr/docs/Web/CSS/Liste_propri%C3%A9t%C3%A9s_CSS_anim%C3%A9es)
- <https://www.alsacreations.com/tuto/lire/873-transitions-css3-animations.html>

# Pratiquons ! - Essayons les transitions

Pré-requis :

- Avoir le contenu de la ressource "Ressource – transform" sur ENT le contenu du dossier "exercice" servira de base de travail
- Premières transitions



Objectif



Base



# Pratiquons ! - Essayons les transitions

Première transition en détails :

- Opacité de l'image de fond
- Apparition du sous-texte
- Déplacement du sous-texte, du titre et l'image



État initial



État final

# Pratiquons ! - Essayons les transitions

- Appliquons les sélecteurs css les plus simples : l'opacité et sa transition sur l'image

```
.element-conteneur.transition-1 img {  
    opacity: 0.7;  
    transition: opacity 0.35s;  
}
```

```
.element-conteneur.transition-1:hover img {  
    opacity: 1;  
}
```

# Pratiquons ! - Essayons les transitions

- Vous devriez obtenir ceci



# Transitions css – Propriété transform

- Propriété CSS3
- Permet d'appliquer des transformations sur un élément comme la translation ou la rotation
- Accepte comme valeurs des fonctions définies
- Possibilité d'appliquer plusieurs fonctions en même temps espacées
- N'utilise pas l'accélération matérielle (GPU)

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transform>
- <https://www.alsacreations.com/article/lire/1418-css3-transformations-2d.html>

# Propriété transform – Fonction translate

- Permet de déplacer un élément sur l'axe x et y
- Accepte comme valeur une longueur
- Possibilité de se déplacer sur l'axe z avec la fonction translate3d – Accepte trois paramètres
- Propriété raccourcie
  - translateX(), translateY(), translateZ()
- translate() n'utilise pas l'accélération matérielle contrairement à translate3d()

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transform-function/translate3d>
- <https://developer.mozilla.org/fr/docs/Web/CSS/transform-function/translate>
- <https://developer.mozilla.org/fr/docs/Web/CSS/transform-function#Translations>
- <https://www.alsacreations.com/astuce/lire/1565-acceleration-materielle-au-service-de-vos-animations-css.html>

# Pratiquons ! - Essayons les transitions

- Toujours sur l'image appliquons la translation

```
.element-conteneur.transition-1 img {  
  opacity: 0.7;  
  transition: opacity 0.35s, transform 0.35s;  
  transform: translate3d(-40px, 0, 0);  
}
```

```
.element-conteneur.transition-1: hover img {  
  opacity: 1;  
  transform: translate3d(0, 0, 0);  
}
```

# Pratiquons ! - Essayons les transitions

- Vous devriez obtenir ceci



# Pratiquons ! - Essayons les transitions

- Déplaçons le titre

```
.element-conteneur.transition-1 h2 {  
  [...]   
  transform: translate3d(0, 40px, 0);  
}
```

```
.element-conteneur.transition-1:hover h2 {  
  transform: translate3d(0, 0, 0);  
}
```



# Pratiquons ! - Essayons les transitions

- Puis le sous-texte

```
.element-conteneur.transition-1 p {  
  [...]  
  transition: opacity 0.2s, transform 0.35s;  
  transform: translate3d(0, 60px, 0);  
}
```

```
.element-conteneur.transition-1:hover p {  
  transform: translate3d(0, 0, 0);  
  opacity: 1;  
}
```

# Pratiquons ! - Essayons les transitions

- Et normalement, on devrait obtenir nos transitions cibles



# Transitions css – Bonnes pratiques

- Préférer `translate` / `translate3d` au lieu des propriétés `top`, `margin`, `width`... pour des questions de performances
- Inutile de rendre accessible une animation, en revanche, les informations importantes n'ont pas à être cachées par une animation
- Faire attention à la vitesse des animations, trop lentes ou rapides, elles peuvent nuire à l'expérience utilisateur

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transform-function#Translations>
- <https://www.alsacreations.com/astuce/lire/1565-acceleration-materielle-au-service-de-vos-animations-css.html>

# Transitions css – Bonnes pratiques

- Éviter les zones de survol trop petites surtout si elles affichent des données critiques
- Ne pas oublier qu'une animation joue notamment sur la persistance rétinienne

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transform-function#Translations>
- <https://www.alsacreations.com/astuce/lire/1565-acceleration-materielle-au-service-de-vos-animations-css.html>

# Pratiquons ! - Essayons les transitions

Deuxième transition en détails :

- Apparition d'une barre blanche
- Déplacement du titre
- Mise à l'échelle et rotation de l'image de fond



État initial



État final

# Pratiquons ! - Essayons les transitions

- Déplaçons la barre blanche

```
.element-conteneur.transition-2 figcaption::before {  
    compléter  
}
```

```
.element-conteneur.transition-2:hover figcaption::before {  
    compléter  
}
```

# Pseudo-éléments

- Permet de cibler une partie du contenu sans pour autant ajouter de nouvelles balises
- N'existe pas pour les lecteurs d'écran → Ne pas mettre de données critiques
- Syntaxe :
  - un sélecteur html + "::" + le nom de la pseudo classe
  - exemple : **mon-sélecteur::pseudo-element** {  
    **color: blue;**  
}

## Sources :

- <https://www.creativejuiz.fr/blog/css-css3/difference-entre-pseudo-element-et-pseudo-classe>
- <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-%C3%A9l%C3%A9ments>
- <https://accessibleweb.com/question-answer/how-is-css-pseudo-content-treated-by-screen-readers/> - en
- <https://developer.mozilla.org/fr/docs/Web/CSS/::first-line>

# Pseudo-éléments

- Vous ne pouvez pas associer plusieurs fois le même pseudo-élément à une balise
- Ne s'affiche pas dans le code source
- `::before` / `::after` sont certainement ce que vous utiliserez le plus

## Sources :

- <https://www.creativejuiz.fr/blog/css-css3/difference-entre-pseudo-element-et-pseudo-classe>
- <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-%C3%A9l%C3%A9ments>
- <https://accessibleweb.com/question-answer/how-is-css-pseudo-content-treated-by-screen-readers/> - en
- <https://developer.mozilla.org/fr/docs/Web/CSS/::first-line>



# Pseudo-éléments - ::before /::after

- Permettent d'ajouter des éléments avant ou après un contenu
- **Nécessite la propriété "content"** pour s'afficher. Contenu pouvant être :
  - une chaîne de caractères, une image (qui ne peut pas être redimensionnée), rien (une chaîne vide), un compteur

## Sources :

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

# Pseudo-éléments - ::before /::after

- Acceptent toutes les déclarations css
- Ne fonctionne pas sur les éléments remplacés comme `<img>` ou `<br>`
- Le contenu dans l'attribut "content" n'est pas sélectionnable

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/::after>
- <https://developer.mozilla.org/fr/docs/Web/CSS/::before>
- <https://a.singlediv.com/> - Exemple de ce qu'on peut faire avec ces pseudo-éléments

# Pratiquons ! - Essayons les transitions

- Déplaçons la barre blanche

```
.element-conteneur.transition-2 figcaption::before {  
    compléter  
}
```

```
.element-conteneur.transition-2:hover figcaption::before {  
    compléter  
}
```

# Pratiquons ! - Essayons les transitions

- Déplaçons le titre
  - Le `<h2>` part d'où il est, et descend de 40px sur l'axe y

```
.element-conteneur.transition-2 h2 {  
    compléter  
}
```

```
.element-conteneur.transition-2:hover h2 {  
    compléter  
}
```

# Pratiquons ! - Essayons les transitions

- Sur l'image appliquons ses transitions
- Pour rappel, il nous faut changer :
  - échelle (scale) et rotation (rotateZ)

```
.element-conteneur.transition-2 img {  
    compléter  
}
```

```
.element-conteneur.transition-2: hover img {  
    compléter  
}
```

# Transitions css – Fonction scale()

- Permet de changer la mise à l'échelle d'un élément
- Accepte jusqu'à deux valeurs (axe x, axe y)
- Ces valeurs doivent être des nombres décimaux négatifs ou positifs
- `scale(1)` = élément garde son échelle de base (sur les deux axes)
- Fonction raccourcie des fonctions "`scaleX`" et "`scaleY`"

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transform-function#Translations>
- <https://www.alsacreations.com/astuce/lire/1565-acceleration-materielle-au-service-de-vos-animations-css.html>

# Pratiquons ! - Essayons les transitions

- Appliquons la mise à l'échelle (1.5) sur l'image

```
.element-conteneur.transition-2 img {  
    compléter  
}
```

```
.element-conteneur.transition-2: hover img {  
    compléter  
}
```

# Transitions css – Fonction rotate()

- Permet de faire pivoter d'un élément selon un axe (centre / centre par défaut)
- Ces valeurs doivent être des valeurs valides pour un angle. Ex : -15deg ou 3.142rad
- Ne déforme pas l'élément
- Possède des fonctions sœurs : rotateX, rotateY, rotateZ
- Possède une version 3d : rotate3d()

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/transform-function/rotate>
- <https://developer.mozilla.org/fr/docs/Web/CSS/transform-function/rotate3d>
- <https://developer.mozilla.org/fr/docs/Web/CSS/transform-function#Rotation>



# Pratiquons ! - Essayons les transitions

- Appliquons la rotation (15deg) sur l'image

```
.element-conteneur.transition-2 img {  
    compléter  
}
```

```
.element-conteneur.transition-2: hover img {  
    compléter  
}
```

# Pratiquons ! - Essayons les transitions

- Et normalement, on devrait obtenir nos transitions cibles



# Quiz

**"Test - Pseudo-élément / pseudo-classe et transitions"  
sur ENT**

Conditions du test :

- 5 minutes
- Une tentative

# Propriétés max-width et max-height

- Définit la dimension maximale d'un élément
  - Permet donc d'éviter qu'un élément dépasse en taille
- Prioritaires comparé à leur équivalent fixe
- Géré par tous les navigateurs modernes
- Très utiles pour le responsive
- Possèdent des propriétés "inverses" : min-width / max-width :
  - Les propriétés "min-" surchargent les propriétés "max-" et "width / height"

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/max-width>
- <https://developer.mozilla.org/fr/docs/Web/CSS/max-height>
- <https://developer.mozilla.org/fr/docs/Web/CSS/min-width>
- <https://developer.mozilla.org/fr/docs/Web/CSS/min-height>
-

# Pratiquons ! - Essayons les transitions

- Appliquons la propriété "max-width" sur l'image

```
.element-conteneur.transition-2 img {  
    max-width: 100%;  
}
```



**Questions ?**