

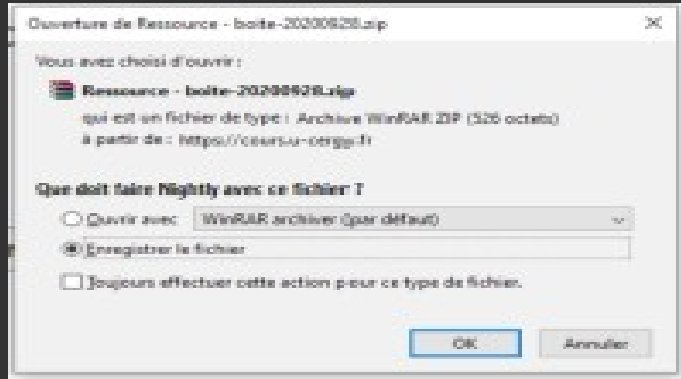
Intégration Web

MMI 1 – TP#5 S1

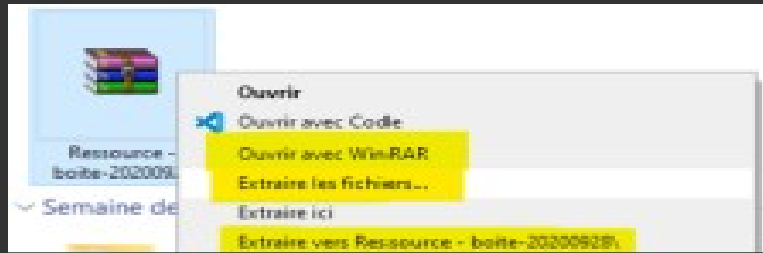


Danielo **JEAN-LOUIS**
Michele **LINARDI**

Petit point sur les ressources de cours

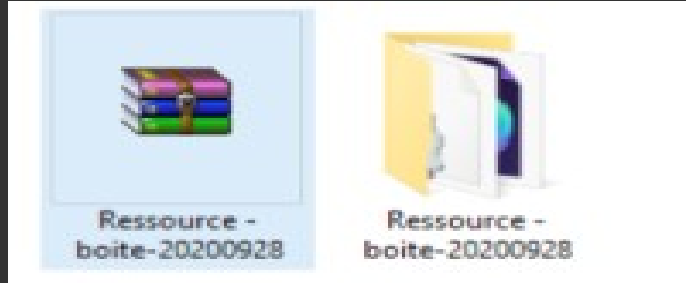


- Les ressources se présentent sous la forme d'une archive une fois téléchargées
- L'archive doit être impérativement extraite sinon vous ne serez pas capables de réaliser correctement les tp



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

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 a un lien vers le site B ça ne veut pas dire que mon site A fait partie du site B et vice-versa

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

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>

- Grâce au système d'imbrication html, une image ou un bloc 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> - Exemples de code

```
<a href="https://www.u-cergy.fr">  
  Site de Cergy Paris Université  
</a>
```

Mon lien s'ouvrira
dans le même onglet

Mon lien s'ouvrira
dans un nouvel onglet

```
<a  
  href="https://www.u-cergy.fr"  
  target="_blank"  
>  
  Site de Cergy Paris Université  
</a>
```

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>

Balise <a> - ancre sur la même page



```
<a  
  href="#ma-cible"  
>  
  Aller à la balise avec l'id "ma-cible"  
</a>
```

```
<div id="ma-cible">  
  <!--contenu de ma balise-->  
</div>
```

Au clic, je défilerais
automatiquement jusqu'à la
balise avec l'id "ma-cible"

Pratiquons ! - Liens (Partie 1)

Pré-requis :

- Avoir la ressource ressources/liens
- A télécharger ici :
<https://downgit.github.io/#/home?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-5/ressources>

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

Balise <a>

- Les liens déjà cliqués ont une apparence différente
- Les liens doivent être facilement repérable
 - couleur différente par exemple
- Différents états gérés par les pseudo-classes

Sources :

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

Pseudo-classe

- Permet d'appliquer des déclarations css sur un élément dans un état spécifique

```
sélecteur:pseudo-classe {  
    /* propriétés CSS */  
}
```

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 des pseudo-classes
- Certaines pseudo-classes sont propres à certaines balises
- Fonctionne avec tous les sélecteurs CSS

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

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

- 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>

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 (touche tabulation)

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)...
- Absence du caractère "#" dans la valeur de l'attribut "href"

Sources :

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

Pratiquons ! - Liens (Partie 2)

Pré-requis :

- Avoir la ressource ressources/liens
- A télécharger ici :
<https://downgit.github.io/#/home?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-5/ressources>

Point accessibilité – Les liens

- Lorsque votre site contient beaucoup de lien privilégiez leur distinction grâce à la pseudo-classe " :visited"
- Les intitulés doivent être clairs

Cliquer ici !



Accéder à votre compte



Sources :

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

Balise <a> - attribut href

- Schéma url : permet d'effectuer des actions spécifiques
 - mailto: envoyer une adresse mail
 - ex : "mailto:m.bluth@example.com"
 - tel: composer un numéro de téléphone
 - ex : "tel:+123456789"
 - Il en existe d'autres propres aux applications mobiles

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"
 - Indication dans le libellé du lien (à préférer)
- Indiquer à l'utilisateur quand un lien ouvre un nouvel onglet

Sources :

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

Balise <a> - attribut href

- Possibilité de combiner lien externe et interne
 - href="description.html#ma-cible"

Ici on accède à la balise ayant l'id "ma-cible" dans la page "description.html"

Sources :

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

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)
- Permet 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>

Point sécurité – Les liens

- Reverse tabnabbing
 - Technique consistant à modifier l'onglet ayant ouvert un lien vers un site malicieux, souvent dans le but de faire du phishing
 - Faille corrigée dans tous les navigateurs sauf Internet Explorer et Edge (Navigateurs de Microsoft)
 - Correction de la faille :
 - Rajouter l'attribut `rel="noopener"` pour les hyperliens ayant l'attribut `target="_blank"`
- Voir la ressource "reverse-tabnabbing" pour exemple

Sources :

- https://owasp.org/www-community/attacks/Reverse_Tabnabbing - anglais
- <https://mathiasbynens.github.io/rel-noopener/> - anglais

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>

Questions ?