

# Intégration Web

MMI 1 – TP#6 S1



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

# Balise <a>

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

Source(s) :

- <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 de balises html, une image ou un bloc de balises peut pointer vers une ressource

```
<a href="">  
  <section>  
      
    <p>Ma description [...]</p>  
  </section>  
</a>
```

Au clic sur l'image ou le texte,  
je serai redirigé

## 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>
```

Source(s) :

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

# Point accessibilité – Les liens

- Par défaut les liens s'ouvrent dans l'onglet courant
- **Prévenir l'utilisateur si votre lien va ouvrir un nouvel onglet**
  - Tous les utilisateurs ne sont pas familiers avec le système d'onglets

## Source(s) :

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

# Balise <a> - ancre sur la même page

- Permet d'atteindre une balise dans la page grâce à son attribut "id"
  - Exemple : chapitrage sur wikipedia
- Une balise 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 croisillon (#)
- **Rappel : la valeur d'un id doit être unique par page**

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/a>
- [https://fr.wikipedia.org/wiki/Hypertext\\_Markup\\_Language#De\\_2007\\_%C3%A0\\_nos\\_jours:\\_HTML\\_5\\_et\\_abandon\\_du\\_XHTML\\_2](https://fr.wikipedia.org/wiki/Hypertext_Markup_Language#De_2007_%C3%A0_nos_jours:_HTML_5_et_abandon_du_XHTML_2)



# Rappel : la valeur d'un id doit être unique par page

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/a>
- [https://fr.wikipedia.org/wiki/Hypertext\\_Markup\\_Language#De\\_2007\\_%C3%A0\\_nos\\_jours\\_:\\_%HTML\\_5\\_et\\_abandon\\_du\\_XHTML\\_2](https://fr.wikipedia.org/wiki/Hypertext_Markup_Language#De_2007_%C3%A0_nos_jours_:_%HTML_5_et_abandon_du_XHTML_2)

# Balise <a> - ancre sur la même page



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

Je définis un lien avec une ancre dans l'attribut "href"

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

Je définis une balise dont la valeur de l'attribut ID est égale à celle définie dans un href **sans le croisillon / dièse (#)**

Au clic sur le lien, 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://download-directory.github.io/?url=https%3A%2F%2Fgithub.com%2FDanYellow%2Fcours%2Ftree%2Fmain%2Fintegration-web-s1%2Ftravaux-pratiques%2Fnumero-6%2Fressources>

# Pratiquons ! - État du lien

[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
  - Exemple : couleur différente
- 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



```
.selecteur-css:pseudo-class {  
    /* propriétés CSS */  
}
```

## Source(s) :

- <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**
- Peut être propre à certaines balises
- Fonctionne avec tous les sélecteurs CSS (type, classes...)

## Source(s) :

- <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

## Source(s) :

- <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 dans le CSS :
  - :link — :visited — :hover — :active
- Vous utiliserez principalement ":hover"

## Source(s) :

- <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-within" et "focus"
  - Préférez "focus-within"
  - 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)

## Source(s) :

- <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

```
• • •  
  
.mon-lien:hover {  
    /* Je cible mon lien au survol */  
}
```

On cible tout élément ayant la classe "mon-lien" lors du survol.  
Et le navigateur appliquera les déclarations CSS à l'intérieur.

## Source(s) :

- <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 ! - Liens (Partie 2)

Pré-requis :

- Avoir la ressource **ressources/liens**

A télécharger ici :

<https://download-directory.github.io/?url=https%3A%2F%2Fgithub.com%2FDanYellow%2Fcours%2Ftree%2Fmain%2Fintegration-web-s1%2Ftravaux-pratiques%2Fnumero-6%2Fressources>

# Balise <a> - lien externe

- Utilisation de l'attribut "href"
- Permet de faire des liens absolus ou relatifs vers une ressource :
  - page web, fichiers (.jpg, .pdf, .doc)...

## Source(s) :

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

# Balise <a> - lien externe



```
<!-- Lien absolu -->  
<a href="https://www.youtube.com/">  
    Aller vers Youtube  
</a>
```

```
<!-- Lien relatif -->  
<a href="pages/page2.html">  
    Aller vers la page 2  
</a>
```

## Exemples

Source(s) :

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

## Balise <a> - lien externe

- Possibilité de passer des paramètres à la page suivante
  - Même manière que la méthode "GET" des formulaires

Source(s) :

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

# Balise <a> - lien externe

Caractère signifiant le début des paramètres d'URL. **Indispensable**

```
<a href="description.php?param1=toto&param2=world">
```

Accéder page 2

```
</a>
```

Paramètres de l'URL séparées par "&".  
Ces données seront envoyées à la page suivante

Paramètre

Séparateur entre chaque  
paire paramètre/valeur

Valeur

**Combien de paramètres allons-nous envoyer au clic ?**



# Chaîne de requête

- Autre nom des paramètres d'URL (slide précédente)
- **Les données ne sont pas sécurisées**
  - Ne pas utiliser dans un formulaire de connexion
- Query string en anglais

Source(s) :

- [https://fr.wikipedia.org/wiki/Cha%C3%AEne\\_de\\_requ%C3%AAt](https://fr.wikipedia.org/wiki/Cha%C3%AEne_de_requ%C3%AAt)

# Chaîne de requête

- Données envoyées récupérables uniquement avec un langage serveur
- Peut être facultatif dans une URL

Source(s) :

- [https://fr.wikipedia.org/wiki/Cha%C3%AEne\\_de\\_requ%C3%AAt](https://fr.wikipedia.org/wiki/Cha%C3%AEne_de_requ%C3%AAt)

# Balise <a> - lien externe

- Paramètres GET dans l'URL
  - Nombre infini de paramètres possibles (dans la limite historique des ~255 caractères, ~2 000 de nos jours)
  - Possibilité de mettre un paramètre sans valeur

Source(s) :

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

# Point sécurité – Les liens

- Reverse tabnabbing
  - Technique consistant à rediriger 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 (Navigateurs de Microsoft)
  - Correction de la faille manuelle :
    - Rajouter l'attribut `rel="noopener"` pour les hyperliens ayant l'attribut `target="_blank"`
- Voir la ressource "reverse-tabnabbing" pour exemple

## Source(s) :

- [https://owasp.org/www-community/attacks/Reverse\\_Tabnabbing](https://owasp.org/www-community/attacks/Reverse_Tabnabbing) - anglais
- <https://mathiasbynens.github.io/rel-noopener/> - anglais

# Pratiquons ! - Liens (Partie 3)

Pré-requis :

- Avoir la ressource **ressources/liens**

A télécharger ici :

<https://download-directory.github.io/?url=https%3A%2F%2Fgithub.com%2FDanYellow%2Fcours%2Ftree%2Fmain%2Fintegration-web-s1%2Ftravaux-pratiques%2Fnumero-6%2Fressources>

# Point accessibilité – Les liens

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

Cliquer ici !



L'intitulé n'est pas clair, l'utilisateur ne sait pas où il va aller au clic

Accéder à votre compte



L'intitulé est clair, l'utilisateur sait où il va aller au clic

Source(s) :

- <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 sites mobiles pour ouvrir le store ou encore une application depuis le site

## Source(s) :

- <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

## Source(s) :

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



# Point accessibilité – Les liens

- Sur mobile, les éléments cliquables doivent être plus gros et espacés
  - Utilisation de la propriété padding pour rendre la zone de click plus large
- Les liens doivent être facilement différenciables

## Source(s) :

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

# Point accessibilité – Les liens

## Source(s) :

- <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

## Source(s) :

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

## Balise <a> - attribut href

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

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

Source(s) :

- <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)
  - Notion abordée en S2

## Source(s) :

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

# Pseudo-classe - :hover

- Permet de cibler plusieurs balises en même temps grâce au système d'imbrications
- **Ne fonctionne pas sur mobile**

## Source(s) :

- <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

Source(s) :

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

# Pratiquons ! - Liens

Appliquez ce que vous venez d'apprendre en ajoutant de nouvelles zones à votre projet avec ces nouvelles notions





**Questions ?**