

# Programmation et projet encadré - L7TI005

mini-projet : préparation au travail en groupe

---

Yoann Dupont [yoann.dupont@sorbonne-nouvelle.fr](mailto:yoann.dupont@sorbonne-nouvelle.fr)

Pierre Magistry [pierre.magistry@inalco.fr](mailto:pierre.magistry@inalco.fr)

2022-2023

Université Sorbonne-Nouvelle  
INALCO  
Université Paris-Nanterre

## Les enjeux de cette portion de cours :

- continuer le mini-projet
- utiliser un format plus adapté pour le rendu final en groupe
  - sortie simple tabulaire
  - aujourd'hui : transformation en tableau HTML
- si le temps : ajouter d'autres colonnes au tableau

# HTML



# HTML, c'est quoi

HTML (**H**yper**T**ext **M**arkup **L**anguage) est un langage de balisage pour représenter des pages web. Format reconnu par tous les navigateurs.

# HTML, c'est quoi

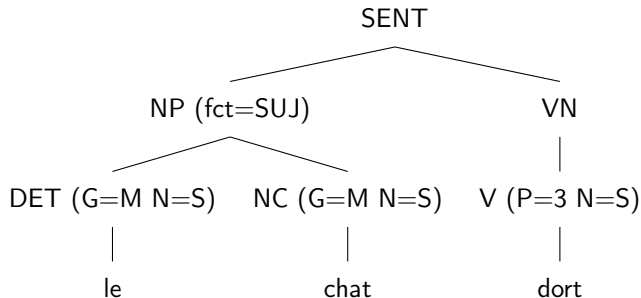
HTML (**H**yper**T**ext **M**arkup **L**anguage) est un langage de balisage pour représenter des pages web. Format reconnu par tous les navigateurs.

Permet de structurer l'information d'un page pour la rendre lisible :

- Dérivé du **SGML** (**S**tandard **G**eneralized **M**arkup **L**anguage) et "frère" du XML
- Permet de marquer des zones dans du contenu textuel
- Ces zones fournissent structure et enrichissements

## Du balisage, à quoi ça ressemble ? I

HTML définit des balises qui marquent explicitement le début et la fin d'une zone. On peut inclure des balises dans d'autres, mais pas de chevauchement : HTML se rapproche donc des constituants syntaxiques :



## Du balisage, à quoi ça ressemble ? II

Pour moins d'ambiguïté, les balises sont marquées explicitement. il y en a 3 types :

- Ouvrantes : `<balise>` → le début d'une zone
- Fermantes : `</balise>` → la fin d'une zone
- Autofermantes ou vides : `<balise/>` → position dans le document<sup>1</sup>

---

<sup>1</sup>Techniquement, cette syntaxe n'est nécessaire qu'en XML. En HTML, il est préférable de s'en passer, comme pour `<br>` par exemple. Source:

[https://developer.mozilla.org/fr/docs/Glossary/Void\\_element](https://developer.mozilla.org/fr/docs/Glossary/Void_element)

# Du balisage, à quoi ça ressemble ? II

Pour moins d'ambiguïté, les balises sont marquées explicitement. il y en a 3 types :

- Ouvrantes : `<balise>` → le début d'une zone
- Fermantes : `</balise>` → la fin d'une zone
- Autofermantes ou vides : `<balise/>` → position dans le document<sup>1</sup>

Les attributs d'une balise (donc du nœud) sont des couples clé/valeur renseignés sur la balise ouvrante ou autofermante :

- `<NP fct="SUJ">`
- `<DET G="M" N="S">`

---

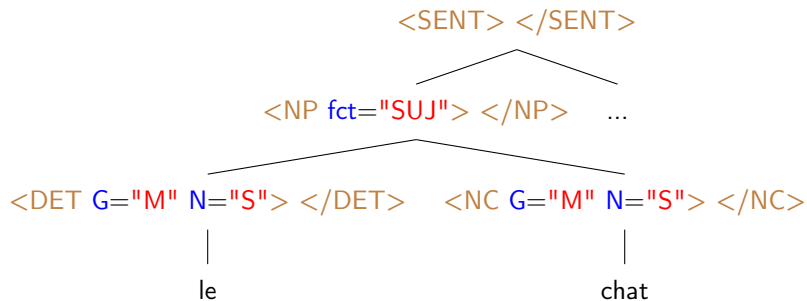
<sup>1</sup>Techniquement, cette syntaxe n'est nécessaire qu'en XML. En HTML, il est préférable de s'en passer, comme pour `<br>` par exemple. Source:

[https://developer.mozilla.org/fr/docs/Glossary/Void\\_element](https://developer.mozilla.org/fr/docs/Glossary/Void_element)



## Du balisage, à quoi ça ressemble ? III

Si on reprend l'exemple mais en XML :



## Du balisage, à quoi ça ressemble ? IV

Le fichier tel qu'il sera écrit au format texte :

```
<SENT>
  <NP fct="SUJ">
    <DET G="M" N="S">Le</DET>
    <NC G="M" N="S">chat</NC>
  </NP>
  <VN>
    <V P="3" N="S">dort</V>
  </VN>
</SENT>
```

HTML reprend la construction globale du balisage, mais a sa propre syntaxe :

```
<html>  
  <head>...</head>  
  <body>...</body>  
</html>
```

# HTML : les spécificités

HTML reprend la construction globale du balisage, mais a sa propre syntaxe :

```
<html>  
  <head>...</head>  
  <body>...</body>  
</html>
```

Où :

- head : l'entête du fichier (avec les métadonnées)
- body : le corps du fichier (avec le contenu textuel et la structure)

---

Bonne ressource pour approfondir HTML : <https://www.w3schools.com/html/default.asp>

# HTML : l'entête

L'entête `head`<sup>2</sup> contient beaucoup d'informations intéressantes. On reviendra plus en détail plus tard sur certaines.

Une métadonnée particulière nous sera intéressante ici : l'encodage (charset)<sup>3</sup>.

---

<sup>2</sup>Documentation entête HTML : [https://www.w3schools.com/html/html\\_head.asp](https://www.w3schools.com/html/html_head.asp)

<sup>3</sup>Documentation charset HTML : [https://www.w3schools.com/html/html\\_charset.asp](https://www.w3schools.com/html/html_charset.asp)

# HTML : l'entête

L'entête `head`<sup>2</sup> contient beaucoup d'informations intéressantes. On reviendra plus en détail plus tard sur certaines.

Une métadonnée particulière nous sera intéressante ici : l'encodage (`charset`)<sup>3</sup>.

```
<html>
  <head>
    [...]
    <meta charset="UTF-8" />
    [...]
  </head>
  <body>...</body>
</html>
```

---

<sup>2</sup>Documentation entête HTML : [https://www.w3schools.com/html/html\\_head.asp](https://www.w3schools.com/html/html_head.asp)

<sup>3</sup>Documentation charset HTML : [https://www.w3schools.com/html/html\\_charset.asp](https://www.w3schools.com/html/html_charset.asp)

# HTML : Créer un tableau

Pour créer un tableau en HTML, nous avons besoin de 4 balises :

- `table` : la balise racine du tableau
- `tr` : ***table row***, une ligne (se place dans `table`)
- `th` : ***table header***, une cellule d'entête (seulement la première ligne)
- `td` : ***table data***, une cellule classique (toutes les lignes pas entête)

# HTML : Créer un tableau

Pour créer un tableau en HTML, nous avons besoin de 4 balises :

- `table` : la balise racine du tableau
- `tr` : *table row*, une ligne (se place dans `table`)
- `th` : *table header*, une cellule d'entête (seulement la première ligne)
- `td` : *table data*, une cellule classique (toutes les lignes pas entête)

```
<table>
  <tr><th>livre</th><th>taille</th></tr>
  <tr>
    <td>Du côté de chez Swann</td><td>1.0Mo</td>
  </tr>
  <tr>
    <td>L'Assommoir</td><td>990 ko</td>
  </tr>
</table>
```



## Exercice : transformer la sortie tabulaire en HTML

En partant du miniprojet de la semaine dernière, nous voulons à présent transformer la sortie tabulaire en sortie au format HTML. La page produite devra contenir :

1. un entête
2. un corps
  - devra contenir au moins le tableau des données récupérées, avec une ligne d'entête et les résultats pour chaque URL.

Ce code HTML devra être écrit dans un fichier `.html` qui devra être lisible par un navigateur web quelconque.

# Structure attendue pour le mini-projet

En supposant que votre dépôt s'appelle **PPE1-2023**, l'arborescence de vos fichiers devra ressembler à cela une fois tous les exercices finis :

```
PPE1-2023
├── journal.md
├── exercices
│   └── ...
├── miniprojet
│   ├── programmes
│   │   └── miniprojet.sh
│   ├── urls
│   │   └── fr.txt
│   └── tableaux
│       └── tableau-fr.html
```