

Introduction à l'informatique

5 – HTML et le web

Julien Deantoni

Contenue de l'UE (non contractuel)

- 1) Introduction, notion de variables, de type de données simple et structuré
- 2) Variables, séquence d'instructions en mémoire, Notion d'algorithme, de boucles et de branchements conditionnels
- 3) Logique Booléenne et comparaison, lecture/écriture dans un fichier
- 4) Écrire des fonctions afin de rendre le code plus lisible et réutilisable
- 5) Introduction au web, structure d'une page en web en HTML**
- 6) Front-end et back-end
- 7) Contrôle continu intermédiaire (partiel)
- 8) Définir une web app avec python, html et CSS
- 9) Résumé et sucre syntaxique

Internet versus web ?

- Internet est un ensemble d'ordinateurs, reliés entre eux par un réseau, leur permettant ainsi d'échanger toutes sortes de données.
 - Basé sur l'ensemble de protocoles TCP/IP
 - Sans centre névralgique
 - permet des applications variées comme le courrier électronique, la messagerie instantanée, le partage de fichiers en pair-à-pair, le streaming, le podcasting, la téléconférence et aussi **le World Wide Web**
- Le « Web » est un système *hypertexte* public fonctionnant sur Internet. Le Web permet de consulter (avec un navigateur), des pages accessibles sur des sites.

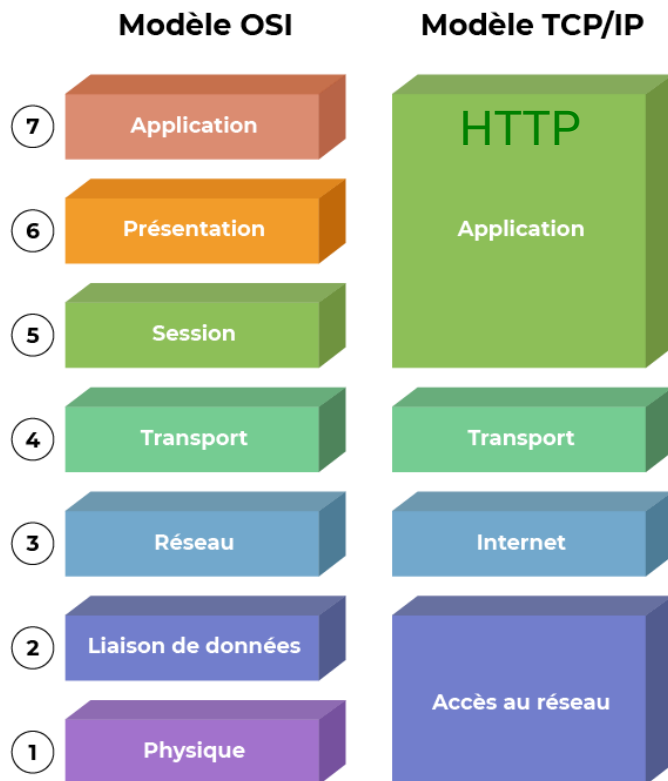
TCP/IP

- Internet Protocol :
 - C'est un protocole basé sur une notion d'adresse unique pour chaque nœud publique directement lié au réseau internet (par exemple votre box internet)
→ l'adresse IP publique.
 - L'adresse IP s'obtient en ayant un fournisseur d'accès à internet.
 - L'adresse IP permet non seulement d'envoyer les bonnes données au bon endroit mais elle permet aussi d'**estimer approximativement** où se trouve l'ordinateur lié à internet (<https://mon-adresse-ip.fr/>)
 - Sur un ordinateur, l'adresse de la machine sur laquelle on « travaille » est toujours `127.0.0.1` par convention, avec le nom `localhost`
- Transmission Control Protocol :
 - Permet d'établir une liaison entre un *serveur* et un *client* afin d'échanger des données

Le modèle OSI

- Le modèle OSI (de l'anglais Open Systems Interconnection) est une norme de communication, en réseau, de tous les systèmes informatiques. C'est un modèle de communications entre ordinateurs proposé par l'ISO (Organisation internationale de normalisation) qui décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions.

https://fr.wikipedia.org/wiki/Mod%C3%A8le_OSI



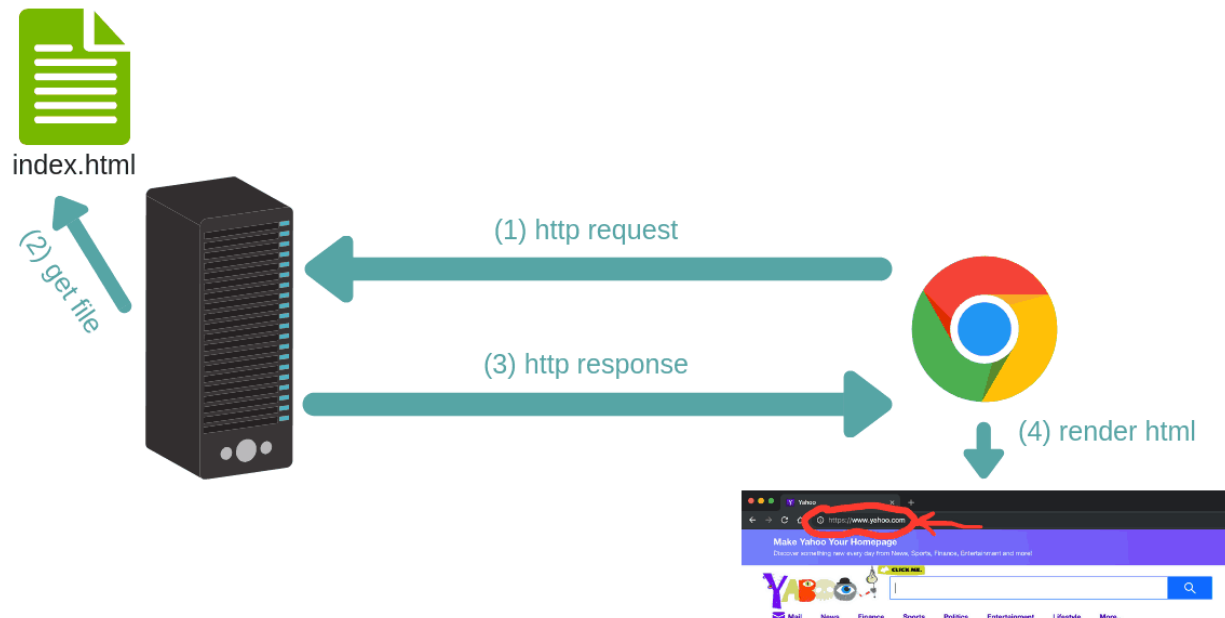
*Permet bien plus que le web !
e.g., ssh, ftp, smtp*

HTTP et le web

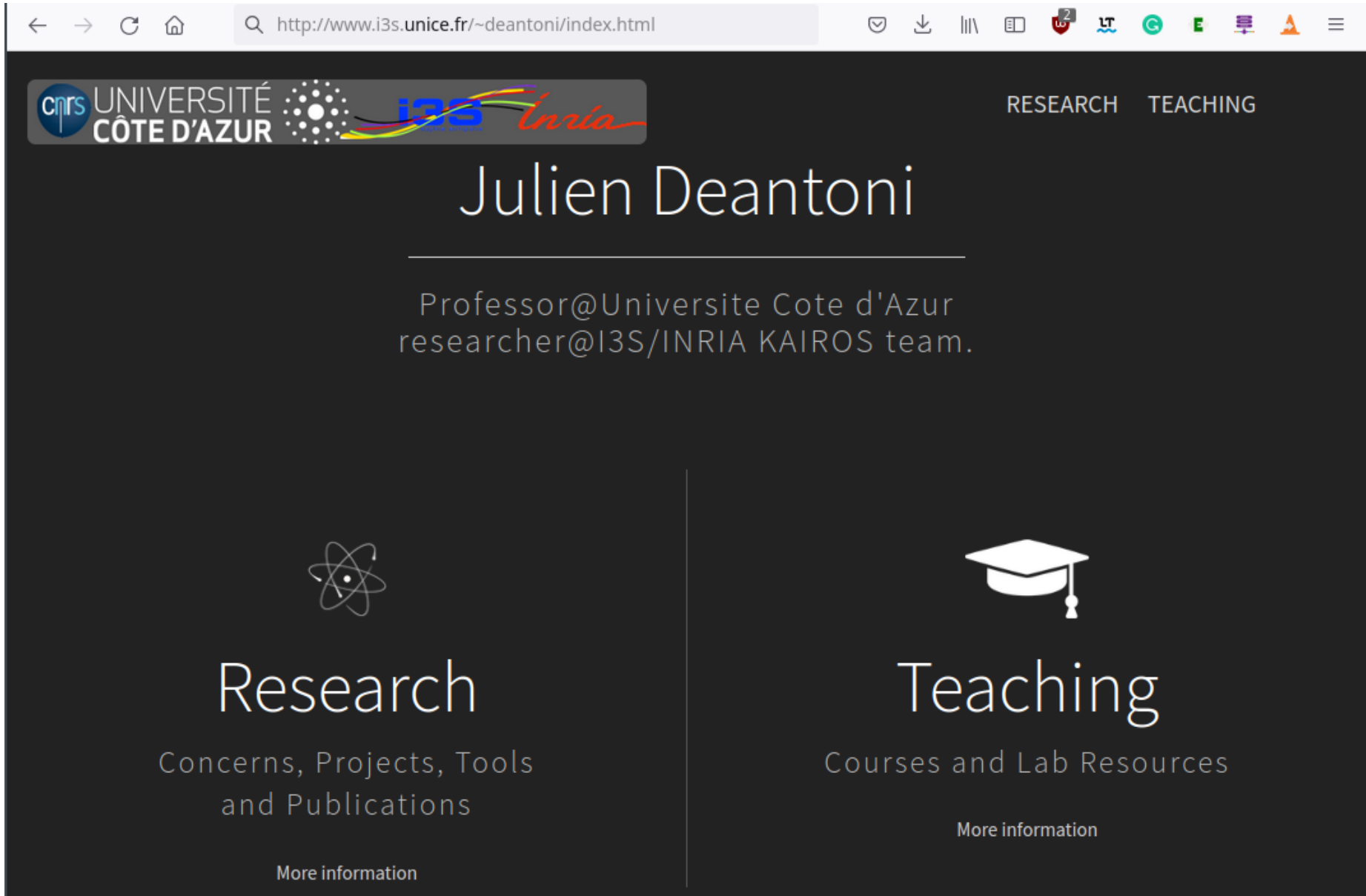
- L'HyperText Transfer Protocol, généralement abrégé HTTP, littéralement « protocole de transfert hypertexte », est un **protocole de communication client-serveur développé pour le World Wide Web**. HTTPS (avec S pour secure, soit « sécurisé ») est la variante sécurisée par le chiffrement et l'authentification.
- HTTP est un protocole de la couche application dans le modèle OSI. Il peut fonctionner sur n'importe quelle connexion fiable. Dans les faits on utilise le protocole TCP comme couche de transport. Un serveur HTTP utilise alors par défaut **le port 80** (443 pour HTTPS).

HTTP et le web

- L'Hypertext Transfer Protocol, généralement abrégé HTTP, littéralement « protocole de transfert hypertexte », est un **protocole de communication client-serveur développé pour le World Wide Web**. HTTPS (avec S pour secure, soit « sécurisé ») est la variante sécurisée par le chiffrement et l'authentification.
- Client - Serveur :
 - Le client est un programme qui veut récupérer du contenu, par exemple votre navigateur web
 - Le serveur attend qu'un client se connecte et lui permet d'accéder à un document donné



<https://www.afternerd.com/blog/python-http-server/>



The screenshot shows a web browser window with the address bar displaying `http://www.i3s.unice.fr/~deantoni/index.html`. The website has a dark background. At the top left, there is a header with logos for CNRS, Université Côte d'Azur, i3s, and Inria. To the right of these logos are the words "RESEARCH" and "TEACHING". In the center, the name "Julien Deantoni" is written in a large, white, sans-serif font. Below the name, a horizontal line separates it from the text "Professor@Universite Cote d'Azur" and "researcher@I3S/INRIA KAIROS team." in a smaller, white, sans-serif font. The page is divided into two main sections by a vertical line. The left section is titled "Research" in a large, white, sans-serif font, with a small atomic symbol icon above it. Below the title, the text "Concerns, Projects, Tools and Publications" is written in a smaller, white, sans-serif font. At the bottom of this section is a link "More information" in a smaller, white, sans-serif font. The right section is titled "Teaching" in a large, white, sans-serif font, with a small graduation cap icon above it. Below the title, the text "Courses and Lab Resources" is written in a smaller, white, sans-serif font. At the bottom of this section is a link "More information" in a smaller, white, sans-serif font.

← → ↻ 🏠 🔍 `http://www.i3s.unice.fr/~deantoni/index.html` 📧 ⬇️ 📖 📅 📌 📌 📌 📌 📌 📌 📌 📌

 UNIVERSITÉ CÔTE D'AZUR  

RESEARCH TEACHING

Julien Deantoni

Professor@Universite Cote d'Azur
researcher@I3S/INRIA KAIROS team.



Research

Concerns, Projects, Tools
and Publications

More information



Teaching

Courses and Lab Resources

More information



- URL de la page demandée :

<http://www.i3s.unice.fr/~deantoni/index.html>

protocole

serveur hôte

chemin

fichier

- Actions effectuées :

- 1) Demande de l'adresse ip du serveur hôte à un serveur particulier nommé DNS (Domain Name Server)

```
jdeanton@localhost:~> host www.i3s.unice.fr
www.i3s.unice.fr is an alias for www-vm01.i3s.unice.fr.
www-vm01.i3s.unice.fr has address 134.59.130.31
jdeanton@localhost:~> host www.i3s.univ-cotedazur.fr
www.i3s.univ-cotedazur.fr is an alias for www-vm01.i3s.univ-cotedazur.fr.
www-vm01.i3s.univ-cotedazur.fr has address 134.59.130.31
```



- URL de la page demandée :

http://www.i3s.unice.fr/~deantoni/index.html

protocole

serveur hôte

chemin

fichier

- Actions effectuées :

- 1) Demande de l'adresse ip du serveur hôte à un serveur particulier nommé DNS (Domain Name Server)
- 2) Requête «CONNECT» à l'hôte puis « GET » avec le chemin et nom de fichier en paramètre

```
jdeanton@linux-hani:~/tmp/IIWtests> wget 134.59.130.2:80/~deantoni/index.html
--2022-10-16 16:46:57-- http://134.59.130.2/~deantoni/index.html
Connecting to 134.59.130.2:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 31555 (31K) [text/html]
Saving to: 'index.html'
```

Récupération du fichier demandé (par le navigateur web)

- 3) Interprétation du fichier récupéré par le navigateur web

HTML : Hyper Text Markup Language



- Langage à **balise** pour l'hyper text. Ce n'est pas un langage de programmation. Aucun algorithme n'est décrit.

Langage à balise

- En informatique, les langages à balise représentent une classe de langages spécialisés dans **l'enrichissement d'information textuelle**. Ils utilisent des **balises**, unités syntaxiques délimitant une séquence de caractères ou marquant une position précise à l'intérieur d'un flux de caractères (par exemple un fichier texte).
- L'inclusion de balises permet de transférer à la fois la structure du document et son contenu. **Cette structure est compréhensible par un programme informatique, ce qui permet un traitement automatisé du contenu.**
- Exemple :

Texte du titre

Balises ouvrantes `<title>` ceci est un titre avec un mot en `<strong>` gras `</strong>` `</title>` **Balises fermantes**

Contenu en gras

HTML : Hyper Text Markup Language



- Une balise est entourée de « chevrons », c'est à dire des symboles `<` et `>`, comme ceci : `<balise>` pour l'ouverture de la balise. Pour la fermeture de balise on rajoute un `/` devant le nom, comme cela : `</balise>`
- Les balises permettent au programme qui lit le fichier (au navigateur web pour html) de comprendre quelle est la nature du texte entre la balise ouvrante et la balise fermante ; et ainsi de la traiter comme bon se doit. Cela va par exemple permettre de définir des titres, des paragraphes, des liens, des images, etc.
- Pour tout langage à balise comme ceux ci dessus, il y a une description des balises qui permet à un programme de savoir **quelles sont les balises possibles et leur structure**. Cette description est faite dans un fichier DTD (Document type definition) ; qui donne le **type** du document
- HTML est donc un langage à balise dont la DTD définit les balises permettant de structurer les documents HyperText

HTML : Hyper Text Markup Language

- Le document est de type HTML (sa DTD est celle d'HTML)

- Le document html commence

- L'entête définit différentes choses dont le titre de l'onglet puis se termine

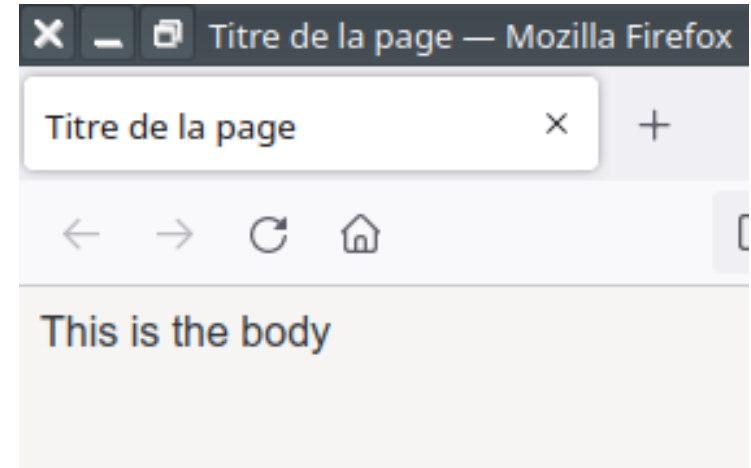
- Le corps du document est ici vide, c'est à dire que la page sera vide

- Le document html finit

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre</title>
  </head>
  <body>
  </body>
</html>
```

⚠ Les balises ne doivent pas être entrelacées, la première qui ouvre est la dernière qui ferme `<p><b>...</p></b>`

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre de la page</title>
  </head>
  <body>
    This is the body
  </body>
</html>
```



- Certaines balises peuvent avoir des attributs qui sont spécifiés après le nom de la balise ouvrante, toujours entre les chevrons.
 - Ils précisent la nature ou le paramétrage de la balise
 - Certaines propriétés sont obligatoires
 - Propres à chaque balise
- Par exemple la balise permettant de faire les liens hypertexte (c'est à dire les liens entre documents) :

Diagram illustrating the structure of an HTML link tag:

```
<a href="http://univ-cotedazur.fr"> site UCA </a>
```

Labels and components:

- nom attribut**: Points to the `href` attribute name.
- Texte du lien**: Points to the text content `site UCA`.
- Balise ouvrante**: Points to the opening tag `<a`.
- valeur de attribut**: Points to the attribute value `"http://univ-cotedazur.fr"`.
- Balises fermantes**: Points to the closing tag `</a>`.

- Certaines balises peuvent avoir des attributs qui sont spécifiés après le nom de la balise ouvrante, toujours entre les chevrons.
 - Ils précisent la nature ou le paramétrage de la balise
 - Certaines propriétés sont obligatoires
 - Propres à chaque balise
- Par exemple la balise permettant de faire les liens hypertext (c'est à dire les liens entre documents) :

Diagram illustrating the structure of an HTML link tag:

```
<a href="http://univ-cotedazur.fr"> site UCA </a>
```

Labels and annotations:

- nom attribut**: Points to the `href` attribute name.
- Texte du lien**: Points to the link text `site UCA`.
- Balise ouvrante**: Points to the opening tag `<a`.
- valeur de attribut**: Points to the attribute value `"http://univ-cotedazur.fr"`.
- Balises fermantes**: Points to the closing tag `</a>`.

Les liens peuvent être relatifs ou absolus; locaux ou non.

- Lien local entre deux pages html:

```
<a href="./uneAutrePage.html"> une autre page </a>
```

- Lien local au sein d'une même page

```
<XXX id= "monID" .....>
```

```
<a href="#monID"> un autre endroit dans la page </a>
```

- Lien distant ci dessous, il peut aussi désigner un fichier particulier et une cible particulière, e.g `http://zaza.com/fichier.html#idDuneBalise`

Diagram illustrating the structure of an HTML link:

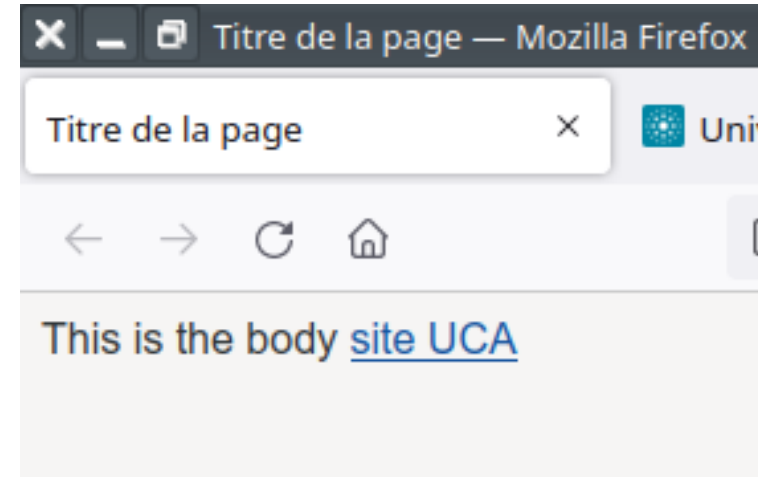
```
<a href="http://univ-cotedazur.fr" site UCA </a>
```

Labels and annotations:

- nom attribut**: Points to the `href` attribute name.
- Texte du lien**: Points to the text `site UCA` between the tags.
- Balise ouvrante**: Points to the opening tag `<a`.
- valeur de attribut**: Points to the value `"http://univ-cotedazur.fr"` of the `href` attribute.
- Balises fermantes**: Points to the closing tag `</a>`.

Les liens peuvent être relatifs ou absolus; locaux ou non.

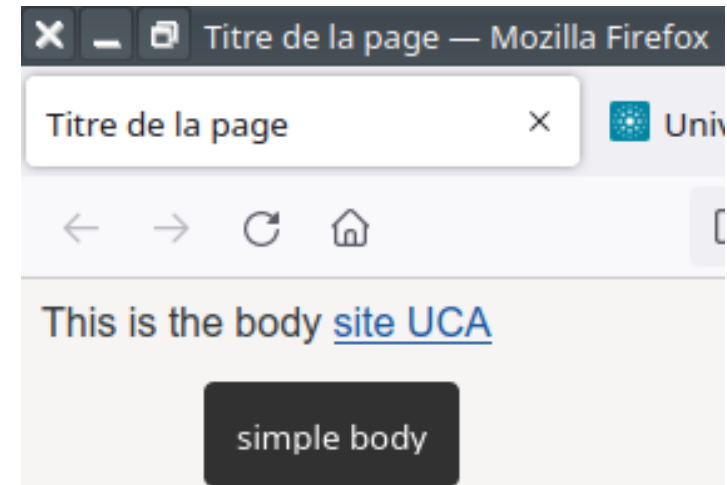
```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre de la page</title>
  </head>
  <body>
    This is the body
    <a href="http://univ-cotedazur.fr"> site UCA </a>
  </body>
</html>
```



HTML : Hyper Text Markup Language

- Attributs communs à toutes les balises :
 - `id` : pour donner un identificateur unique
 - `class` : pour attribuer une famille de style (de représentation)
 - `title` : élément d'information (visible souris immobile, tooltip)
 - `lang` : pour préciser la langue utilisée
- Liste exhaustive : [https://www.w3schools.com/tags/ref_standardattributes.a
sp](https://www.w3schools.com/tags/ref_standardattributes.asp)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre de la page</title>
  </head>
  <body title="simple body">
    This is the body
    <a href="http://univ-cotedazur.fr"> site UCA </a>
  </body>
</html>
```



- Certaines balises sont dites *orphelines*, i.e., elles n'englobent pas de texte et leur effet est uniquement capturé par la balise elle-même et dans ses attributs. C'est à la fois une balise ouvrante et fermante.

< baliseOrpheline att1= "valeur1" />

- Par exemple la balise permettant d'incorporer des images :

nom attribut

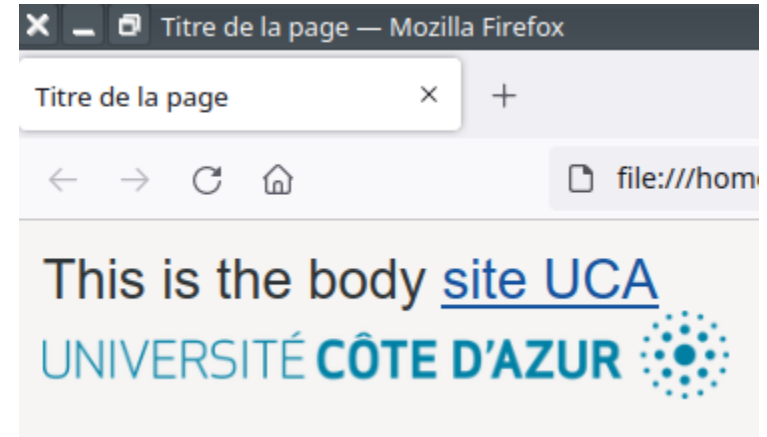

Balise
orpheline

valeur de attribut

Les liens peuvent être relatifs ou absolus; locaux ou non.

- Ou la balise permettant de faire un retour à la ligne


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8"/>
    <title>Titre de la page</title>
  </head>
  <body title="simple body">
    This is the body
    <a href="http://univ-cotedazur.fr"> site UCA </a>
    <br/>
    
  </body>
</html>
```



- **Objectif: Structuration d'un document**

- Délimitation des « parties » du document par des balises.
- Une balise contient des textes et/ou d'autres balises autorisées.

- **Concept de balise:**

`<balise></balise>`

`<balise/>` (balises sans texte, aka balise orpheline)

`<balise> texte </balise>`

`<balise attribut='valeur'></balise>`

- **Structure arborescente:**

`<root>`

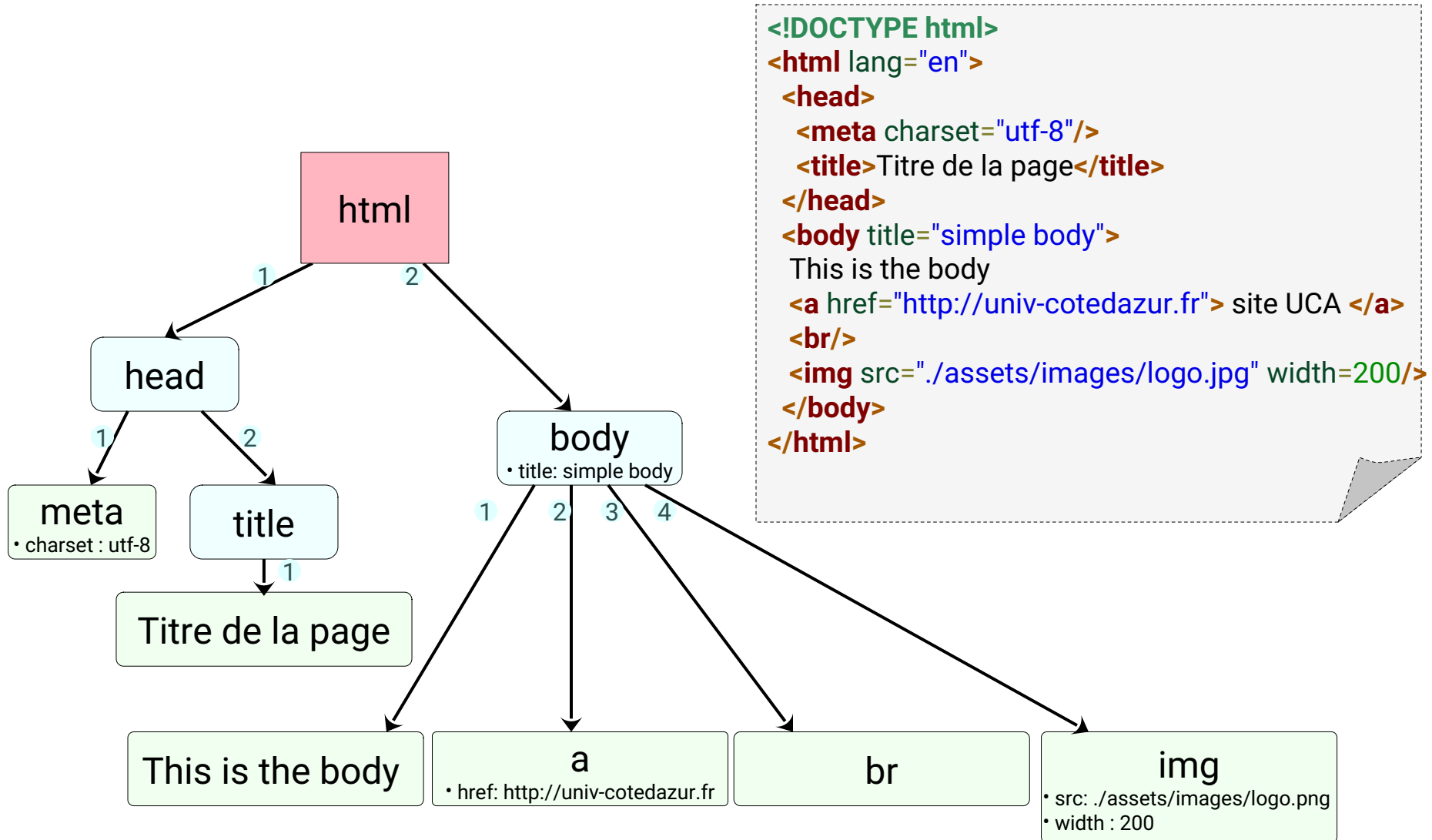
`<level1>`

`<level2></level2>`

`</level1>`

`</root>`

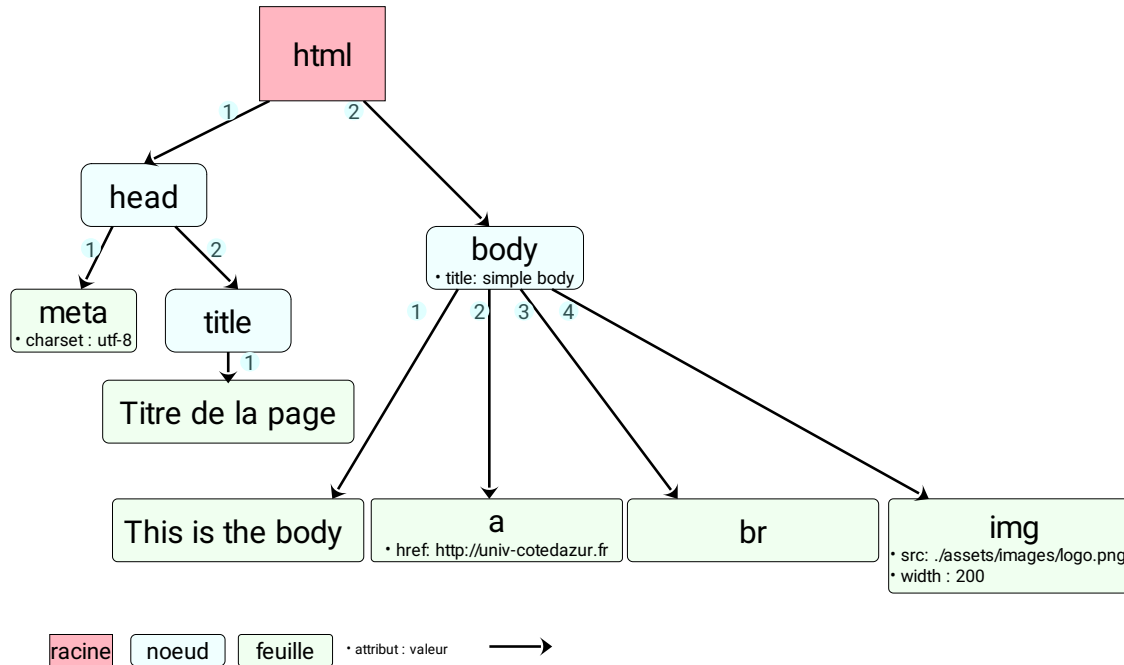
Structure arborescente d'HTML



racine noeud feuille • attribut : valeur →

Structure arborescente d'HTML

Une manière simpliste parmi d'autres



```
@dataclass
class Attribute:
    name: str
    value: str

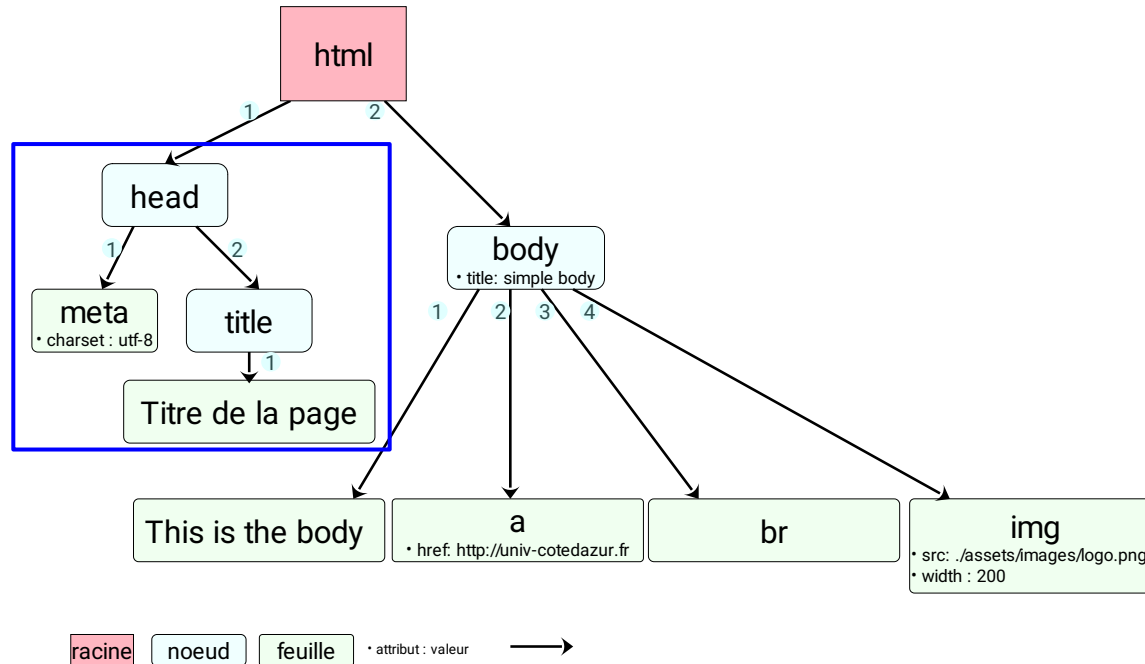
@dataclass
class Node:
    name: str
    value: str
    isOrphean: bool
    attributes: list[Attribute]
    children: list[Node]

@dataclass
class Tree:
    root: Node
```

- Cette structure, en mémoire est facile à manipuler informatiquement et est largement utilisée. Le concept est générique et peut être adapté à différents cas (arbre d'expression, arbre binaire de recherche, grammaire, html).

Structure arborescente d'HTML

Une manière simpliste parmi d'autres



```
@dataclass
class Attribute:
    name: str
    value: str

@dataclass
class Node:
    name: str
    value: str
    isOrphean: bool
    attributes: list[Attribute]
    children: list[Node]

@dataclass
class Tree:
    root: Node
```

- Cette structure, en mémoire est facile à manipuler informatiquement et est largement utilisée. Le concept est générique et peut être adapté à différents cas (arbre d'expression, arbre binaire de recherche, grammaire, html).

```
title: Node = Node("title", "Titre de la page", False, [], [])
meta: Node = Node("meta", "", True, list([Attribute("charset", "utf-8")]), [])
head: Node = Node("head", "", False, [], [meta, title])
```

- **Objectif: Structuration d'un document**

- Délimitation des « parties » du document par des balises.
- Une balise contient des textes et/ou d'autres balises autorisées.

- **Concept de balise:**

`<balise></balise>`

`<balise/>` (balises sans texte, aka balise orpheline)

`<balise> texte </balise>`

`<balise attribut='valeur'></balise>`

- **Structure arborescente:**

`<root>`

`<level1>`

`<level2></level2>`

`</level1>`

`</root>`

Titres (Headings)

- Titres:

```
<h1>Titre 1</h1>
```

```
<h2>Titre 2</h2>
```

```
<h3>Titre 3</h3>
```

```
<h4>Titre 4</h4>
```

```
<h5>Titre 5</h5>
```

```
<h6>Titre 6</h6>
```

Titre 1

Titre 2

Titre 3

Titre 4

Titre 5

Titre 6

Formulaires

- Permet la saisie d'information.
- Tous les formulaires HTML débutent par un élément `<form>`
`<form action="/my-handling-form-page" method="post">`
`</form>`
- **Action:** Ou envoyer les données
- **Method:** Quelle méthode HTTP utiliser

Formulaires

- Différentes possibilités pour la saisie :
 - *input* : *tel, color, url, range, email, search, date, etc.*

```
<input name="Email" type="email" placeholder="compléter votre email" required>
```

- *select*
 - *textarea*
 - *button*
 - *text*
 - *radio*
 - *etc.*
- Quelques attributs:
 - **Placeholder** : texte indicatif dans le champ.
 - **Required** : oblige la saisie du champ.

Grouping content

- **<p>** : un paragraphe
- **<div>** : regroupement logique utilisé pour le CSS et le JS
- **<blockquote>** : faire une citation
- **** : liste non ordonnée, contient des
- **** : liste ordonnée, contient des
- **** : élément d'une liste

```
<ul>
```

```
  <li>Item 1</li>
```

```
  <li>Item 2:
```

```
    <ol>
```

```
      <li>Sub-item1</li>
```

```
    </ol>
```

```
  </li>
```

```
</ul>
```

- Item 1
- Item 2:
 - 1. Sub-item1

⇒ L'ensemble des balises HTML est disponible ici :

<https://www.w3schools.com/TAGs/>

Grouping content

- **<p>** : un paragraphe
- **<div>** : regroupement logique utilisé pour le CSS et le JS
- **<blockquote>** : faire une citation
- **** : liste non ordonnée, contient des
- **** : liste ordonnée, contient des
- **** : élément d'une liste

```
<ul>
```

```
  <li>Item 1</li>
```

```
  <li>Item 2:
```

```
    <ol>
```

```
      <li>Sub-item1</li>
```

```
    </ol>
```

```
  </li>
```

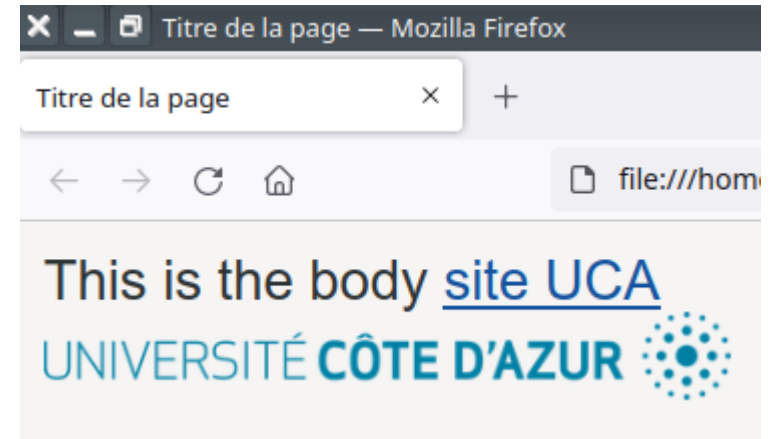
```
</ul>
```

- Item 1
- Item 2:
 - 1. Sub-item1

⇒ L'ensemble des balises HTML est disponible ici :

<https://www.w3schools.com/TAGs/>

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre de la page</title>
  </head>
  <body>
    This is the body
    <a href="http://univ-cotedazur.fr"> site UCA </a>
    <br/>
    
  </body>
</html>
```



HTML ne définit que la structure, pas la manière de représenter visuellement le texte sous jacent, même si il y a une représentation par défaut.

→ un fichier spécifique dans un autre langage (CSS) permet de donner la *couche* représentation, c'est à dire le **style** de chaque élément.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre de la page</title>
    <link href="fichierDeStyle.css" rel="stylesheet" />
  </head>
  <body>
    This is the body
    <a href="http://univ-cotedazur.fr"> site UCA </a>
    <br/>
    
  </body>
</html>
```

HTML ne définit que la structure, pas la manière de représenter visuellement le texte sous jacent, même si il y a une représentation par défaut.

→ un fichier spécifique dans un autre langage (CSS) permet de donner la *couche* représentation, c'est à dire le **style** de chaque élément. La bonne pratique est de référencer le fichier css depuis le fichier html par la balise orpheline `<link>`

Les liens peuvent être relatifs ou absolus; locaux ou non.

- Les feuilles de style en cascade (CSS) sont utilisées pour formater la mise en page d'une page Web.
- Avec CSS, vous pouvez contrôler la couleur, la police, la taille du texte, l'espacement entre les éléments, la manière dont les éléments sont positionnés et disposés, les images d'arrière-plan ou les couleurs d'arrière-plan à utiliser, différents affichages pour différents appareils et tailles d'écran, et beaucoup plus!
- Le mot cascade signifie qu'un style appliqué à un élément parent s'appliquera également à tous les éléments enfants du parent. Ainsi, si vous définissez la couleur du corps du texte sur "bleu", tous les titres, paragraphes et autres éléments de texte du corps auront également la même couleur (sauf si vous spécifiez autre chose) !

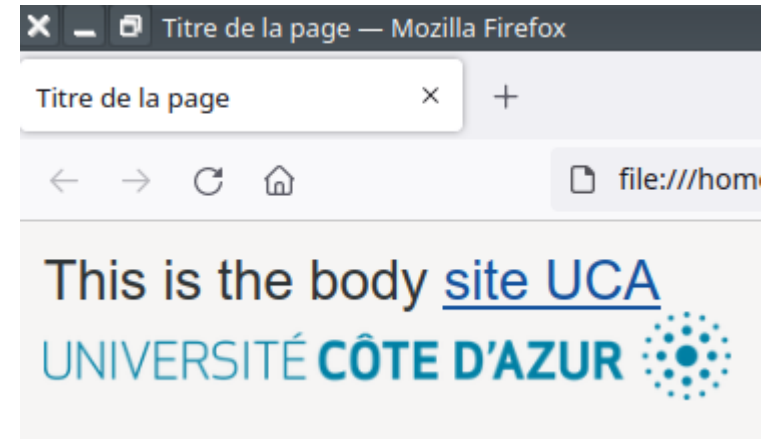
https://www.w3schools.com/html/html_css.asp

- CSS est basé sur la notion de **règle** qui s'applique à des nœuds particuliers de l'arbre de la structure html.

Notes : Pour 1 document il peut y avoir une infinité de styles : <http://www.csszengarden.com>
On ne verra pas toutes les propriétés ! <https://www.w3schools.com/w3css/default.asp>

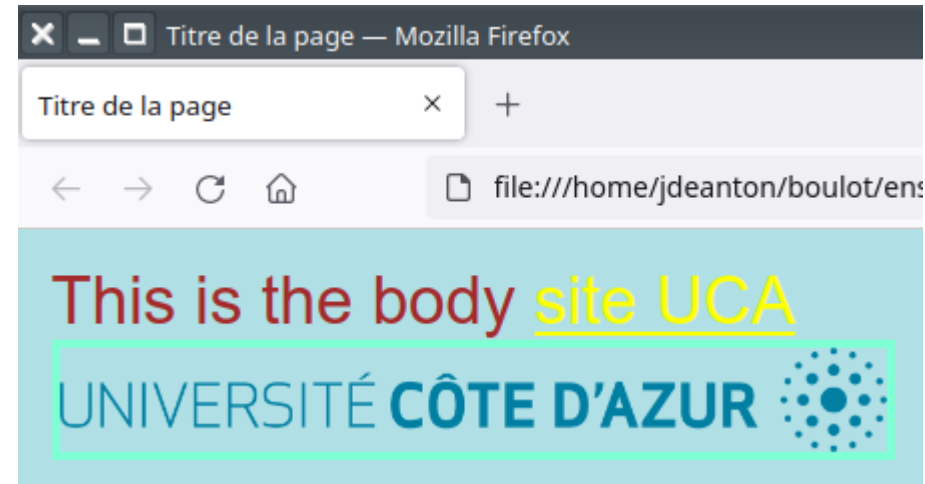
CSS : Cascading Style Sheets

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre de la page</title>
    <link href="fichierDeStyle.css" rel="stylesheet" />
  </head>
  <body>
    This is the body
    <a href="http://univ-cotedazur.fr"> site UCA </a>
  </body>
</html>
```



```
body {
  background-color: powderblue;
  color: brown;
}
a {
  color: yellow;
}
img {
  border: 2px solid #7FFFD4;
}
```

fichierDeStyle.css



3 règles
CSS

Les sélecteurs permettent de spécifier les nodes de l'arbre sous jacent à HTML

```
selector, ... {  
  property value ...;  
  ...  
}
```

Toutes les balises h1, h2, h3

- **Exemples:**

```
h1, h2, h3 {  
  Border: 2px solid grey;  
}
```

**Toutes les balises avec
l'attribut class = "title"**

```
.title {  
  font-weight: bold;  
}
```

La balise avec id="main"

```
#main {  
  font-style: italic;  
}
```

Toutes les balises

```
* {  
  font-style: italic;  
}
```

Sélecteurs avancés

Les sélecteurs avancés permettent de spécifier des relations dans l'arbre sous jacent à HTML (i.e., des sous arbres)

`h1 + p {` ← **Tous les p suivant un h1**

`...
}`

`h1 p {` ← **Tous les p dans un h1**

`...
}`

`h1 > p {` ← **Tous les p directement fils d'un h1**

`...
}`

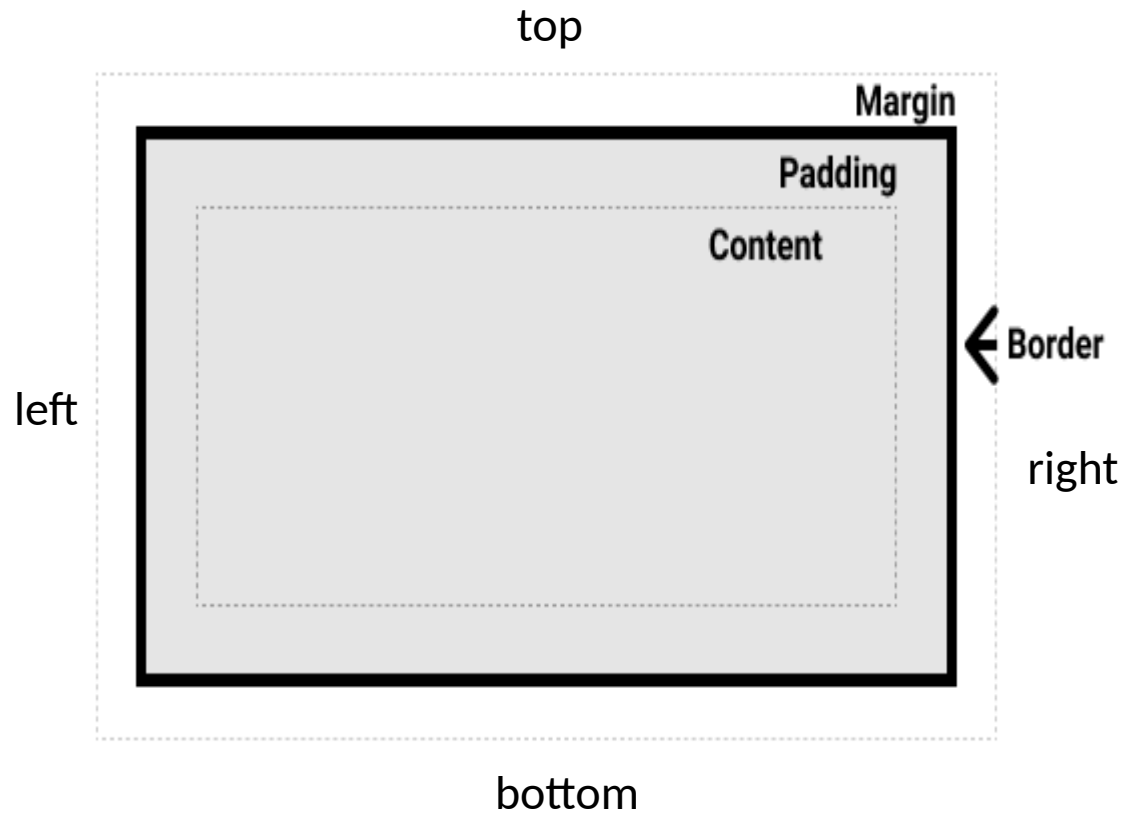
`h1[title]{` ← **Tous les h1 avec l'attribut title**

`...
}`

Le modèle à boîte

- Tout élément HTML est contenu dans une boîte.
- Il y a deux types de boîtes:
 - *Inline* : alignées horizontalement, prennent la largeur du contenu.
 - *Block* : alignées verticalement, prennent toute la largeur du contenant.

Le modèle à boîte



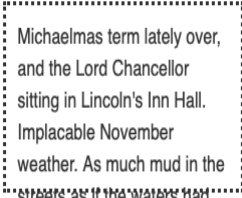
Les positions

- Cinq types de positions:
 - **Static** : juste positionnée dans le flux normal
 - **Relative**: Placée dans le flux normal puis décalée horizontalement ou verticalement
 - **Fixed** : Toujours au même endroit de la vue de la page, même si on fait défiler la page.
 - **Absolue**: Pour sortir du flux normal, *fixed* par rapport à sa première balise mère à position non *static*
- **Sticky**: mélange entre *relative* et *fixed*. Relative mais bascule à fixed à partir d'une valeur particulière du défilement de la page (pas encore supporté par tous les navigateurs)

Démo w3schools

Dépassement de contenu

- Overflow : dépassement du contenu d'un élément dans son bloc.
 - Essayez ici : <https://developer.mozilla.org/fr/docs/Web/CSS/overflow>
- Quelques valeurs:
 - `visible` (par défaut) : le contenu n'est pas rogné et déborde
 - `hidden` : le contenu est rogné
 - `scroll` : le contenu est rogné mais accessible avec une scrollbar
 - `auto` : le navigateur décide.



Michaelmas term lately over,
and the Lord Chancellor
sitting in Lincoln's Inn Hall.
Implacable November
weather. As much mud in the
streets as if the waters had
but newly retired from the
face of the earth.

Visible

Gestion du texte

- color
- font-family (pour utiliser différentes polices)
- font-size
- font-weight
- font-style (pour, par exemple, mettre en italique)
- line-height
- text-align
- text-decoration (pour ajouter des lignes dessous, dessus et en travers du texte)
- text-transform (pour modifier la casse du texte)
- text-shadow

Pseudo-classes des liens

- Les **pseudo-classes** constituent des classes prédéfinies en CSS.
- Les pseudo-classes de liens permettent de changer l'affichage des liens dans une page web.
- Elles doivent être déclarées dans cet ordre:
 1. link `a:link {} /* lien non-visité */`
 2. visited `a:visited { color: purple; } /* lien visité */`
 3. focus `a:focus { color: green; } /* lien visé */`
 4. hover `a:hover { color: green; } /* lien survolé */`
 5. active `a:active { color: red; } /* lien activé */`

https://www.w3schools.com/Css/css_link.asp

Pseudo-classes : raccourcis pour la selection

- **:first-child** – premier enfant d'un élément parent (e.g., `div p:first-child`)
- **:last-child** – dernier enfant d'un élément parent (e.g., `div p:last-child`)
- **:only-child** – enfant lorsqu'il est unique
- **:empty** – élément sans enfant (e.g., `p:empty`)