

# HTML

Pour présenter des données sur le web

Note : un serveur web ne fournit pas seulement des pages « html »

# Fichier html5 (minimum)

```
<!doctype html>  
<html lang="fr">
```

Précise le type de document

Élément « racine » (root)

```
  <head>  
    <meta charset="utf-8" />  
    <title>du texte</title>  
  </head>
```

Entête du document qui contient :

- l'encodage de la page (optionnel)
- Le titre du document (<title>) (obligatoire)

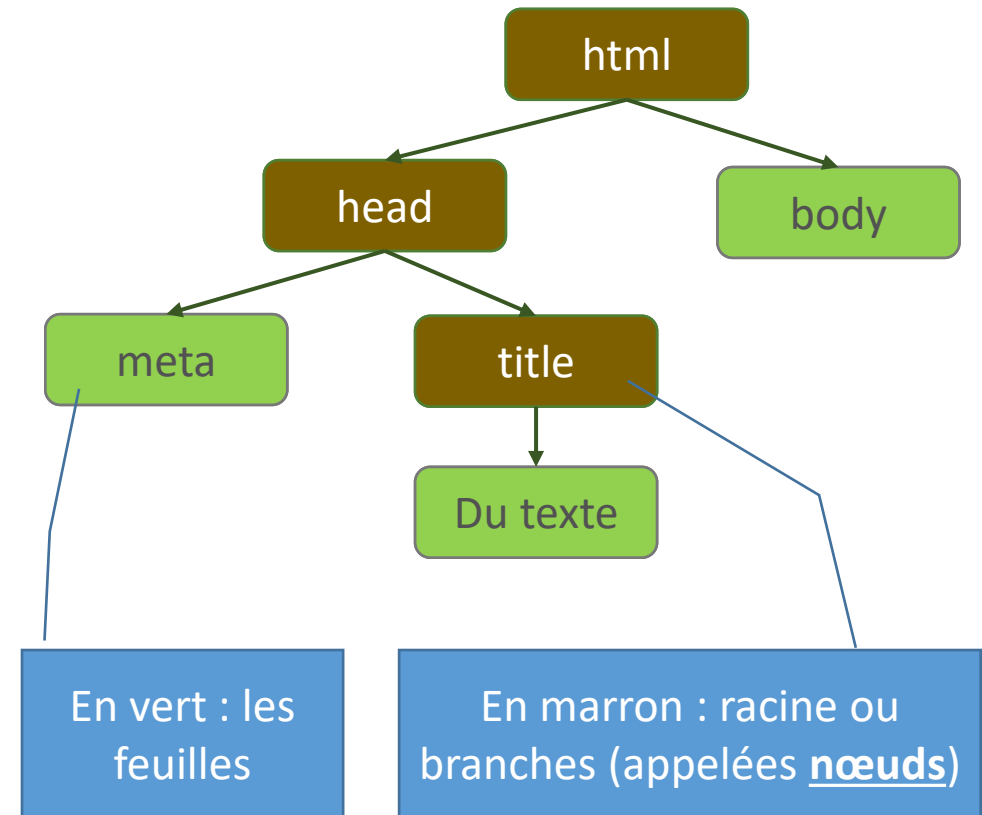
```
  <body>  
  
  </body>
```

Le « corps » du document (ce qu'il contient)

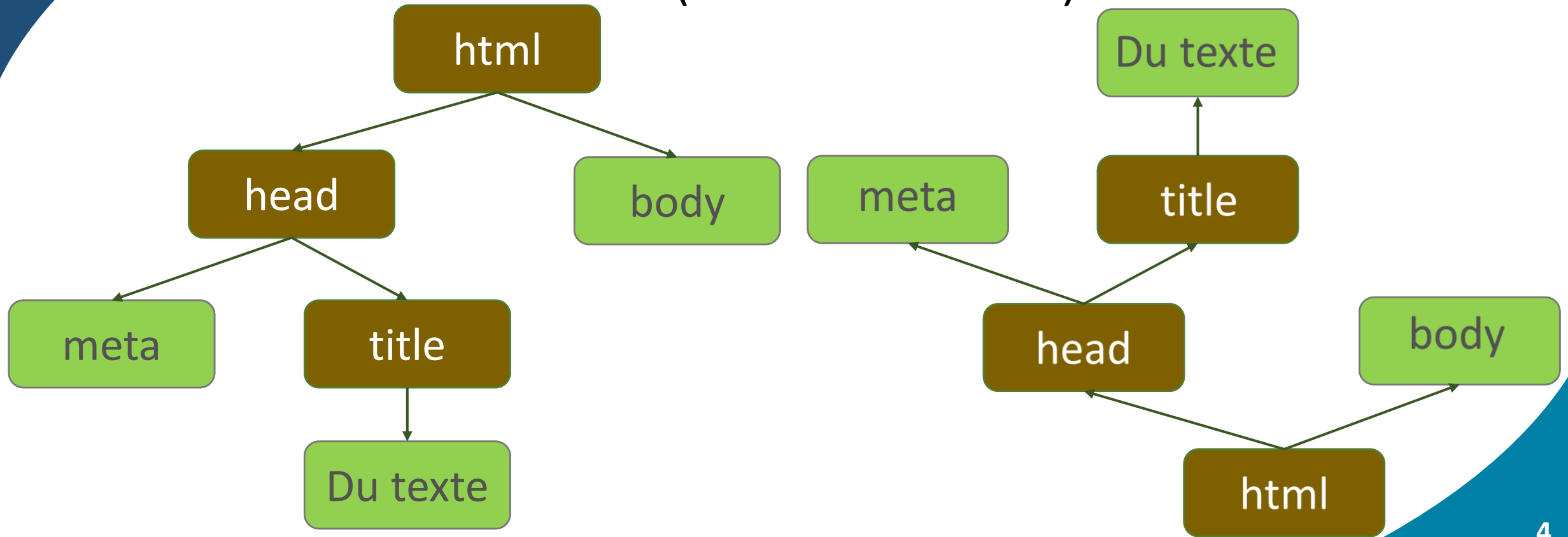
```
</html>
```

# Un document = un arbre ?

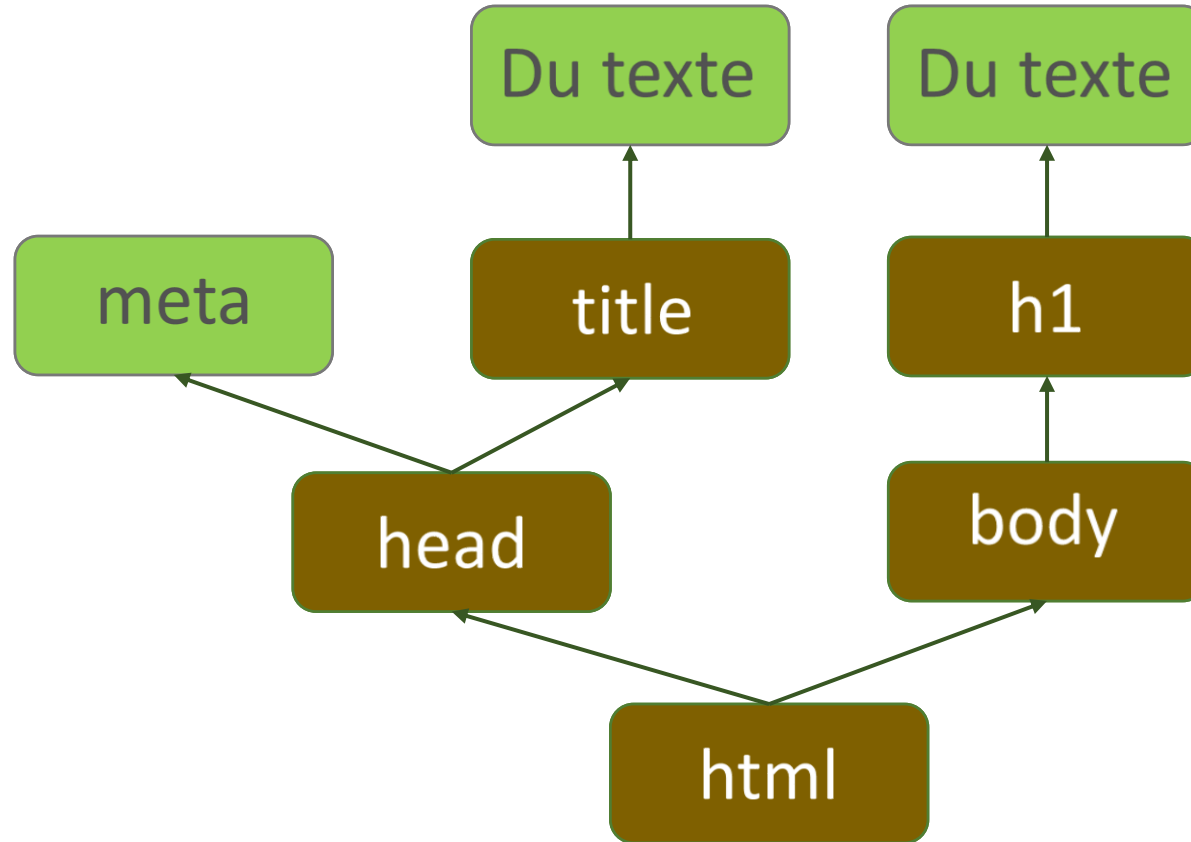
- De la structure minimale, on retient que :
  - La balise html contient
    - Une balise head, qui contient
      - Éventuellement une balise méta (pour l'encodage)
      - Une balise title qui contient
        - Le titre (du texte)
    - Une balise body vide, mais qui peut contenir...
- Balise <html> = Racine (et tronc)  
= *Root*
  - **Nœud** = **branche** = balise « qui contient » = *Node*
  - **Feuille** = texte ou balise « qui ne contient rien d'autre » = *Leaf*



En informatique, les arbres ont souvent leur racine (et une seule) en haut...



Une feuille (si sa nature lui permet)  
peut devenir une branche (un nœud)



# Exemple : « le dormeur du val » de Rimbaud

## Texte visualisé dans le navigateur

### Le dormeur du val

C'est un trou de verdure où chante une rivière  
Accrochant follement aux herbes des haillons  
D'argent ; où le soleil, de la montagne fière,  
Luit : c'est un petit val qui mousse de rayons.

Un soldat jeune, bouche ouverte, tête nue,  
Et la nuque baignant dans le frais cresson bleu,  
Dort ; il est étendu dans l'herbe, sous la nue,  
Pâle dans son lit vert où la lumière pleut.

...

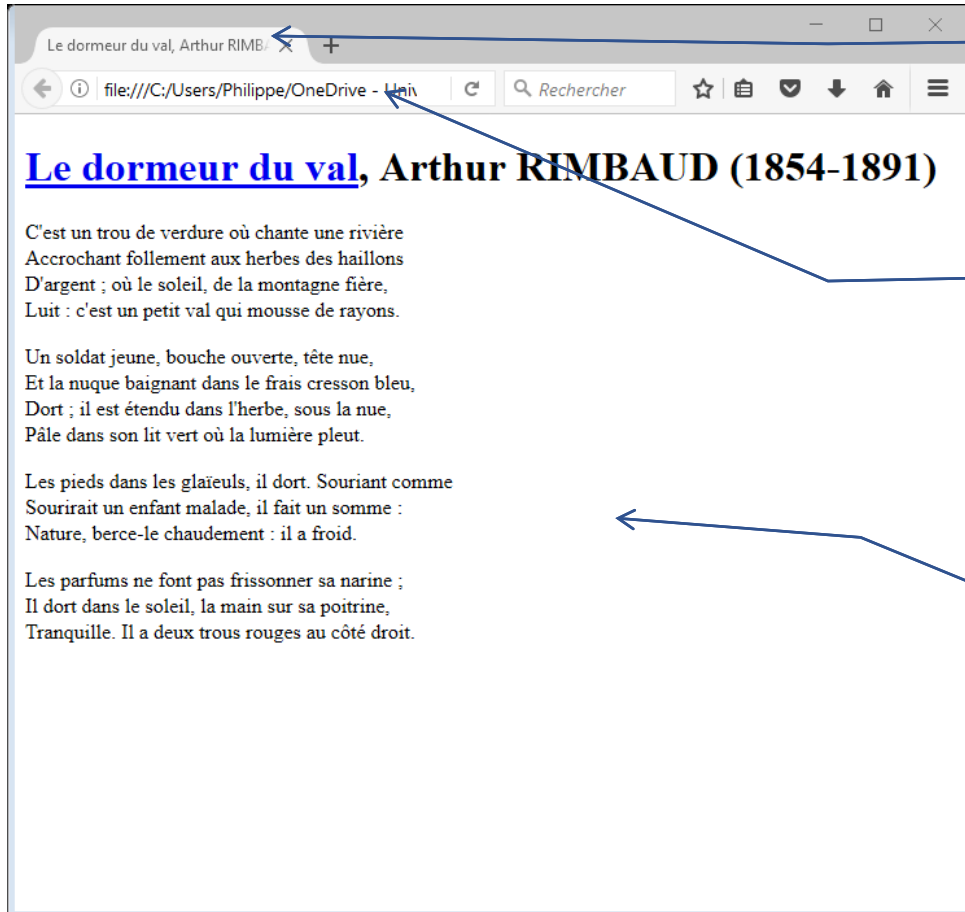
## Fichier reçu par le navigateur

```
<!doctype html><html>
<head>  <meta charset="utf-8" />
        <title>le dormeur du val</title>
</head>
<body>
<h1>Le dormeur du val</h1>
<p>C'est un trou de verdure où chante une rivière<br />
Accrochant follement      aux herbes des haillons<br />
D'argent ;      où le soleil,  de la montagne fière, <br />
Luit : c'est un petit val qui mousse de rayons.</p>
<p>Un soldat jeune, bouche ouverte, tête nue, <br /> Et la
nuque baignant dans le frais cresson bleu, <br /> Dort ; il est
étendu dans l'herbe, sous la nue, <br /> Pâle dans son lit
vert où la lumière pleut.</p>
<p>...</p>
</body></html>
```

Arthur RIMBAUD (1854-1891)

Philippe Rénévier-Gonin

# Exemple de document : affichage



Onglet du document : comporte le titre du document

Adresse du document(ici un fichier ouvert directement depuis l'ordinateur, autrement dit ouvert « localement »)

Affichage « principal » du document, appelé le « corps » du document (ce qu'il contient)

# Exemple : le code source (mal indenté)

```

<!doctype html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Le dormeur du val, Arthur RIMBAUD (1854-1891)</title>
</head>
<body>
<h1><a
href="http://poesie.webnet.fr/lesgrandsclassiques/poemes/arthur_rimbaud/le_dormeur_du_val.html">Le dormeur du val</a>, Arthur RIMBAUD (1854-1891)</h1>
<p>C'est un trou de verdure où chante une rivière<br />
  Accrochant follement aux herbes des haillons<br />
  D'argent ; où le soleil, de la montagne fière, <br />
  Luit : c'est un petit val qui mousse de rayons.</p>

<p>Un soldat jeune, bouche ouverte, tête nue, <br />
  Et la nuque baignant dans le frais cresson bleu, <br />
  Dort ; il est étendu dans l'herbe, sous la nue, <br />
  Pâle dans son lit vert où la lumière pleut.</p>

<p>Les pieds dans les glaïeuls, il dort. Souriant comme<br />
  Sourirait un enfant malade, il fait un somme :<br />
  Nature, berce-le chaudement : il a froid.</p>

<p>Les parfums ne font pas frissonner sa narine ; <br />Il
dort dans le soleil,

  la main sur sa poitrine, <br />
  Tranquille. Il a deux trous rouges au côté droit.</p>

</body>
</html>

```

Entêtes du document (pas directement visibles)

Titre du document (visible sur l'onglet)

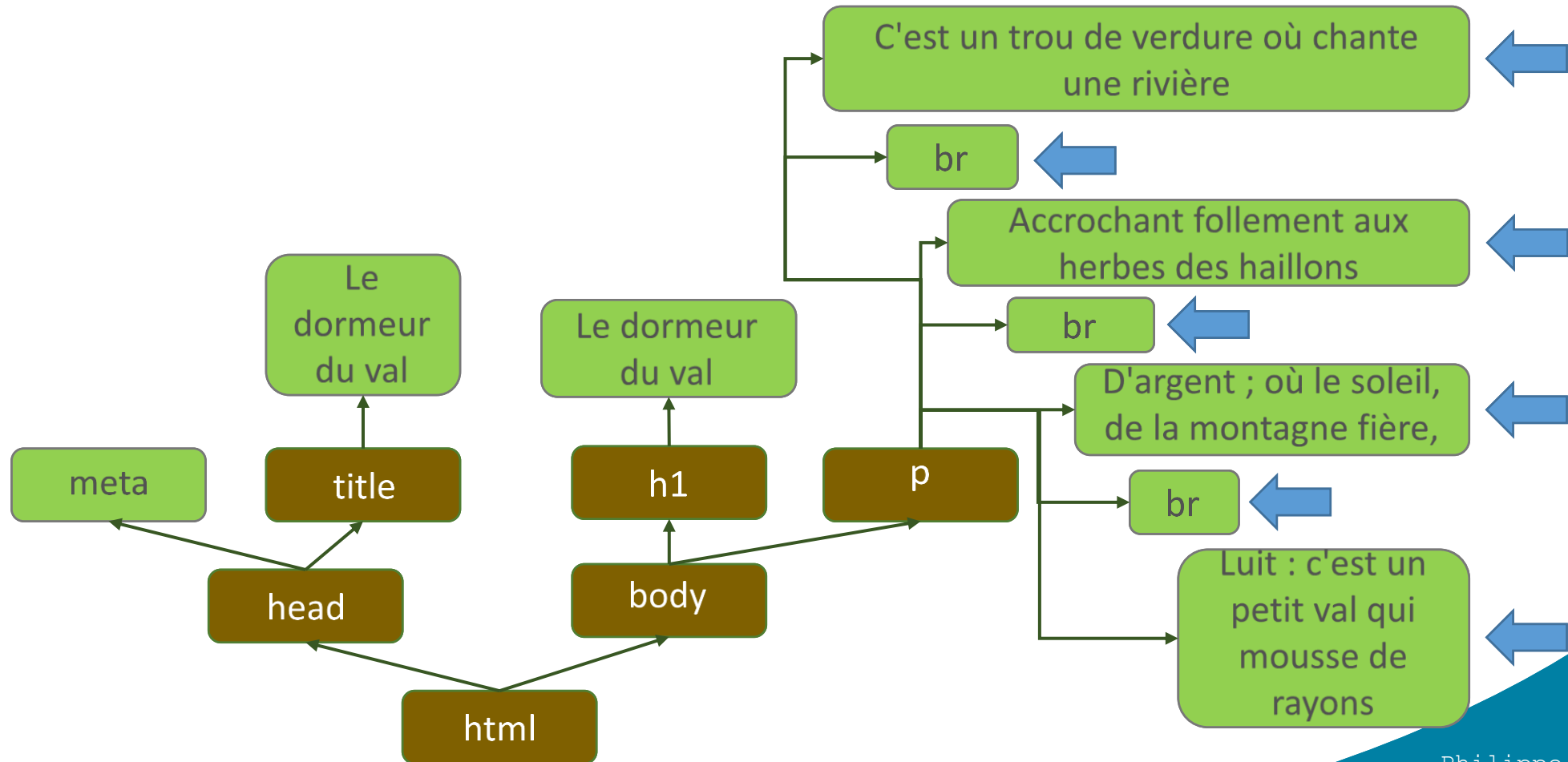
Corps du document (ce qu'il contient)

Exemple de balise « ouvrante », marquant le début d'un paragraphe (ici une strophe)

Exemple de balise « fermante », marquant la fin d'un paragraphe (ici une strophe)



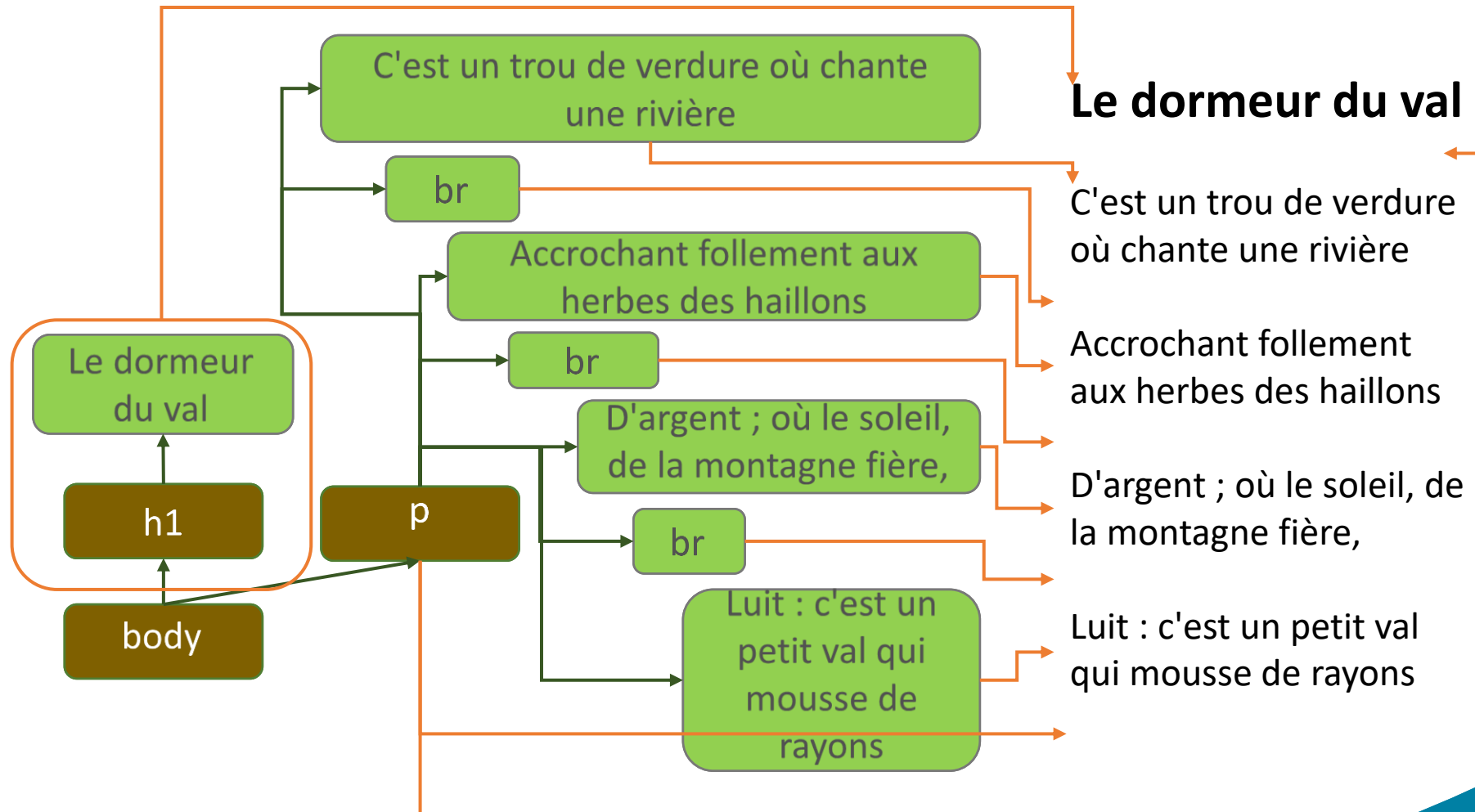
Avec la 1<sup>re</sup> strophe du « dormeur du val » du fichier reçu par l'ordinateur



# Arbre : représentation et structure

- L'arbre est une représentation pour un humain
- La structure est utilisée par les programmes (navigateur)
  - Pour rechercher du contenu
  - Calculer le rendu
- Pour du html sans css, le rendu se fait en lisant l'arbre (racine en bas) de bas en haut et de gauche à droite
  - On finit une branche avant de passer à la suivante (parcours en profondeur)

# De l'arbre vers le rendu (linéaire)



# Le langage html, des balises

- Structuration d'un fichier pour en décrire le contenu
  - Délimitation de « partie » du document par des balises
  - Utilisation par le navigateur (ou un programme) pour reconnaître ces parties d'un document
- Une balise
  - une instruction de structuration
  - contient des textes et/ou d'autres balises autorisées
- Définition d'une balise
  - Un nom : par exemple « **p** » pour un paragraphe
  - Balise ouvrante : une syntaxe pour marqué le début de cette partie :  
`<nom_de_la_balise>` par exemple `<p>`
  - Balise fermante : une syntaxe pour marquer la fin de cette partie :  
`</nom_de_la_balise>` par exemple `</p>`

# Rôle d'une balise

- Définir le rôle d'une partie du document au sein de ce document
- Exemple
  - Un paragraphe : balise <p>
  - Un titre (le plus important) : balise <h1>
- balise = étiquette (*tag*) apportant des informations sur une partie du document
  - Cette partie est délimitée par la balise ouvrante et la balise fermante
- À éviter : utilisation de balise pour l'apparence (gras, italique, etc.)
  - C'est le rôle du CSS

# Définition du html

- C'est défini par le W3C : <https://www.w3.org>
- Différentes versions de html
  - DTD = Document Type Definition
  - « grammaire de référence »
    - Définition de toutes les balises existantes
    - Définition de leurs contenus possibles
- Version utilisée pour ce cours : html5
  - <https://html.spec.whatwg.org/>
- **Outil de validation** : <http://validator.w3.org>
  - Pour savoir si le document respecte bien « la » grammaire

# Interprétation du html

Indique le début du titre (affichage plus gros, en gras)

Indique le début d'un paragraphe

Les espace, tabulations et retours à la ligne non pris en compte (vus comme un seul espace)

```
<h1>Le dormeur du val</h1>
<p>C'est un trou de verdure où chante une rivière<br />
Accrochant follement aux herbes des haillons<br />
D'argent ; où le soleil, de la montagne fière, <br />
Luit : c'est un petit val qui mousse de rayons.</p>
<p>Un soldat jeune, bouche ouverte, tête nue, <br />
Et la nuque baignant dans le frais cresson bleu, <br />
Dort ; il est étendu dans l'herbe, sous la nue, <br />
Pâle dans son lit vert où la lumière pleut.</p>
<p>Les pieds dans les glaïeuls, il dort. Souriant comme<br />
Sourirait un enfant malade, il fait un somme :<br />
Nature, berce-le chaudement : il a froid.</p>
<p>Les parfums ne font pas frissonner sa narine ;
<br />Il dort dans le soleil,

la main sur sa poitrine, <br />
Tranquille. Il a deux trous rouges au côté droit.</p>
```

Indique la fin du titre

Indique la fin d'un paragraphe

Texte du deuxième paragraphe

Indique un saut de ligne

# Éléments recherchés et interprétés par les navigateurs dans le code source

- Interprétés = non affichés directement, mais transformés
- Éléments recherchés :
  - `<nom_de_la_balise> ... </nom_de_la_balise>`
  - `<nom_de_la_balise />` (cas des balises sans textes)
- Texte  
*les espaces, tabulations et retours à la ligne consécutifs sont considérés comme un seul espace*
- `&caractères_nommés;`
  - Format : `&nom;` remplacé par un caractère donné
  - Des caractères particuliers
    - `&amp;` pour le caractère « & »
    - `&lt;` pour le caractère « < »
    - `&gt;` pour le caractère « > »
    - `&aelig;` pour le caractère « æ »
    - Etc.
  - <https://www.w3.org/TR/html5/syntax.html#named-character-references>



# Attributs/Propriétés d'une balise

- Précise la nature ou le fonctionnement de la balise
  - Certaines propriétés sont obligatoires
  - Propres à chaque balise
- Exemple:
  - `<h1 id="titrePrincipal">Mon Document</h1>`
  - Conseil : valeur entre guillemet
  - Pour les attributs « boolean » (juste la présence), conseil : `checked="checked"`
- Propriétés « globales » (quasiment communes à toutes les balises)
  - `id` : pour donner un identificateur unique à la balise/élément
  - `class` : pour attribuer une famille de style (css)
  - `style` (*inline style information*) : pour écrire du css spécifique à cette balise
  - `title` (*element title*) : élément d'information (visible souris immobile)
  - `lang` : pour préciser la langue utilisée
  - Etc. <https://www.w3.org/TR/html5/dom.html#global-attributes>

# Structuration : emboitement ou suite

- Emboitement : relation parent-enfant  
Une balise peut être à l'intérieur d'une balise
  - La balise englobante est dite « parent »
  - La balise englobée est dite « enfant »
  - Exemple
    - `<head>` est le parent de `<meta>` et `<title>`
    - `<meta>` et `<title>` sont deux enfants de `<head>`
- Suite : relation de fratrie  
Une balise peut suivre ou précéder une autre balise
  - Elles sont dites appartenir à la même fratrie
  - Exemple
    - `<meta>` et `<title>` sont dans la même fratrie
    - `<head>` et `<body>` sont dans la même fratrie
- Un document html commence toujours par la balise `<html>`

```
<!doctype html>
<html lang="fr" >

  <head>
    <meta charset="utf-8" />
    <title>du texte</title>
  </head>

  <body>

  </body>

</html>
```

The diagram shows the HTML code with green boxes highlighting the nested structure. A large box encloses the entire document. Inside it, a box for `<head>` contains `<meta>` and `<title>`. Another box for `<body>` is shown below. A bracket on the left side groups `<head>` and `<body>` as siblings within the `<html>` container.

# Quelle(s) balise(s) peut contenir une balise ?

- Tout est défini dans le doctype : <https://www.w3.org/TR/html5/>
- Pour chaque balise, une description
  - Exemple : `<html>`
  - Ce que peut contenir la balise
  - Où elle peut être placée
  - À quelle(s) catégorie(s) elle appartient (permet de savoir où elle peut être placée)
  - Ses attributs
  - Etc.

## 4.1.1 The `html` element

### Categories:

None.

### Contexts in which this element can be used:

As the root element of a document.

Wherever a subdocument fragment is allowed in a compound document.

### Content model:

A `head` element followed by a `body` element.

### Content attributes:

Global attributes

# Quelle(s) balise(s) peut contenir une <ul> ?

## Présentation

Le département informatique de la faculté des sciences propose une licence en informatique (campus de Valrose, Nice), une formation MAGE (L3 et Master, campus de Sophia Antipolis) et des Master...

- [Adresses et contacts](#)
- [Informations pratiques](#)
- [Annuaire](#)
- [Publications pédagogiques](#)

## 4.4.6 The `ul` element

### Categories:

[Flow content](#).

If the element's children include at least one `li` element: [Palpable content](#)

### Contexts in which this element can be used:

Where [flow content](#) is expected.

### Content model:

Zero or more `li` and [script-supporting](#) elements.

### Content attributes:

[Global attributes](#)

- Exemple de <ul> : une liste
- Une <ul> peut contenir
  - Rien (« Zero »)
  - Un ou plusieurs <li>
  - Des éléments d'insertion de script (typiquement la balise Script)
- D'autres informations présentes
  - Ses catégories
  - Où une balise <ul> peut être utilisée
  - Les attributs
  - Etc.

# Pourquoi bien fermer les balises ?

- Dans les spécifications, certaines balises peuvent ne pas être fermées
- Mais que se passe-t-il si une balise « fermante » est oubliée ?

# Pourquoi bien fermer les balises ?

- Dans les spécifications, certaines balises peuvent ne pas être fermées
- Mais que se passe-t-il si une balise « fermante » est oubliée ?
  - Le navigateur ouvre tout de même le document
  - Le résultat n'est pas (toujours) celui attendu
- Simplification : toujours fermer les balises ouvertes
  - Avant de passer à une autre balise
  - Ou avant de fermer une balise « parent »
  - Ou, dans le cas de `</html>`, pour clore le document
- En bonus, utile pour des traitements automatiques des documents html

# Cas des balises « vides »

- ne contiennent pas de texte
- fermées aussitôt ouvertes
- Exemples :

- `<meta>`
- `<br>`
- `<img>`

- ~~`<meta> </meta>`~~

- Notation spécifique pour ces balises « vides » (non obligatoire) : `< />`
  - `<meta charset="utf-8" />`
  - `<br />`
  - ``
  - Avantage : compatibilité avec XML (et traitement informatique des pages web)
  - Inconvénient : il faut être plus rigoureux (en mettant bien les " par exemple)

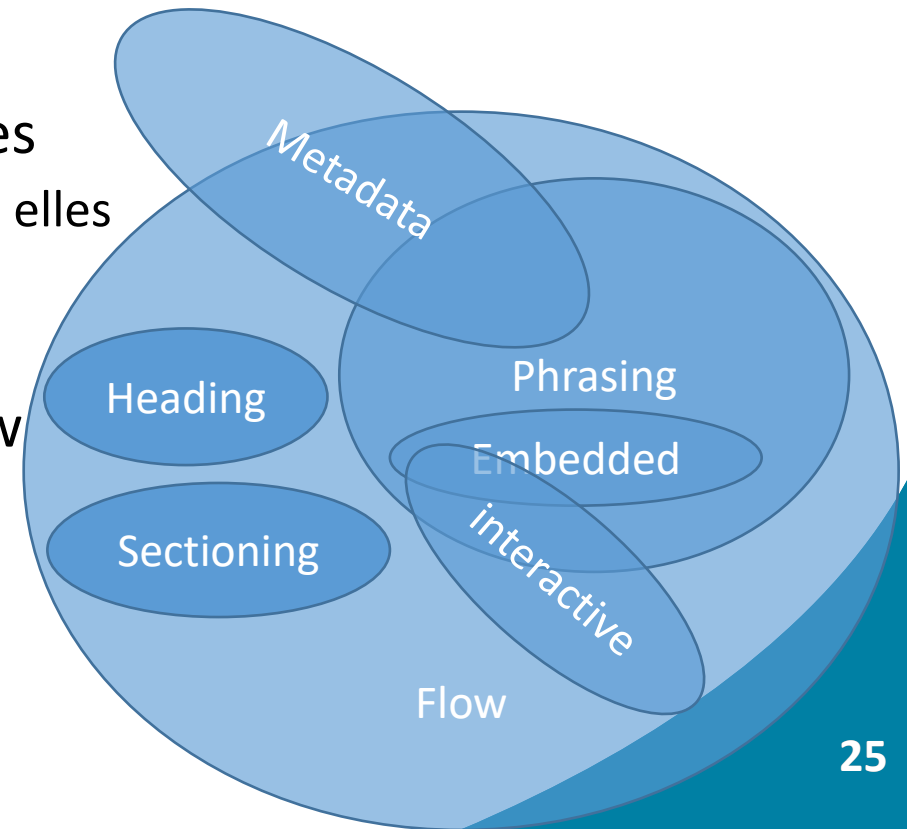
# Fonctions des balises

- Chaque balise a une fonction clairement identifiée
  - Gestion des pages HTML
    - Découpage du fichier, entête, corps
    - Informations pour le navigateur, métadonnées, titre de la page
    - Récupération des éléments complémentaires, feuilles de style, scripts, etc.
  - Structuration de l'information
    - Découpage en sections, articles, paragraphes, listes, etc.
    - Hiérarchisation du contenu
    - Mise en place de la navigation
  - Présentation de l'information
    - Insertion d'éléments extérieurs, les images, la vidéo, les scripts, etc.
    - Annotation sémantique des éléments textuels
    - Mise en place de l'interaction, formulaires de saisie



# Catégories de balises

- HTML5 définit plusieurs catégories de balises suivant leur fonction ou leur contenu
  - Une balise peut relever de 0 à plusieurs catégories
    - `html`, `head` et `body` ne relèvent d'aucune catégorie, elles sont la base de la structure du fichier, pas du contenu
  - Balises des métadonnées - Metadata
  - Balises qui manipulent le flot d'information - Flow
    - Balises de titre - Heading
    - Balises de structuration - Sectioning
    - Balises de contenu - Phrasing
    - Balises d'insertion d'éléments extérieurs - Embedded
    - Balises d'interaction – Interactive



# Tour d'horizon des Balises HTML (1/4)

- La balise racine (root) : **<html>**
- Les métadonnées
  - **<head>** : entêtes du document
  - **<title>** : titre du document (non affiché sur la page)
  - **<link />** : pour lier le document à des ressources, comme le css
  - **<meta />**, **<style>**, **<base />**
- Les sections
  - **<body>** : contenu de la page
  - **<article>**, **<section>** : pour découper la page en plus petite partie
  - **<nav>** : pour une partie « navigation » ; **<aside>** pour les apartés (les encarts)
  - **<h1> ... <h6>** : les titres (h1 > h2 > ... > h6)
  - **<header>**, **<footer>** : entêtes et « pieds de page » de section/article/page
  - **<address>**

## Tour d'horizon des Balises HTML (2/4)

- Les regroupements (Grouping content)
  - **Paragraphe : <p>**
    - **<pre>** : paragraphe « préformaté » avec un police type « courrier », toutes les espaces, retours à ligne, etc. sont visibles
  - **Liste :**
    - **<ol>, <ul>** : liste ordonnée (1. 2. 3. etc.) et liste « à puce » (comme celle-ci)
    - **<li>** : un élément de la liste (liste item)
    - **<dl>, <dt>, <dd>** : pour faire des listes de définitions (un glossaire, un dictionnaire, etc.)
  - **<figure>, <figurecaption>** : pour insérer des illustrations
  - **<div>** : une section sans « rôle » particulier
  - **<main>** : partie principale de la page
  - **<hr />** (ligne horizontale = une séparation), **<blockquote>** (pour une citation)

## Tour d'horizon des Balises HTML (3/4)

- Les éléments de phrases (text-level semantics)
  - **<a>** : lien
  - **<br />** : retour à la ligne
  - **<em>**, **<strong>**, **<cite>**, **<q>**, **<dfn>**, **<abbr>**, **<time>**, **<code>**, **<var>**, **<samp>**, **<sup>**, **<sub>**, **<mark>**
  - **<span>**
  - **<s>**, **<wbr>**
  - Etc.
- Edition: **<del>**, **<ins>**

# Tour d'horizon des Balises HTML (4/4)

- Pour inclure des éléments extérieurs (embedded content)
  - **<img />** (**<map>** , **<aera />**) : inclure une image dans le document
  - **<video>**, **<audio>**, etc. : inclure une vidéo ou un son
  - **<iframe>**, **<object>**, etc. : inclure un document dans le document (les deux restent séparés)
- Les Tableaux (tabular data)
  - **<table>**, etc. : pour faire des tables/tableaux
- Les Formulaires (forms)
  - **<form>**, **<input>**, etc.
- Les éléments liés à JavaScript (scripting)
  - **<script>** : pour ajouter du javascript
  - **<canvas>** : un « lieu » pour dessiner par programmation javascript
  - Etc.
- Etc.

# Comment faire un document html ?

- On commence par la dtd suivi de la balise <html> : définition du document
  - → Elle nécessite une <head>, un <body>
- <head>: partie comprenant les entêtes et meta-données du document
  - Un <title> ... </title> obligatoire
  - La balise « meta » est conseillée
  - Les balises « script », « style », etc. sont optionnelles
- <title> : titre du document
  - Titre de la page (en haut du navigateur)
  - Contient du texte (non vide)
- <body> : corps du document
  - Contient le document Partie centrale du navigateur
  - Attribut spécifique : onload, onunload, etc.
  - Contenu : « flow content »

# Différentes sortes de contenu

- <http://www.w3.org/TR/html5/images/content-venn.svg>
- [Metadata content](#) "base link meta noscript script style template title"
- [Flow content](#) "a abbr address area (if descendant of a map) article aside audio b bdi bdo blockquote br button canvas cite code data datalist del dfn div dl em embed fieldset figure footer form h1 h2 h3 h4 h5 h6 header hr i iframe img input ins kbd keygen label main map mark math meter nav noscript object ol output p pre progress q ruby s samp script section select small span strong sub sup svg table template textarea time u ul var video wbr Text"
- [Sectioning content](#) article aside nav section
- [Heading content](#) h1 à h6
- [Phrasing content](#) "a abbr area (if descendant of a map) audio b bdi bdo br button canvas cite code data datalist del dfn em embed i iframe img input ins kbd keygen label map mark math meter noscript object output progress q ruby s samp script select small span strong sub sup svg template textarea time u var video wbr Text"
- [Embedded content](#) "audio canvas embed iframe img math object svg video"
- [Interactive content](#) "a audio (if controls present) button embed iframe img (if usemap attribute present) input (if type not hidden) keygen label object (if usemap attribute present) select textarea video (if controls present)"
- [Palpable content](#) Éléments visibles (avec du contenu non vide, non caché)
- [Script-supporting elements](#)
- [Transparent content](#) Le contenu possible est le même que celui pour l'élément parent

# Zoom (détails) sur quelques balises

Liens (<a>)

Sections et structures (<header>, <section>, etc.)

Regroupements (<p>, <li>, etc.)

Images (<img>)

Morceaux de phrases, mise en valeurs de textes (<em>, <strong>, etc.)



# <a>, tout simplement

- Établir un lien vers un autre document
- Faire la différence entre le texte du lien (que l'on lit) et la cible du lien (là où on va)
- Texte lisible = texte entre la balise ouvrante et la balise fermante
- Cible = attribut href = désigne une page par son URL  
(*Uniform Resource Locator*)
  - *Si href absent, alors balise sans effet*
- `<a href="https://html.spec.whatwg.org/images/content-venn.svg">aperçu des balises html et leur regroupement</a>`
- `<a href="correction.html">la correction aux exercices</a>`

# <a>, tout simplement

- Établir un lien vers un autre document
- Faire la différence entre le texte du lien (que l'on lit) et la cible du lien (là où on va)
- Texte lisible = texte entre la balise ouvrante et la balise fermante
- Cible = attribut href = désigne une page par son URL  
(*Uniform Resource Locator*)
  - *Si href absent, alors balise sans effet*
- `<a href="https://html.spec.whatwg.org/images/content-venn.svg">aperçu des balises html et leur regroupement</a>`
- `<a href="correction.html">la correction aux exercices</a>`

# <a>, tout simplement

- Établir un lien vers un autre document
- Faire la différence entre le texte du lien (que l'on lit) et la cible du lien (là où on va)
- Texte lisible = texte entre la balise ouvrante et la balise fermante
- Cible = attribut href = désigne une page par son URL  
(Uniform Resource Locator)
  - *Si href absent, alors balise sans effet*
- `<a href="https://html.spec.whatwg.org/images/content-venn.svg">aperçu des balises html et leur regroupement</a>`
- `<a href="correction.html">la correction aux exercices</a>`

# Liens absolus

- Adresse complète
  - Équivalent de « Lycée Polyvalent ALGOUD-LAFFEMAS 37-39 rue Barthélémy de Laffemas - BP 26 26901 VALENCE cedex 9 »
- Généralement pour indiquer une ressource ou une référence externe
- URL absolue
  - Protocole:// : pour indiquer au navigateur comment communiquer avec le site web (serveur). Généralement http:// ou https://
  - Hôte : nom du site web (c.f. DNS) ou adresse IP
  - [:Port] : dépend de la configuration du site/serveur web (facultatif)
  - Chemin : indique dans quel dossier est le document ou la ressource (« / » en séparation)
  - Nom : indique le nom du fichier contenant le document ou la ressource (.html, .png, etc.)
- Exemple :  
`https://html.spec.whatwg.org/images/content-venn.svg`
- *url rewriting*
  - Configuration du site/serveur web pour ré-interpréter (et traduire) les URL (l'url ne contient plus un chemin de fichier mais et un chemin « logique »).

# Liens absolus

- Adresse complète

- Équivalent de « Lycée Polyvalent ALGOUD-LAFFEMAS 37-39 rue Barthélémy de Laffemas - BP 26 26901 VALENCE cedex 9 »

- Généralement pour indiquer une ressource ou une référence externe

- URL absolue

- Protocole:// : pour indiquer au navigateur comment communiquer avec le site web (serveur). Généralement http:// ou https://
  - Hôte : nom du site web (c.f. DNS) ou adresse IP
  - [:Port] : dépend de la configuration du site/serveur web (facultatif)
  - Chemin : indique dans quel dossier est le document ou la ressource (« / » en séparation)
  - Nom : indique le nom du fichier contenant le document ou la ressource (.html, .png, etc.)

- Exemple :

`https://html.spec.whatwg.org/images/content-venn.svg`

- url rewriting

- Configuration du site/serveur web pour ré-interpréter (et traduire) les URL (l'url ne contient plus un chemin de fichier mais et un chemin « logique »).

# Liens absolus

- Adresse complète
  - Équivalent de « Lycée Polyvalent ALGOUD-LAFFEMAS 37-39 rue Barthélémy de Laffemas - BP 26 26901 VALENCE cedex 9 »
- Généralement pour indiquer une ressource ou une référence externe
- URL absolue
  - Protocole:// : pour indiquer au navigateur comment communiquer avec le site web (serveur). Généralement http:// ou https://
  - Hôte : nom du site web (c.f. DNS) ou adresse IP
  - [:Port] : dépend de la configuration du site/serveur web (facultatif)
  - Chemin : indique dans quel dossier est le document ou la ressource (« / » en séparation)
  - Nom : indique le nom du fichier contenant le document ou la ressource (.html, .png, etc.)
- Exemple :  
`https://html.spec.whatwg.org/images/content-venn.svg`
- *url rewriting*
  - Configuration du site/serveur web pour ré-interpréter (et traduire) les URL (l'url ne contient plus un chemin de fichier mais et un chemin « logique »).

# Liens absolus

- Adresse complète

- Équivalent de « Lycée Polyvalent ALGOUD-LAFFEMAS 37-39 rue Barthélémy de Laffemas - BP 26 26901 VALENCE cedex 9 »

- Généralement pour indiquer une ressource ou une référence externe

- URL absolue

- Protocole:// : pour indiquer au navigateur comment communiquer avec le site web (serveur). Généralement http:// ou https://
- Hôte : nom du site web (c.f. DNS) ou adresse IP
- [Port] : dépend de la configuration du site/serveur web (facultatif)
- Chemin : indique dans quel dossier est le document ou la ressource (« / » en séparation)
- Nom : indique le nom du fichier contenant le document ou la ressource (.html, .png, etc.)

- Exemple :

`https://html.spec.whatwg.org/images/content-venn.svg`

- *url rewriting*

- Configuration du site/serveur web pour ré-interpréter (et traduire) les URL (l'url ne contient plus un chemin de fichier mais et un chemin « logique »).

# Liens absolus

- Adresse complète
  - Équivalent de « Lycée Polyvalent ALGOUD-LAFFEMAS 37-39 rue Barthélémy de Laffemas - BP 26 26901 VALENCE cedex 9 »
- Généralement pour indiquer une ressource ou une référence externe
- URL absolue
  - Protocole:// : pour indiquer au navigateur comment communiquer avec le site web (serveur). Généralement http:// ou https://
  - Hôte : nom du site web (c.f. DNS) ou adresse IP
  - [:Port] : dépend de la configuration du site/serveur web (facultatif)
  - Chemin : indique dans quel dossier est le document ou la ressource (« / » en séparation)
  - Nom : indique le nom du fichier contenant le document ou la ressource (.html, .png, etc.)
- Exemple :  
`https://html.spec.whatwg.org/images/content-venn.svg`
- *url rewriting*
  - Configuration du site/serveur web pour ré-interpréter (et traduire) les URL (l'url ne contient plus un chemin de fichier mais et un chemin « logique »).



# Liens relatifs

- Emplacement du document par rapport au document actuel
  - Équivalent de « deux rues plus loin, au 28 »
- Pratique en cas de déplacements des fichiers
- Indiquer le chemin
  - `/` : comme séparateur
  - `..` : pour remonter d'un niveau dans la hiérarchie des dossiers
  - `nom_dossier` : pour « aller » dans le dossier `nom_dossier` (ne pas oublier les « / »)
  - `nom_fichier` : pour indiquer le document par le nom du fichier (`index.html`, `graphique.png`, etc.)
- Exemple :
  - je lis la page `http://un.site.domaine/chemin/index.html`
  - Le lien `documents/cours/html/cours1.html`
    - Cible un document qui est dans le fichier `cours1.html`
    - Ce fichier est
      - dans un dossier `html`,
      - qui est dans un dossier `cours`,
      - qui est dans un dossier `documents`
      - qui est au même niveau que la page `index.html`
    - Le lien absolu correspondant est `http://un.site.domaine/chemin/documents/cours/html/cours1.html`

# Liens internes (à la page)

- Pour aller directement à une partie du document
  - Page longue
  - Table des matièresExemple : <https://www.w3.org/TR/html5/text-level-semantic.html>
- Fonctionnement :
  - Référence : href= "(nom de la page)#id\_d\_une\_balise"
    - Le nom de la page peut être omis (on reste alors dans la même page)
  - "id\_d\_une\_balise" = la valeur d'un attribut id d'une balise de la page (n'importe laquelle, là où est le point d'intérêt)
  - Quand on clique sur le lien, on est amené directement à la balise ciblée

# Liens vers une partie d'une autre page

- Si je veux faire un lien vers la documentation html5 sur <a> :  
<https://html.spec.whatwg.org/multipage/text-level-semantics.html#the-a-element>
- Référence : href= "*URL#id\_d\_une\_balise*"
  - Fonctionne comme le lien interne à la même page,
  - Sauf que l'URL indique un autre document
  - L'url peut être absolue (avec http://...) ou relative

# Propriétés de <a>

- Contient : « phrasing content » (morceaux de phrases)
  - Sauf ce qui est interactif : un autre lien, les éléments de formulaire, l'import de ressources externes (`vidéo`, `iframe`, `embed`, etc.), une image cliquable
- Attributs spécifiques
  - `href` : la cible du lien
  - Les autres (optionnels) sont conditionnés à la présence d'un `href` comme `download` (qui force le téléchargement) ou `target` (par exemple `target="_blank"` pour ouvrir dans un nouvel onglet) ou `rel` pour décrire la relation entre le document et la cible du lien.
  - <https://html.spec.whatwg.org/multipage/links.html#attr-hyperlink-rel>

# Structurations

- `<header>` : notamment pour les menu / barres de navigation
- `<footer>` : notamment pour les pieds de page
- `<nav>` : notamment pour les barres de navigation
- `<article>` : un contenu
- `<section>` : sous partie d'un article ou regroupe d'article
- `<aside>` : un encart
- `<root>` : balise unique, pour le contenu principal du document



# Relations entre les balises de structurations

- `<article>`, `<section>`, `<aside>`, `<nav>` contiennent du Flow
  - (pas de main direct ou indirect pour `nav`, `aside`)
- `<header>` et `<footer>` contiennent du Flow sauf `<header>`, `<footer>` et `<main>`
- Une `<section>` peut avoir plusieurs `<article>` (ou `<nav>`, `<aside>`, etc.)
- Un `<article>` peut avoir plusieurs `<section>` (ou `<nav>`, `<aside>`, etc.)
- Le document (body) peut avoir des `<header>` et des `<footer>`
- `<article>`, `<section>`, `<aside>`, `<nav>` peuvent avoir des `<header>` et des `<footer>`
  - Il est conseillé de mettre un titre à chaque article (dans un header ou non)

# Les titres

- <h1> : titre(s) de plus au niveau
  - Différent de title
  - Conseillé d'en mettre 1 seul en reprenant le title (accessibilité)
- Sous titre : <h2>
- Jusqu'à 6 niveaux : <h1> à <h6>
- Contenu : phrasing
  - Conseil ergonomique : éviter de mettre des liens dans le titre
- Exemple : <http://dabblet.com/gist/c8baa26a5982ebea791b>

# Regroupement : listes

- `<ul>` (non ordonnée) ou `<ol>` (ordonnée)
  - Contiennent uniquement des (0 ou +) `<li>` (ou du script)
  - Attributs :
    - `start (ol)` : une valeur de début (un entier)
    - `reversed (ol, reversed="reversed")` : pour un décompte
    - `type (ol)` : type de numéro (decimal, lower-alpha, upper-alpha, lower-roman, upper-roman)
- `<li>` (list item)
  - « Flow »
  - Attribut (si enfant de ol) : value (une valeur)
- <http://dabblet.com/gist/be630d0fb5159ed4ad51>



# Listes de définitions

- `<dl>` (Définition list)
  - `(dt+ | dd+ )*` : 0 ou 1 ou plus couple (dt, dd, dd...) [avec au moins 1 dt et un dd]
- `<dt>` (term) ou `<dd>` (définition)
  - `<dt>` contient du « flow » sauf `<header>`, `<footer>`, du section content ni du heading content
  - `<dd>` contient du « flow »

# Regroupement : Paragraphes

- `<p>` (paragraph)
  - Paragraphe de texte
  - Contenu : phrasing
- `<pre>` (texte pré-formaté)
  - espaces et retours à la ligne conservés
  - Contenu : phrasing
- `<div>` : élément de regroupement pour le CSS ou le javascript (regroupement sans sens particulier)
  - contient du Flow

# Regroupement : figure et légende

- `<figure>` : pour insérer des figures
  - Pour regrouper illustration et légende
  - Contient :  
*Flow Content* ou  
*Flow Content* suivi d'une `<figurecaption>` ou  
`<figurecaption>` suivi de *Flow Content*
- `<figcaption>` : pour définir une légende à la figure
  - Contient : *Flow Content* (mais généralement du texte)
- <http://dabblet.com/gist/69b6633bfabfa2fa5b33>

# Balise <img> – IMAGE

- Balise pour insérer des images.
  - Différents formats (png, jpg, gif, svg, etc.), selon les navigateurs
- Balise vide, les paramètres sont des attributs
- ``
  - `src="philippe.jpg"` : adresse de l'image (adresse relative ou absolue en http://)
  - `alt="picture of philippe at 29"` : description de l'image (en remplacement)  
"The alt attribute on images is a very important accessibility attribute. Authoring useful alt attribute content requires the author to carefully consider the context in which the image appears and the function that image may have in that context."  
<https://html.spec.whatwg.org/multipage/images.html#alt>
- Note : accessibilité = rendre la page "*lisible*" pour des personnes avec un ou des handicaps

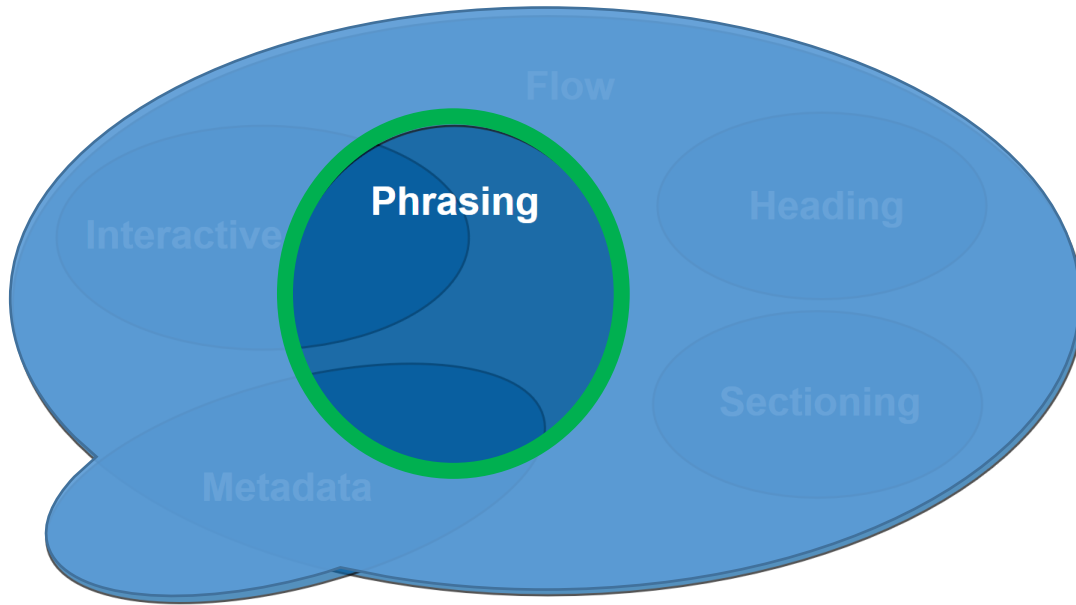
# Des liens dans des images

- `<map>`
  - Attribut name (= id)
    - Utiliser par une image avec l'attribut usemap (de `<img>`)
  - Contient des `<area>`
- `<aera>`
  - Une sorte de lien, mais défini par une forme géométrique
  - Pour la partie lien : href et si href présent, download, rel, target, type, hreflang
  - Pour la partie image : alt
    - Attribut obligatoire si l'attribut href est utilisé
  - Pour la géométrie :
    - shape : rect, circle, poly, default (= rect)
    - coords :
      - Si « rect » : 4 entiers, x0, y0, x1, y1 où (x0, y0) est le coin en haut à gauche de la zone et (x1, y1) est son coin en bas à droite
      - Si « circle » : 3 entiers x0, y0, r où (x0, y0) est le centre du cercle et r son rayon
      - Si « poly » : 2n entiers pour n points (n > 2), en une suite de (xi, yi )
      - Origine : coin en haut à gauche de l'image, dimension après affichage (donc déformation)
- <http://dabblet.com/gist/9cc7e3b0cdf7eb4c6a12>

# *Phrasing content :* « morceaux » de phrase

Description de groupes de mots, inclusion d'objets extérieurs,  
balises en lien avec la programmation, etc.

# Place des « morceaux de phrase » dans html



Source :

<https://html.spec.whatwg.org/images/content-venn.svg>

- Regroupement défini par le W3C
- *Flow content* = la plus part des balises.
  - peuvent être sous body
  - ou sous d'autres balises...
- *Phrasing content* = des éléments d'une phrase, d'une ligne ou des éléments qui seront remplacés
  - Périmètre = l'intérieur d'un paragraphe, d'un bloc.

# Les 50+ balises inclues dans « phrasing content »

Source : <https://html.spec.whatwg.org/multipage/dom.html#phrasing-content>

a	<b>Description :</b> abbr, cite, code, dfn, em, kbd, mark, q, ruby, s, samp, small, strong, sub, sup, var	
span	<b>Ligne :</b> br, wbr, bdi, bdo	<b>Données :</b> time, data
<b>Suivi :</b> del, ins		
<b>Formulaire :</b> button, datalist, input, keygen, label, meter, output, progress, select, textarea		
<b>Inclusion :</b> img   map, area   svg, math   audio, video, iframe, embed, object		
<b>Programmation :</b> script, noscript, canvas, template		
<b>À ne pas utiliser :</b> i, b, u		

## + du texte :

- le texte à proprement parlé
- vide
- les caractères espaces, retours à ligne, tabulation, etc. entre les balises

Organisation et présentation  
en 10 catégories



# Descriptions d'éléments de phrase

- `abbr`, `cite`, `code`, `dfn`, `em`, `kbd`, `mark`, `q`, `ruby`, `s`, `samp`, `small`, `strong`, `sub`, `sup`, `var`
- Décrire, caractériser le rôle d'un groupe de mots
  - C'est important : `mark`, `em`, `strong`
  - C'est de l'informatique : `code`, `kbd`, `samp`, `var`
  - C'est obsolète : `s`
  - C'est une référence bibliographique, une citation : `cite`, `q`
  - C'est un exposant, un indice : `sup`, `sub`
  - C'est une abréviation : `abbr`
  - C'est une définition : `dfn`
  - C'est une clause, une précision : `small`
  - C'est pour préciser la prononciation : `ruby`

# <span>, la balise « outil »

- Encapsulation sans sens prédéfini, pour définir une nouvelle encapsulation :
  - En combinaison avec un attribut, souvent « class », (c.f. css)
  - Terme technique : `<span class="terme_technique">...</span>`
  - Expression dans une autre langue : `<span lang="en">...</span>`
    - Abbréviation des noms des langues :
      - Cadre général : [https://fr.wikipedia.org/wiki/ISO\\_639](https://fr.wikipedia.org/wiki/ISO_639)
      - À deux lettres : [https://fr.wikipedia.org/wiki/Liste\\_des\\_codes\\_ISO\\_639-1](https://fr.wikipedia.org/wiki/Liste_des_codes_ISO_639-1)
- Élément exploité par
  - le css
  - le javascript
  - ou d'autres technologies

# Gestion des lignes

- `<br />` (line break)
  - Retour à la ligne
  - Rappel : { retour à ligne, tabulation, espace } =<sub>en html</sub> espace
- `<wbr />` : pour indiquer au navigateur où un retour à la ligne peut être fait
- pour un espace non « cassable » : `&nbsp;`;
  - (ou `&#160;` ou `&#xA0;`)
  - *non-breaking space*
- `<bdi>` : lorsque le sens de lecture/écriture n'est pas connu
- `<bdo dir="/tr (ou rtl)">` : pour force le sens de de lecture/écriture
- cas de documents multilingues

# Balises à la sémantique de niveau texte

- Les catégories
  - a
  - description d'éléments de phrase
  - span
  - gestions des lignes
  - données
- sont regroupées dans un chapitre « [text-level semantics](#) » de la documentation html5
- Utiles pour la structuration (et avec css, la mise en forme), pour les traitements automatiques (génération d'index, de glossaire,...), etc.

# Inclusion de ressources extérieures

- C.f. chapitre « [embedded content](#) » de la documentation html5
- Pour inclure des images dans le document
  - `img` : il faut préciser l'emplacement de l'image (et une description)  

```

```
  - `map, area` : pour définir des zones cliquables dans les images
- Vidéo / Musique : `audio`, `video`
- Pour intégrer des ressources externes (`object`) ou des applications–plugin externes (`embed`)
- Pour intégrer un document dans un document (`iframe`) : exemple youtube, etc.
- Pour intégrer d'autres langages à balises
  - `svg` : des images vectorielles
  - `math` : des formules mathématiques (dépend du navigateur)

# À ne pas utiliser : mise en forme à la « CSS »

- Les balises concernées sont
  - `i` pour *italic* (italique)
  - `b` pour **bold** (gras)
  - `u` pour underlined (souligné)
- Présentes dans le dans un chapitre « [text-level semantics](#) »
- Dans la documentation : conseil d'utiliser d'autres balises
  - sens trop vague,
  - utilisées pour mettre en forme
    - Mettre en italique, en gras ou souligner : **css**

# <table> - HTML

Pour faire des tableaux

# Exemple de table

- <http://dabblet.com/gist/8410318c07bfa4084b90>
- `<table>` contient, dans l'ordre,
  - `<caption>` (optionnel)
  - `<colgroup>` (optionnel)
  - `<thead>` (optionnel)
  - `<tfoot>` (optionnel)
  - `<tbody>` (1 ou plusieurs)
    - Exception : `<tobdy>` peut être optionnel si les balises précédentes sont bien fermées et que le corps du tableau commence par un `<tr>`
      - `<tr>` peut donc être directement un enfant de `<table>`
      - Dans ce cas, `<tfoot>` peut venir en dernier (mais un seul `<tfoot>` en tout)
- Le contenu est fait par ligne `<tr>` (dans `<thead>`, `<tfoot>` ou `<tbody>`)
  - Nombre de lignes = nombre de `<tr>`, attention aux attributs **rowspan** de `<td>` ou `<th>`
- Les cases par `<td>` ou `<th>`
  - Nombre de colonnes = nombre total de `<td>` et `<th>` par ligne, attention aux attributs **colspan** de `<td>` ou `<th>`



## Balise <table>

- Un seul attribut, à usage limité : border  
border="" → il ne faut pas dessiner le contour  
border="1" → il faut dessiner le contour
  - Le quadrillage se fera en css

## Balise <caption>

- Précise la légende du tableau
- Contient du Flow exception faite d'une autre <table> (directement ou indirectement)

# Parties d'un tableau

- `<thead>` (titre sur lignes « supérieures »), `<tfoot>` (bas du tableau) et `<tbody>` (bloc de données)
  - Contiennent des `<tr>` (0 ou +) ou du `<script>`
- `<tr>`
  - Contient des `<td>`, `<th>` (0 ou plus) et/ou du `<script>`

# Cases (ou cellules) d'un tableau

- `<th>` : table header
  - Contient du Flow excepté du Sectioning Content, `<header>`, `<footer>` et du Heading Content
- `<td>` : table data
  - Contient du Flow
- Attributs communs
  - `rowspan` : pour indiquer sur combien de ligne s'étend la cellule
  - `colspan` : pour indiquer sur combien de colonne s'étend la cellule
  - Attention : à bien compter les td/th sur les autres lignes / colonnes...
  - Attention : pas toujours visible sans CSS...
  - <http://dabblet.com/gist/b40a4864acb7a6aaae64>
  - headers
- Attributs propres à `<th>`
  - `scope` : row | col | rowgroup (= tbody, thead ou tfoot) | colgroup | auto
    - Pour dire à quoi s'applique le titre
    - Auto : décidé par le navigateur
  - `abbr` : pour fournir une description (en principe plus courte) du titre