

Web

HTML

Bruno BEAUFILS

2024/2025

1. Web et documents

2. Principes de HTML

3. Syntaxe HTML

4. Sémantique HTML

5. Références

6. Devoir

- ▶ Historiquement
 - ▶ un système permettant à des **logiciels** d'échanger des **documents**
 - ▶ un service de communication au dessus de l'Internet
 - ▶ interopérabilité, universalité
- ▶ Création
 - ▶ notamment par Tim BERNERS-LEE
 - ▶ CERN
- ▶ Plusieurs composants
 - ▶ protocole de communication : HTTP
 - ▶ langage de description de documents : **HTML**
- ▶ Échanges de documents nécessite
 - ▶ production
 - ▶ consultation

Création documentaire

- ▶ 2 aspects importants
 - ▶ **fond** : informations, données
 - ▶ **forme** : présentation
- ▶ Exemples
 - ▶ nombre : quantité vs notation/représentation (indo-arabes, romains, etc.)
 - ▶ roman : auteur vs typographe
- ▶ Informatique a permis de **séparer fond et forme** facilement
 - ▶ Avantages
 - ▶ **accessibilité** (handicap, automatisation, etc.)
 - ▶ **universalité** = pas lié à un seul mode de représentation
 - ▶ seules les **données** sont importantes dans les documents
 - ▶ Problèmes
 - ▶ Outils mal conçus (Word) et éducation/formation déficiente
 - ▶ Utilisateurs plus souvent habitués au **WYSIWYG** qu'au **WYSIWYM**

Web conçu pour être un système **universel** d'échange de documents

- ▶ 2 outils pour décrire les documents
 - ▶ **HTML** : description des informations
 - ▶ **CSS** : description de la mise en forme
- ▶ 1 langage de programmation pour manipuler le tout
 - ▶ Javascript

fond
forme

Création documentaire

- ▶ 2 aspects importants
 - ▶ **fond** : informations, données
 - ▶ **forme** : présentation
- ▶ Exemples
 - ▶ nombre : quantité vs notation/représentation (indo-arabes, romains, etc.)
 - ▶ roman : auteur vs typographe
- ▶ Informatique a permis de **séparer fond et forme** facilement
 - ▶ Avantages
 - ▶ **accessibilité** (handicap, automatisation, etc.)
 - ▶ **universalité** = pas lié à un seul mode de représentation
 - ▶ seules les **données** sont importantes dans les documents
 - ▶ Problèmes
 - ▶ Outils mal conçus (Word) et éducation/formation déficiente
 - ▶ Utilisateurs plus souvent habitués au **WYSIWYG** qu'au **WYSIWYM**

Web conçu pour être un système **universel** d'échange de documents

- ▶ 2 outils pour décrire les documents
 - ▶ **HTML** : description des informations
 - ▶ **CSS** : description de la mise en forme
- ▶ 1 langage de programmation pour manipuler le tout
 - ▶ Javascript

fond
forme

Création documentaire

- ▶ 2 aspects importants
 - ▶ **fond** : informations, données
 - ▶ **forme** : présentation
- ▶ Exemples
 - ▶ nombre : quantité vs notation/représentation (indo-arabes, romains, etc.)
 - ▶ roman : auteur vs typographe
- ▶ Informatique a permis de **séparer fond et forme** facilement
 - ▶ Avantages
 - ▶ **accessibilité** (handicap, automatisation, etc.)
 - ▶ **universalité** = pas lié à un seul mode de représentation
 - ▶ seules les **données** sont importantes dans les documents
 - ▶ Problèmes
 - ▶ Outils mal conçus (Word) et éducation/formation déficiente
 - ▶ Utilisateurs plus souvent habitués au **WYSIWYG** qu'au **WYSIWYM**

Web conçu pour être un système **universel** d'échange de documents

- ▶ 2 outils pour décrire les documents
 - ▶ **HTML** : description des informations
 - ▶ **CSS** : description de la mise en forme
- ▶ 1 langage de programmation pour manipuler le tout
 - ▶ Javascript

fond
forme

1. Web et documents

2. Principes de HTML

3. Syntaxe HTML

4. Sémantique HTML

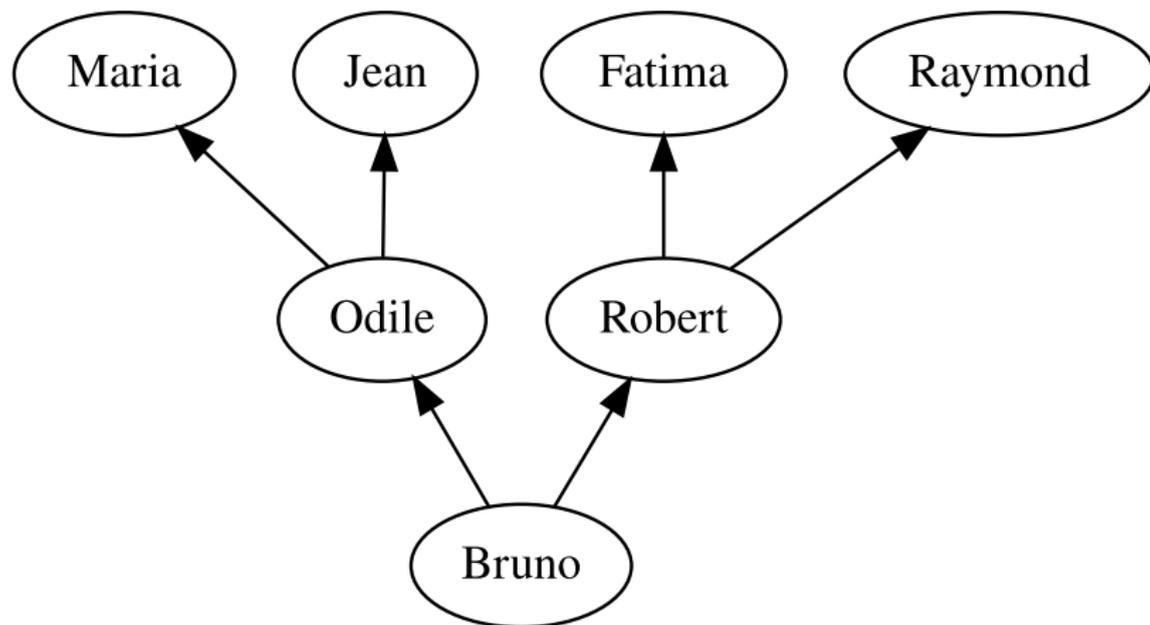
5. Références

6. Devoir

- ▶ un **graphe**
 - ▶ un ensemble de **noeuds** (*sommets*) reliés par des **arêtes**
- ▶ **acyclique**
 - ▶ pas de cycles dans les relations
- ▶ **connexe**
 - ▶ tous les noeuds sont connectés
- ▶ **orienté**
 - ▶ les arêtes ont une direction et un sens (départ et arrivée)

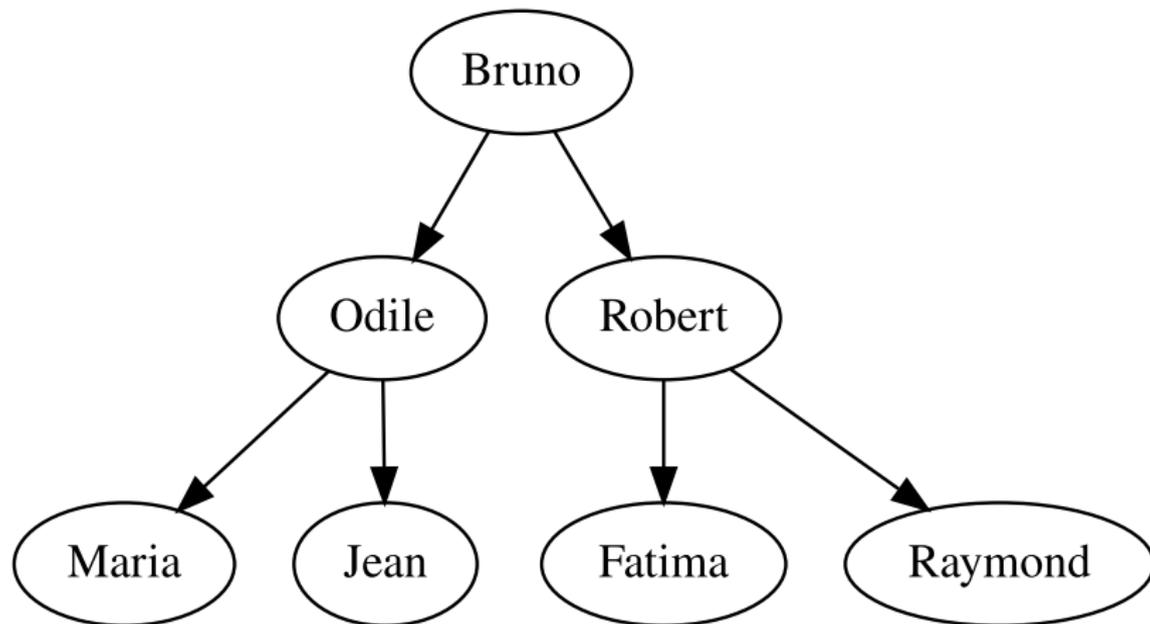
Arbres informatiques (exemple)

Arbre généalogique



Arbres informatiques (exemple)

Arbre informatique



HTML

HyperText Markup Language = Language de balisage d'hypertexte

Langage de description de document

- ▶ description du contenu (fond) pas de la mise en forme
- ▶ **données** et **structuration**

Structuration par un arbre

- ▶ structure **hiérarchique**
 - ▶ ensemble de **noeuds** relié par des **arêtes**
- ▶ chaque noeud possède un type et une valeur
 - ▶ type = **nature** de l'information contenu dans le noeud
 - ▶ valeur = **information** elle même
 - ▶ texte
 - ▶ ensemble **ordonné** d'autres noeuds
 - ▶ vide

un graphe particulier

- ▶ une **racine**
 - ▶ vocabulaire de la filiation (mère, frères, filles, ancêtres, etc.)

le noeud origine

HTML

HyperText Markup Language = Language de balisage d'hypertexte

Langage de description de document

- ▶ description du contenu (fond) pas de la mise en forme
- ▶ **données** et **structuration**

Structuration par un arbre

- ▶ structure **hiérarchique**
 - ▶ ensemble de **noeuds** relié par des **arêtes**
- ▶ chaque noeud possède un type et une valeur
 - ▶ type = **nature** de l'information contenu dans le noeud
 - ▶ valeur = **information** elle même
 - ▶ texte
 - ▶ ensemble **ordonné** d'autres noeuds
 - ▶ vide

un graphe particulier

- ▶ une **racine**
 - ▶ vocabulaire de la filiation (mère, frères, filles, ancêtres, etc.)

le noeud origine

Un langage textuel simple

- ▶ Représentation de l'arbre par du **texte simple**

- ▶ un document = un fichier texte
- ▶ une syntaxe particulière pour le texte

- ▶ Contenu

1. une ligne spécifiant le type du document
2. une **suite d'éléments** structurant le document

`<!DOCTYPE html>`

- ▶ Outils nécessaires à la création : éditeur de textes

- ▶ Multi systèmes (Windows/MacOS/Linux) :
 - ▶ bluefish,
 - ▶ Visual Studio Code,
 - ▶ Emacs
- ▶ Windows : Notepad++
- ▶ Certains aide au respect de la syntaxe
 - ▶ colorisation, visualisation arborescente, etc.

à privilégier

Un langage textuel simple

- ▶ Représentation de l'arbre par du **texte simple**

- ▶ un document = un fichier texte
- ▶ une syntaxe particulière pour le texte

- ▶ Contenu

1. une ligne spécifiant le type du document
2. une **suite d'éléments** structurant le document

`<!DOCTYPE html>`

- ▶ Outils nécessaires à la création : éditeur de textes

- ▶ Multi systèmes (Windows/MacOS/Linux) :

- ▶ bluefish,
- ▶ Visual Studio Code,
- ▶ Emacs

- ▶ Windows : Notepad++

- ▶ Certains aide au respect de la syntaxe

- ▶ colorisation, visualisation arborescente, etc.

à privilégier

Un langage textuel simple

- ▶ Représentation de l'arbre par du **texte simple**

- ▶ un document = un fichier texte
- ▶ une syntaxe particulière pour le texte

- ▶ Contenu

1. une ligne spécifiant le type du document
2. une **suite d'éléments** structurant le document

`<!DOCTYPE html>`

- ▶ Outils nécessaires à la création : éditeur de textes

- ▶ Multi systèmes (Windows/MacOS/Linux) :

- ▶ bluefish,
- ▶ Visual Studio Code,
- ▶ Emacs

- ▶ Windows : Notepad++

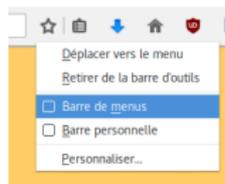
- ▶ Certains aide au respect de la syntaxe

- ▶ colorisation, visualisation arborescente, etc.

à privilégier

Exercice : préparation de l'environnement de travail

1. Créez un dossier nommé `html` sur votre Bureau
 - ▶ vous y placerez tous les fichiers de la séance
2. Démarrez Firefox
3. Personnalisez Firefox
 - ▶ Faire apparaître le menu
 - 3.1 faire apparaître le menu de personnalisation
 - 3.2 bouton droit de la souris près de la barre d'adresse
 - 3.3 sélectionner *Barre de menus*



Exercice : un premier document HTML

1. Créez un fichier nommé premier.html avec le contenu suivant

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta charset="utf-8"/>
    <title>
      Mon premier document
    </title>
  </head>
  <body>
    <p>
      Voici mon premier paragraphe
    </p>
  </body>
</html>
```

Exercice : un premier document HTML (suite)

2. Dessinez l'arbre de ce document tel que vous l'imaginez
3. Regardez votre fichier avec Firefox
 - ▶ Menu puis *Fichier* puis *Ouvrir un fichier...*
 - ▶ Accès rapide : `Control + O`
4. Regardez l'arbre du document
 - ▶ Menu puis *Outils de développement* puis *Inspecteur*
 - ▶ Accès rapide : `Shift + Control + C`

1. Web et documents

2. Principes de HTML

3. Syntaxe HTML

4. Sémantique HTML

5. Références

6. Devoir

Composants syntaxiques

▶ Élément

- ▶ la nature (**sémantique**) de la donnée
- ▶ un contenu (la donnée elle même)
- ▶ construit avec des balises

type du noeud
valeur du noeud

```
<balise> contenu </balise>
```

▶ Attribut

- ▶ informations **supplémentaires** concernant l'élément
- ▶ nature et valeur
- ▶ introduite dans la balise ouvrante

étiquette sur le noeud

```
<balise attribut="valeur">contenu</balise>
```

▶ Commentaire

- ▶ informations non utiles pour les données

```
<!-- Ceci est un commentaire sur le document -->
```

Composants syntaxiques

▶ Élément

- ▶ la nature (**sémantique**) de la donnée
- ▶ un contenu (la donnée elle même)
- ▶ construit avec des balises

type du noeud
valeur du noeud

```
<balise> contenu </balise>
```

▶ Attribut

- ▶ informations **supplémentaires** concernant l'élément
- ▶ nature et valeur
- ▶ introduite dans la balise ouvrante

étiquette sur le noeud

```
<balise attribut="valeur">contenu</balise>
```

▶ Commentaire

- ▶ informations non utiles pour les données

```
<!-- Ceci est un commentaire sur le document -->
```

Composants syntaxiques

▶ Élément

- ▶ la nature (**sémantique**) de la donnée
- ▶ un contenu (la donnée elle même)
- ▶ construit avec des balises

type du noeud
valeur du noeud

```
<balise> contenu </balise>
```

▶ Attribut

- ▶ informations **supplémentaires** concernant l'élément
- ▶ nature et valeur
- ▶ introduite dans la balise ouvrante

étiquette sur le noeud

```
<balise attribut="valeur">contenu</balise>
```

▶ Commentaire

- ▶ informations non utiles pour les données

```
<!-- Ceci est un commentaire sur le document -->
```

- ▶ Élément vide
 - ▶ pas de contenu
 - ▶ l'information **est** la balise ou ses attributs

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

- ▶ Éléments imbriqués
 - ▶ représentation arborescente d'un document
 - ▶ DOM = Modèle Objet du Document
- ▶ Exemple complet : [src/gotlib.html](#)
 - ▶ Ouvrir dans Firefox montre la mise en forme
 - ▶ Pour voir le document sans mise en forme :
 - ▶ Menu contextuel puis *Code source de la page*
 - ▶ Accès rapide : Control + U
 - ▶ Pour voir l'arbre du document :
 - ▶ Menu puis *Outils de développement* puis *Inspecteur*
 - ▶ Accès rapide : Shift + Control + C

▶ Élément vide

- ▶ pas de contenu
- ▶ l'information **est** la balise ou ses attributs

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

▶ Éléments imbriqués

- ▶ représentation arborescente d'un document
- ▶ DOM = Modèle Objet du Document

▶ Exemple complet : [src/gotlib.html](#)

- ▶ Ouvrir dans Firefox montre la mise en forme
- ▶ Pour voir le document sans mise en forme :
 - ▶ Menu contextuel puis *Code source de la page*
 - ▶ Accès rapide : Control + U
- ▶ Pour voir l'arbre du document :
 - ▶ Menu puis *Outils de développement* puis *Inspecteur*
 - ▶ Accès rapide : Shift + Control + C

- ▶ Élément vide
 - ▶ pas de contenu
 - ▶ l'information **est** la balise ou ses attributs

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

- ▶ Éléments imbriqués
 - ▶ représentation arborescente d'un document
 - ▶ DOM = Modèle Objet du Document
- ▶ Exemple complet : [src/gotlib.html](#)
 - ▶ Ouvrir dans Firefox montre la mise en forme
 - ▶ Pour voir le document sans mise en forme :
 - ▶ Menu contextuel puis *Code source de la page*
 - ▶ Accès rapide : Control + U
 - ▶ Pour voir l'arbre du document :
 - ▶ Menu puis *Outils de développement* puis *Inspecteur*
 - ▶ Accès rapide : Shift + Control + C

1. Web et documents

2. Principes de HTML

3. Syntaxe HTML

4. Sémantique HTML

5. Références

6. Devoir

Racine du document : `<html>`

contient 2 éléments seulement

- ▶ `<head>` entête du document : **caractéristiques et méta-données**
- ▶ `<body>` **contenu** principal

Racine du document : `<html>`

contient 2 éléments seulement

- ▶ `<head>` entête du document : **caractéristiques et méta-données**
- ▶ `<body>` **contenu** principal

Entête du document : <head>

contient des caractéristiques du document

- ▶ `<title>` le titre du document
- ▶ `<meta/>` des caractéristiques du document
 - ▶ par exemple l'encodage du texte

```
<meta charset="utf-8"/>
```

- ▶ `<link/>` des liens vers des documents utiles à ce document

Entête du document : <head>

contient des caractéristiques du document

- ▶ `<title>` le titre du document
- ▶ `<meta/>` des caractéristiques du document
 - ▶ par exemple l'encodage du texte

```
<meta charset="utf-8"/>
```

- ▶ `<link/>` des liens vers des documents utiles à ce document

Corps du document : <body>

contient le contenu effectif du document

▶ Éléments **structurant** :

- ▶ <h1>, <h2>, ..., <h6>,
- ▶ <article>,
- ▶ <section>,
- ▶ <aside>, <nav>,
- ▶ <header>, <main>, <footer>

▶ Éléments de **textes** :

▶ simples :

- ▶ paragraphe <p>,
- ▶ citation <blockquote>, <q>,
- ▶ rupture de paragraphe
, ligne horizontale <hr/>

▶ listes :

- ▶ non-ordonnée ,
- ▶ ordonnée ,
- ▶ de définition <dl>,
- ▶ items des listes ou <dd>, <dt>

▶ préformaté : <pre>

▶ formatage logique (*sémantique*) : , , <code>, <kbd>, etc.

Corps du document : <body> (suite)

▶ Éléments **extérieurs** :

- ▶ image ,
- ▶ ancre vers d'autres documents <a>

hyperlien

▶ Éléments **tabulaires** :

- ▶ <table>,
- ▶ <thead>, <tbody>, <tfoot>,
- ▶ <tr>,
- ▶ <td>, <th>

*parties
ligne
cellule*

▶ Éléments de **formulaire**

- ▶ <form>
- ▶ <input>, <select>, <option>, <textarea>, <button>, <datalist>, <output>

▶ Éléments de **regroupement** (*uniquement quand rien d'autre ne convient*) :

- ▶ regroupement en bloc <div>,
- ▶ regroupement en ligne

1. Web et documents

2. Principes de HTML

3. Syntaxe HTML

4. Sémantique HTML

5. Références

6. Devoir

- ▶ Mozilla Developer Network (HTML)
 - ▶ Commencer avec le web
 - ▶ **Les bases de HTML**
 - ▶ **Introduction à HTML**
 - ▶ Références HTML
- ▶ DevDocs pour HTML
- ▶ « *Stop using so many divs! An intro to semantic HTML* »

1. Web et documents

2. Principes de HTML

3. Syntaxe HTML

4. Sémantique HTML

5. Références

6. Devoir

Exercice : structuration d'une page web

1. Ecrivez un document HTML vous décrivant à la forme d'un CV
 - ▶ Nommez le `cv.html` dans votre dossier `html`
 - ▶ Il doit contenir **5 parties ordonnées** obligatoires
 - ▶ **titre** : identité et projet professionnel
 - ▶ **identité détaillée** : date de naissance, lieu de naissance email
 - ▶ **formation**
 - ▶ **expériences professionnelles**
 - ▶ **divers** : hobby, points saillants
 - ▶ D'autres données **obligatoires** doivent être présente dans le document
 - ▶ un lien vers une photo de vous
 - ▶ un lien vers vos identités sur les réseaux sociaux
2. Envoyez-moi votre travail (le fichier `cv.html`)
 - ▶ attaché à votre email
 - ▶ adresse : **bruno.beaufils@univ-lille.fr** ou **bruno@ape-cee.fr**
 - ▶ objet du mail : **[M2APE] devoir 2 Prénom NOM**
 - ▶ le **strict** respect de ces contraintes est important