

**Web**

**Bruno BEAUFILS**

**2021/2022**

# 1. Introduction

## 2. Documents numériques

HTML

## 1. Introduction

## 2. Documents numériques

HTML

# 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 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 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 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. Introduction

## 2. Documents numériques

HTML

## 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*

# Tableau

# HTML : un langage 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
- ▶ Outils nécessaires à la création : éditeur de textes
  - ▶ Multi systèmes (Windows/MacOS/Linux) :
    - ▶ bluefish,
    - ▶ Visual Studio Code,
    - ▶ Brackets,
    - ▶ Atom,
    - ▶ Emacs
  - ▶ Windows : Notepad++
  - ▶ Certains aide au respect de la syntaxe
    - ▶ colorisation, visualisation arborescente, etc.

`<!DOCTYPE html>`

*à privilégier*

# HTML : un langage 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
- ▶ Outils nécessaires à la création : éditeur de textes
  - ▶ Multi systèmes (Windows/MacOS/Linux) :
    - ▶ **bluefish**,
    - ▶ Visual Studio Code,
    - ▶ Brackets,
    - ▶ Atom,
    - ▶ Emacs
  - ▶ Windows : **Notepad++**
  - ▶ Certains aide au respect de la syntaxe
    - ▶ colorisation, visualisation arborescente, etc.

`<!DOCTYPE html>`

*à privilégier*

# HTML : un langage simple

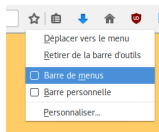
- ▶ Représentation de l'arbre par du **texte simple**
  - ▶ un document = un fichier texte
  - ▶ une syntaxe particulière pour le texte
- ▶ Contenu
  1. une ligne spécifiant le type du document
  2. une **suite d'éléments** structurant le document
- ▶ Outils nécessaires à la création : éditeur de textes
  - ▶ Multi systèmes (Windows/MacOS/Linux) :
    - ▶ **bluefish**,
    - ▶ Visual Studio Code,
    - ▶ Brackets,
    - ▶ Atom,
    - ▶ Emacs
  - ▶ Windows : Notepad++
  - ▶ Certains aide au respect de la syntaxe
    - ▶ colorisation, visualisation arborescente, etc.

`<!DOCTYPE html>`

*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*



- ▶ Paramétrez les outils de développement
  - ▶ Menu puis *Outils* puis *Développement web* puis *Outils de développement*
  - ▶ Accès rapide : `Shift + Control + I`
  - ▶ Détachez la fenêtre
  - ▶ Changer les *Options des outils*
  - ▶ Désactivez tous les *Outils de développement par défaut*

# 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`



# Tableau

# Les composants du code HTML

## ▶ 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 -->
```

# Les composants du code HTML

## ▶ 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

- ▶ 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 -->
```

# Tableau

- ▶ 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](http://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



# Tableau

# Documents

## 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

## Corps du document : `<body>`

contient le contenu effectif du document

# Documents

## 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

## Corps du document : `<body>`

contient le contenu effectif du document

# Documents

## 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

## Corps du document : `<body>`

contient le contenu effectif du document

# Tableau

- ▶ Éléments **structurant** :
  - ▶ `<h1>`, `<h2>`, ..., `<h6>`,
  - ▶ `<article>`,
  - ▶ `<section>`,
  - ▶ `<aside>`, `<nav>`,
  - ▶ `<header>`, `<main>`, `<footer>`
- ▶ Éléments de **textes** :
  - ▶ simples :
    - ▶ paragraphe `<p>`,
    - ▶ citation `<blockquote>`, `<q>`,
    - ▶ *rupture de paragraphe* `<br/>`, *ligne horizontale* `<hr/>`
  - ▶ listes :
    - ▶ non-ordonnée `<ul>`,
    - ▶ ordonnée `<ol>`,
    - ▶ de définition `<dl>`,
    - ▶ items des listes `<li>` ou `<dd>`, `<dt>`
  - ▶ préformaté : `<pre>`
  - ▶ formatage logique (*sémantique*)
    - ▶ `<em>`, `<strong>`, `<code>`, `<kbd>`, etc.

# Corps du document (suite)

## ▶ Éléments **extérieurs** :

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

*hyperlien*

## ▶ Éléments de **tableaux** :

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

*parties  
ligne  
cellule*

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

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

## ▶ Éléments de **formulaires**

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

# Tableau



# Références

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

# 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 courte et projet professionnel
  - ▶ **identité détaillée** : nom, prénom, 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

- ▶ dans une archive au format `.zip` contenant votre dossier `html`
- ▶ archive nommée d'après votre identifiant
- ▶ adresse : **bruno@ape-cee.fr**
- ▶ objet du mail : **[M2APE] devoir 2 Prénom NOM**
- ▶ le **strict** respect de ces contraintes est important