

Le langage HTML

Mathieu Raynal

Qu'est ce que le HTML ?

- **HyperText Markup Language**
 - Langage à balises
 - Langage de description de page Web
 - Décrit la **structure** du document à afficher
- Historique
 - **Début 1997** – Le World Wide Web Consortium (W3C) publie la norme HTML 3.2
 - **Fin 97** – Version 4,0
 - **Depuis 2007** - HTML 5

Exemple de fichier HTML

Exemple de balises

```
<!DOCTYPE html>
<html>
<head>
  <title>Programme NSI - Classe de première</title>
</head>
<body>
  <h1>Programme de numérique et sciences informatiques de première
générale</h1>
  <h2>Préambule</h2>
  <p>L'enseignement de spécialité de numérique et sciences informatiques du
cycle terminal de la voie générale vise l'appropriation des fondements de
l'informatique pour préparer les élèves à une poursuite d'études dans
l'enseignement supérieur, en les formant à la pratique d'une démarche
scientifique et en développant leur appétence pour des activités de
recherche.</p>

  <p>L'objectif de cet enseignement, non professionnalisant, est
l'appropriation des concepts et des méthodes qui fondent l'informatique,
dans ses dimensions scientifiques et techniques. Cet enseignement s'appuie
sur l'universalité de quatre concepts fondamentaux et la variété de leurs
interactions<math>\text{&nbsp;}</math></p>

  <ul id="toto">
    <li>Les données, qui représentent sous une forme numérique unifiée des
informations très diverses : textes, images, sons, mesures physiques,
sommes d'argent, etc.</li>
    <li>Les algorithmes, qui spécifient de façon abstraite et précise des
traitements à effectuer sur les données à partir d'opérations élémentaires.
</li>
    <li>Les langages, qui permettent de traduire les algorithmes abstraits en
programmes textuels ou graphiques de façon à ce qu'ils soient exécutables
par les machines.</li>
    <li>Les machines, et leurs systèmes d'exploitation, qui permettent
d'exécuter des programmes en enchaînant un grand nombre d'instructions
simples, assurant la persistance des données par leur stockage, et de
gérer les communications. On y inclut les objets connectés et les
réseaux.</li>
  </ul>
</body>
</html>
```

Comment visualiser le résultat ?

- La navigateur effectue le rendu de la page HTML
 - Il interprète le fichier HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Programme NSI - Classe de première</title>
</head>
<body>
  <h1>Programme de numérique et sciences informatiques de première générale</h1>
  <h2>Préambule</h2>
  <p>L'enseignement de spécialité de numérique et sciences informatiques du cycle terminal de la voie générale vise l'appropriation des fondements de l'informatique pour préparer les élèves à une poursuite d'études dans l'enseignement supérieur, en les formant à la pratique d'une démarche scientifique et en développant leur appétence pour des activités de recherche.</p>
  <p>L'objectif de cet enseignement, non professionnalisant, est l'appropriation des concepts et des méthodes qui fondent l'informatique, dans ses dimensions scientifiques et techniques
```

Programme de numérique et sciences informatiques de première générale

Préambule

L'enseignement de spécialité de numérique et sciences informatiques du cycle terminal de la voie générale vise l'appropriation des fondements de l'informatique pour préparer les élèves à une poursuite d'études dans l'enseignement supérieur, en les formant à la pratique d'une démarche scientifique et en développant leur appétence pour des activités de recherche.

L'objectif de cet enseignement, non professionnalisant, est l'appropriation des concepts et des méthodes qui fondent l'informatique, dans ses dimensions scientifiques et techniques. Cet enseignement s'appuie sur l'universalité de quatre concepts fondamentaux et la variété de leurs interactions :

- Les données, qui représentent sous une forme numérique unifiée des informations très diverses : textes, images, sons, mesures physiques, sommes d'argent, etc.
- Les algorithmes, qui spécifient de façon abstraite et précise des traitements à effectuer sur les données à partir d'opérations élémentaires.
- Les langages, qui permettent de traduire les algorithmes abstraits en programmes textuels ou graphiques de façon à ce qu'ils soient exécutables par les machines.
- Les machines, et leurs systèmes d'exploitation, qui permettent d'exécuter des programmes en enchaînant un grand nombre d'instructions simples, assurant la persistance des données par leur stockage, et de gérer les communications. On y inclut les objets connectés et les réseaux.

Un peu de vocabulaire : les différentes balises

- Balise ouvrante

- Est constituée, au minimum, d'un nom entre chevrons

```
<head>
```

- Peut contenir en plus des attributs

```
<link rel="stylesheet" type="text/css" href="nsi.css">
```

- Balise fermante

- Est constituée uniquement d'un nom entre chevrons précédé du caractère « / »

```
</head>
```

Un peu de vocabulaire : attribut d'une balise

- Une balise ouvrante peut contenir des attributs
 - Ils sont définis après le nom de la balise
- Un attribut est une paire **nomAttribut="valeur"** ou **nomAttribut='valeur'**

```
<link rel="stylesheet" type="text/css" href="nsi.css">
```

Un peu de vocabulaire : élément

- Un élément se compose de
 - Une balise ouvrante
 - Un contenu
 - Une balise fermante (avec le nom identique à celui de la balise ouvrante)
- Un élément vide est constitué uniquement d'une balise ouvrante
 - et se termine par « /> »

```
<ul>  
  <li>Préambule</li>  
  <li>Démarche de projet</li>  
  <li>Modalités de mise en oeuvre</li>  
  <li>Éléments de programme</li>  
</ul>
```

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

Contenu d'un élément

- Le contenu peut être constitué
 - De texte
 - D'autres éléments
- Deux éléments ne peuvent s'entrecouper
 - Soit ils sont complètement disjoints
 - Soit l'un est inclus dans l'autre

```
<aside>
  <h1>Programme complet de 1<sup>ère</sup></h1>
  <br><center><a href="1e_NSI.pdf"></a></center>
</aside>
```


Quelques convention à respecter

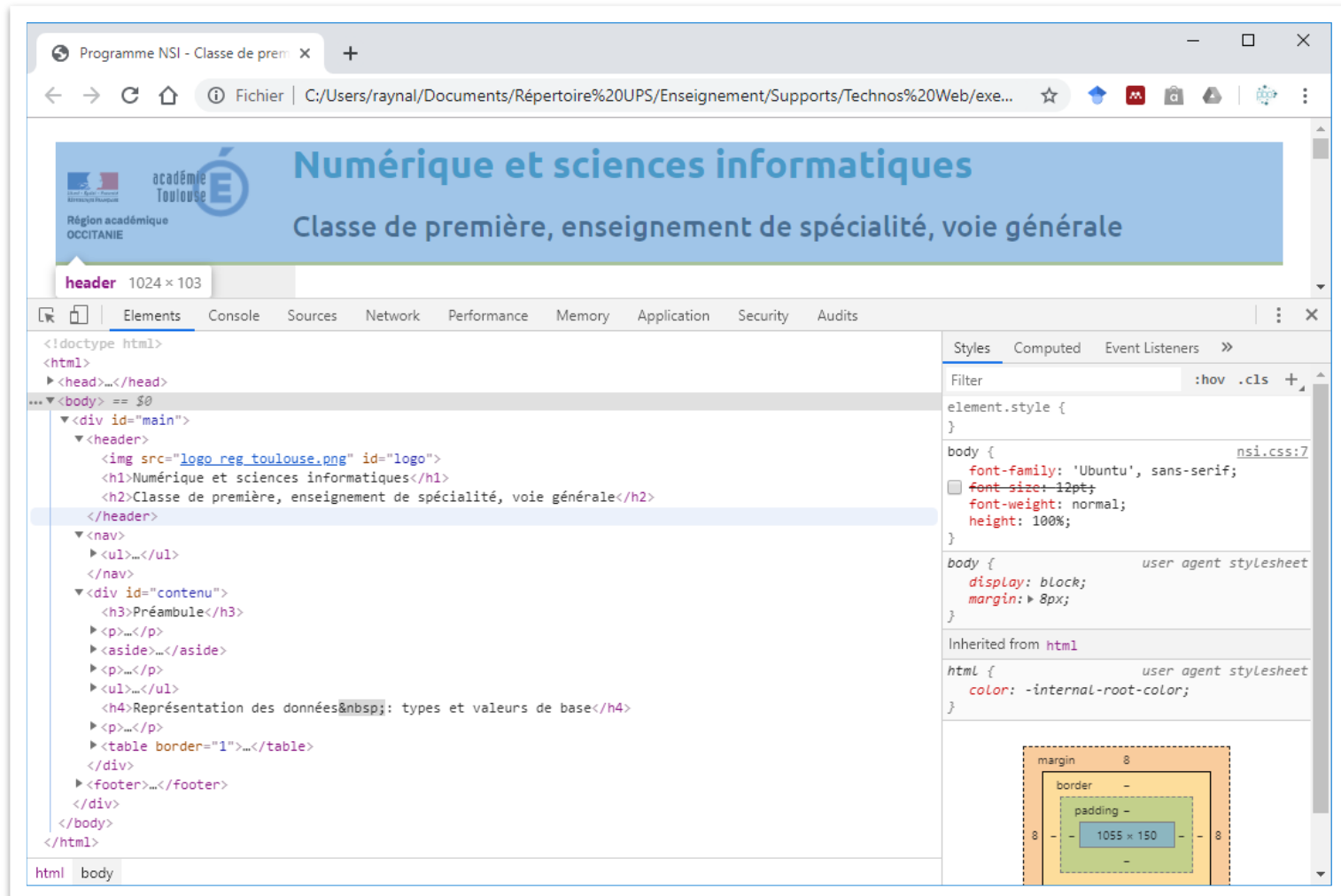
- Les navigateurs sont assez permissifs ...
 - Interprètent les balises
 - Capables de corriger des erreurs
- ... mais il est préférable de respecter les conventions suivantes pour plus de clarté
 - Le nom des balises en **minuscule**
 - Dans les balises
 - Un espace entre chaque attribut
 - Pas d'espace entre le nom de l'attribut, le signe '=' et la valeur de l'attribut
 - Chaque valeur d'attribut comprise entre " " ou ' '

Les outils pour développer sa page HTML

- Écrire du code HTML
 - Éditeur de texte simple : Bloc notes, NotePad, **Sublime Text**
<https://www.sublimetext.com/>
 - IDE : eclipse, intelliJ
- Visualiser le résultat
 - Dans un navigateur : Internet explorer, Firefox, Chrome, etc.

Corriger les éventuelles erreurs

- Dans le menu du navigateur
 - « Plus d'outils » → « Outils de développement »



Vérifier la validité / conformité de son document

- Validité du document : <https://validator.w3.org/>

- Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
From line 11, column 4; to line 11, column 47
`ader>` `` `<h`
- Error** The `center` element is obsolete. [Use CSS instead](#).
From line 31, column 9; to line 31, column 16
`>` `
` `<center>` `<a href`
- Error** Bad value `40px` for attribute `width` on element `img`: Expected a digit but saw `p` instead.
From line 31, column 38; to line 32, column 17
`_NSI.pdf">` `` `` `</`
- Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).

Attention à l'accessibilité

- Un document HTML est principalement destiné à être affiché sur des écrans
- Mais il est aussi utilisé par des lecteurs d'écran
 - Prend en compte les éléments utilisés pour la restitution
- Important d'utiliser les bons éléments
 - Les éléments de style ont disparu au profit des feuilles de style
 - De plus en plus d'éléments sémantiques dans HTML 5
- WAI : Web Accessibility Initiative
 - WCAG : Web Content Accessibility Guidelines
 - Liste d'outils : <https://www.w3.org/WAI/ER/tools/>

Structure d'un fichier HTML

- Première ligne du fichier : le doctype
 - Indique qu'il s'agit d'une page au format HTML
- L'élément racine : **html**
- Constitué d'un en-tête, défini par l'élément **head**
- Et d'un contenu, décrit par l'élément **body**
- Des commentaires peuvent être insérés partout

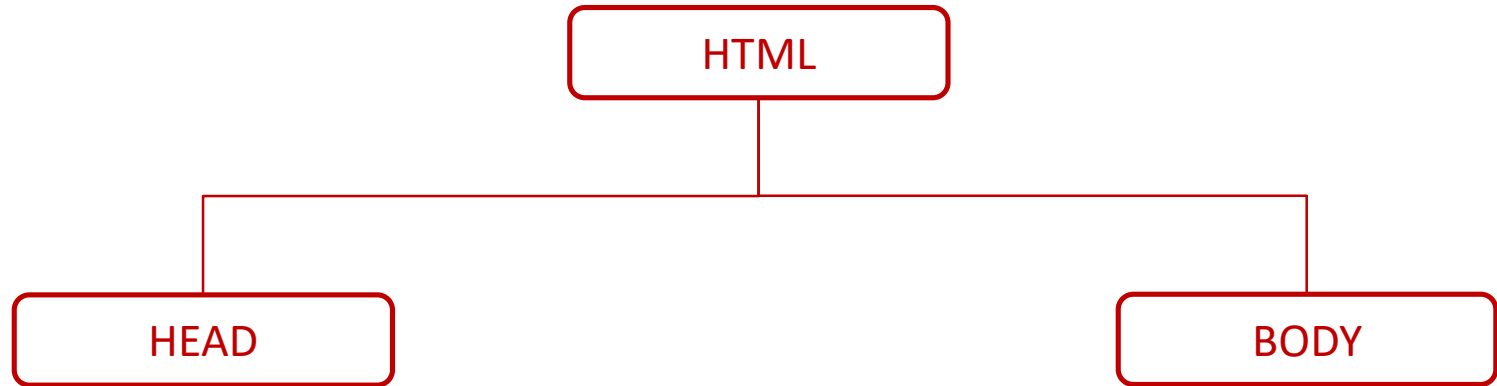
```
<!DOCTYPE html>
```

Mettre des commentaires

- Informations qui ne seront pas affichées sur la page
- Les commentaires peuvent être placés n'importe où dans le document

```
<!-- Ceci est un commentaire -->
```

Une structure arborescente



Exemple de fichier HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Programme NSI - Classe de première</title>
</head>
<body>
  <h1>Programme de numérique et sciences informatiques de première
  générale</h1>
  <h2>Préambule</h2>
  <p>L'enseignement de spécialité de numérique et sciences informatiques du
  cycle terminal de la voie générale vise l'appropriation des fondements de
  l'informatique pour préparer les élèves à une poursuite d'études dans
  l'enseignement supérieur, en les formant à la pratique d'une démarche
  scientifique et en développant leur appétence pour des activités de
  recherche.</p>

  <p>L'objectif de cet enseignement, non professionnalisant, est
  l'appropriation des concepts et des méthodes qui fondent l'informatique,
  dans ses dimensions scientifiques et techniques. Cet enseignement s'appuie
  sur l'universalité de quatre concepts fondamentaux et la variété de leurs
  interactions<nbsp>;</p>

  <ul id="toto">
    <li>Les données, qui représentent sous une forme numérique unifiée des
    informations très diverses : textes, images, sons, mesures physiques,
    sommes d'argent, etc.</li>
    <li>Les algorithmes, qui spécifient de façon abstraite et précise des
    traitements à effectuer sur les données à partir d'opérations élémentaires.
    </li>
    <li>Les langages, qui permettent de traduire les algorithmes abstraits en
    programmes textuels ou graphiques de façon à ce qu'ils soient exécutables
    par les machines.</li>
    <li>Les machines, et leurs systèmes d'exploitation, qui permettent
    d'exécuter des programmes en enchaînant un grand nombre d'instructions
    simples, assurant la persistance des données par leur stockage, et de
    gérer les communications. On y inclut les objets connectés et les
    réseaux.</li>
  </ul>
</body>
</html>
```

L'en-tête

L'élément Head : L'en-tête

- Contient des informations sur le document HTML ...
- ... mais ne sont pas affichées sur la page HTML
 - Le titre de la page : élément **title**

```
<title>Programme NSI - Classe de première</title>
```

- L'encodage utilisé : élément **meta**, attribut *charset*

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

- Les liens avec des fichiers extérieurs
 - CSS
 - Javascript

L'élément Base

- Définit l'URL de base à utiliser pour recomposer toutes les URL relatives contenues dans un document
- Il ne peut y avoir qu'un seul élément base au sein d'un document
- Attributs
 - href : URL de base à utiliser
 - target :
 - _self : contexte courant
 - _blank : nouveau contexte de navigation
 - _parent : dans le contexte parent
 - _top : ancêtre du contexte courant, sans parent

L'élément Link

- Utilisé pour définir un lien entre le document et une autre ressource, telle que :

- Feuille de style

```
<link href="style.css" rel="stylesheet">
```

- Icône utilisée en barre de titre

```
<link href="ups.ico" rel="icon">
```

- Ses principaux attributs

- rel : définit le type de lien
- href : détermine l'url de la ressource à lier
- hreflang : définit la langue de la ressource utilisée
- type : définit le type de contenu

L'élément Script

- Cet élément sert à intégrer un script
 - généralement du code JavaScript
- L'élément peut également faire référence à un fichier de script extérieur. Dans ce cas, ses attributs :
 - src : définit l'URI du script à exécuter
 - type : définit le type de script (pas obligatoire en HTML5)

```
<script src="adapter-min.js"></script>
```

L'élément Style

- Cet élément permet de définir la mise en forme pour le document.
 - Par défaut, ces instructions de mise en forme sont écrites en CSS

```
<style type="text/css">
  h1{
    background-color: red;
  }

  h2{
    background-color: green;
  }
</style>
```

→ Favoriser les feuilles de style externes et les lier avec **link**

L'élément Title

- Définit le titre du document
- Le titre est affiché dans
 - la barre de titre du navigateur
 - l'onglet
- Ne doit pas contenir d'autres éléments

```
<title>Cours de présentation du langage HTML</title>
```


L'élément Meta

- Permet de définir des métadonnées qui ne sont pas décrites par les autres éléments
 - charset : définit l'encodage des caractères utilisé dans le document

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

- name : définit le nom d'une meta-donnée. La valeur associée à ce nom doit être contenu dans l'attribut content
 - author
 - description
 - keywords

```
<meta name="keywords" content="cours, web, html">
```

Le contenu

Le contenu

- Contient toutes les informations qui seront affichées sur la page
- Structuré à partir d'éléments « sémantiques »
- 2 types principaux d'éléments
 - Éléments « **en ligne** »
 - ne commence pas sur une nouvelle ligne
 - contient du texte et/ou d'autres éléments « en ligne »
 - Éléments « **en bloc** »
 - passe à la ligne avant et après l'élément
 - Contient du texte ou tout autre élément

Structuration du texte

Les titres et sous-titres

- 6 niveaux de titre
 - éléments de type « en bloc »
- Titre de niveau le plus élevé : élément **h1**
- Titre de niveau le plus faible : élément **h6**

```
<h1>Numérique et sciences informatiques</h1>  
<h2>Classe de première, enseignement de spécialité, voie  
générale</h2>
```

Regroupement de titres

- Élément **hgroup** : Regroupe des titres de plusieurs niveaux
 - Pratique pour appliquer le même style
 - Evite de former différentes sections dans le plan

```
<hgroup>  
  <h1>Les standards du web</h1>  
  <h2>Quelques définitions</h2>  
</hgroup>
```

Paragraphe et saut de ligne

- Définir un paragraphe : élément de type « en bloc » **p**

```
<h3>Préambule</h3>  
<p>L'enseignement de spécialité de numérique et sciences  
informatiques du cycle terminal de la voie générale vise  
l'appropriation des fondements de l'informatique pour  
préparer les élèves à une poursuite d'études dans  
l'enseignement supérieur, en les formant à la pratique  
d'une démarche scientifique et en développant leur  
appétence pour des activités de recherche.</p>
```

- Retour à la ligne : élément vide **br**

Éléments spécifique du texte

- Éléments «en ligne» qui permettent d'ajouter de l'information structurelle à un fragment de texte
 - **em** : emphase
 - **strong** : emphase plus forte
 - **small** : texte écrit en plus petit
 - **mark** : texte marqué, généralement en gras ou surligné
 - **abbr** : indique une abréviation
 - Attribut title pour apporter la description complète
 - **address** : indique une adresse
 - **sub** : mettre le contenu en indice
 - **sup** : mettre le contenu en exposant

Éléments pour les citations

- **blockquote** : élément « en bloc » pour afficher une citation assez conséquente
 - Paragraphe généralement en retrait
 - Attribut cite pour donner l'URL de sa référence
- **q** : élément « en ligne » qui met le contenu entre guillemets
- **cite** : élément « en ligne »
 - Permet de renseigner la source

Pré-formater un texte

- Élément de type « en bloc » pour afficher un texte pré-formater

```
<pre>
                Il était une fois :
                    un
exemple
                de pré-formatage
</pre>
```

Éléments pour afficher du code

- Éléments « en ligne » permettant de mettre en évidence les chaînes de caractères liées à de la programmation
 - **code** : Contient un fragment de code de programmation
 - **samp** : Contient un exemple de sortie de programme
 - **kbd** : Indique un texte à saisir par l'utilisateur
 - Également utilisé pour afficher des raccourcis clavier
 - **var** : Indique une variable ou un argument dans un programme

Caractères spéciaux

- Pour afficher les caractères difficilement accessibles via les touches du clavier, il faut les saisir au moyen d'un code précédé de & et suivi d'un ;
- Quelques exemples
 - * * : espace insécable
 - *«* : «
 - *»* : »
 - *©* : ©

Élément time

- Permet de représenter des dates et des heures

```
Ce travail est à rendre pour le <time datetime="2021-09-25">25 Septembre</time>
```

- Pour que ce soit exploitable, il faut l'attribut datetime qui fournit une date et une heure dans un format valide
 - Pour une année : AAAA
 - Année et mois : AAAA-MM
 - Année, mois et jour : AAAA-MM-JJ
 - Date et heure : AAAA-MM-JJ**T**HH:Min:Sec.MS

Liste non-ordonnée

- Liste non ordonnée
 - élément de type block **ul** (*unordered list*)
 - *Contient que les items constituant la liste*
- Item d'une liste
 - élément de type block **li** (*item list*)

```
<ul>  
  <li>Préambule</li>  
  <li>Démarche de projet</li>  
  <li>Modalités de mise en oeuvre</li>  
  <li>Éléments de programme</li>  
</ul>
```

- Préambule
- Démarche de projet
- Modalités de mise en oeuvre
- Éléments de programme

Liste ordonnée

- Liste ordonnée
 - élément de type block **ol** (*ordered list*)
 - Contient que les items constituant la liste
 - Les items sont numérotés dans leur ordre d'ajout à la liste

```
<ol>  
  <li>Introduction</li>  
  <li>Les éléments de structuration du texte</li>  
  <li>Les éléments de constitution d'une liste</li>  
  <li>Les liens</li>  
  <li>Les tableaux</li>  
</ol>
```

1. Introduction
2. Les éléments de structuration du texte
3. Les éléments de constitution des listes
4. Les liens
5. Les tableaux

Liste de définition

- L'élément **dl** annonce une liste de définition contenant
 - Des termes contenus dans des éléments **dt**
 - Et leur définition encapsulée dans un élément **dd**

```
<dl>
  <dt>URI (Uniform Ressource Identifier)</dt>
  <dd>L'identifiant de ressource uniforme désigne un élément permettant d'identifier une ressource.</dd>

  <dt>URL (Uniform Ressource Locator)</dt>
  <dd>L'URL est une information permettant de localiser un élément.
  L'adresse d'un site internet est une URL.
  Mais, une adresse postale est aussi une URL !</dd>
</dl>
```

URI (Uniform Ressource Identifier)

L'identifiant de ressource uniforme désigne un élément permettant d'identifier une ressource.

URL (Uniform Ressource Locator)

L'URL est une information permettant de localiser un élément. L'adresse d'un site internet est une URL. Mais, une adresse postale est aussi une URL !

Créer un paragraphe rétractable

- Élément **details**
- Possibilité de lui donner un titre : élément **summary**

```
<details>  
  <summary>Lorem ipsum</summary>  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do  
  eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut  
  enim ad minim veniam, quis nostrud exercitation ullamco laboris  
  nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor  
  in reprehenderit in voluptate velit esse cillum dolore eu fugiat  
  nulla pariatur. Excepteur sint occaecat cupidatat non proident,  
  sunt in culpa qui officia deserunt mollit anim id est laborum.  
</details>
```

▼ Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Les images

Image

- Privilégier les images au format
 - JPEG pour les photos
 - PNG pour les graphiques
 - GIF pour les images animées
- Ajouter une image : élément vide de type inline **img**
- Les attributs de la balise img
 - *src* : définit l'emplacement de l'image à afficher (*obligatoire*)
 - *alt* : Donne une information textuelle sur l'image (*optionnel*)

```

```

Figure

- Permet de mettre une légende à une image
- Ajouter une légende avec l'élément **figcaption**
- Élément « sémantique » de type block **figure**
 - contient l'image (*élément img*)
 - suivi de sa légende (*élément figcaption*)

```
<figure>
  
  <figcaption>Plan de l'Université</figcaption>
</figure>
```

Les liens

Lien vers une autre document

- Les liens sont caractérisés par l'élément de type inline **a**

Plus d'informations sur le site web du `Rectorat`

- Le contenu de l'élément sera la zone cliquable
- Les attributs
 - *href* permet d'indiquer vers quelle page on souhaite être dirigé
 - Une page externe en indiquant son URL
 - Une autre page du site en indiquant le chemin relatif pour y accéder
 - *target="_blank"* permet d'ouvrir le document lié au lien dans une nouvelle fenêtre ou onglet
 - *title* permet de définir un texte qui sera ouvert dans une infobulle lorsque le curseur survolera le lien

Lien vers un élément du document HTML

- Pour pouvoir pointer vers un élément du document, il faut que celui-ci soit identifié
 - Attribut *id* avec une valeur unique

```
<h3 id="preamble">Préambule</h3>
```

- Lien vers un élément du document : *#id*

```
<a href="#preamble">Préambule</a>
```

- Lien vers un élément d'un autre document : *lien_du_doc#id*

Les tableaux

Tableau

- Déclaration d'un tableau : élément de type block **table**
 - Contient les différentes lignes du tableau
- Créer une ligne : élément **tr**
 - Contient les cellules de la ligne
- Définir une cellule : élément **td**

Produits	kcal/100g
Abricot	43
Ananas	55
Artichaut	47
Asperge	18

```
<table border="1">  
  <tr>  
    <td>Produits</td>  
    <td>kcal/100g</td>  
  </tr>  
  <tr>  
    <td>Abricot</td>  
    <td>43</td>  
  </tr>  
  <tr>  
    <td>Ananas</td>  
    <td>55</td>  
  </tr>  
  <tr>  
    <td>Artichaut</td>  
    <td>47</td>  
  </tr>  
  <tr>  
    <td>Asperge</td>  
    <td>18</td>  
  </tr>  
</table>
```

Les cellules du tableau

- Ligne d'en-tête
 - première ligne du tableau
 - utiliser l'élément **th** à la place de **td**
- Fusionner des cellules
 - De différentes colonnes : attribut *colspan*
 - De différentes lignes : attribut *rowspan*

Produits		kcal/100g
Fruits	Abricot	43
	Ananas	55
Légumes	Artichaut	47
	Asperge	18

```
<table border="1">
  <tr>
    <th colspan="2">Produits</th>
    <th>kcal/100g</th>
  </tr>
  <tr>
    <td rowspan="2">Fruits</td>
    <td>Abricot</td>
    <td>43</td>
  </tr>
  <tr>
    <td>Ananas</td>
    <td>55</td>
  </tr>
  <tr>
    <td rowspan="2">Légumes</td>
    <td>Artichaut</td>
    <td>47</td>
  </tr>
  <tr>
    <td>Asperge</td>
    <td>18</td>
  </tr>
</table>
```

Les groupes de lignes

- **thead** : définit un ensemble de lignes qui définit l'en-tête des colonnes d'un tableau.
- **tfoot** : permet de définir un ensemble de lignes qui résument les colonnes d'un tableau.
- **tbody** : permet de regrouper un ou plusieurs éléments tr afin de former le corps d'un tableau HTML

Les groupes de colonnes

- **colgroup** définit un groupe de colonnes au sein d'un tableau.
- **col** définit une colonne appartenant à un tableau et est utilisé afin de définir la sémantique commune à toutes ses cellules.

Formulaire

L'élément form

- Regroupe l'ensemble des éléments appartenant au formulaire
- Attributs nécessaires
 - *method* : définit quelle manière utiliser pour envoyer les données
 - GET
 - POST
 - *action* : détermine quel fichier va traiter les données recueillies

```
<form method="post" action="formulaire.php">
```

L'élément input

- Élément vide qui regroupe la majorité des éléments utilisables dans un formulaire
 - Champ de saisie monoligne, case à cocher, etc.
- Attributs nécessaires
 - *type* : détermine le type d'élément
 - *name* : donne un nom à l'élément
 - Important pour récupérer les données

Champ de saisie

- Élément input avec type="text"
- Attributs possibles
 - *value* pour pré-remplir le champ avec une valeur par défaut
 - *maxlength* pour limiter le nombre de caractères que l'on peut saisir
 - *placeholder* pour donner une indication sur le contenu du champ

Champ de saisie spécifique

- Différents types pour avoir un champ de saisie mais avec un format de données ou un affichage spécifique
 - *password*
 - *email*
 - *url*
 - *tel*
 - *number*
 - Attributs possibles : *min, max, step*
 - *range*
 - Attributs possibles : *min, max, step*
 - *color*
 - *date, time, week, month, datetime, datetime-local*
 - *search*

- Libellé avant un élément du formulaire : Élément **label**
- Lier le label au champ de saisie : attribut *for*
 - Nécessite que l'élément input ait un attribut *id*

```
<label for="nom">Nom</label><input type="text" id="nom" name="nom">
```

Cases à cocher

- Élément input avec type="checkbox"
- Label après pour entouré le texte
- Case sélectionnée par défaut : Attribut *checked* sans valeur

```
<input type="checkbox" id="latin" name="latin" checked>  
<label for="latin">Option Latin</label>
```

Choix unique

- Élément input avec type="radio"
- Un seul choix possible
- Les cases sont regroupées grâce à la valeur de *name*
- Utiliser l'attribut *value* pour savoir quelle case a été sélectionnée

```
<input type="radio" id="latin" name="option">  
<label for="latin">Option Latin</label>  
  
<input type="radio" id="grec" name="option" checked>  
<label for="grec">Option Grec</label>
```

Transfert de fichiers

- Élément input avec type="file"
- Permet de sélectionner un ou plusieurs fichiers
- Les fichiers sélectionnés seront transférés vers le serveur au moment de la validation du formulaire
- Attributs spécifiques
 - accept : définit les formats de fichier acceptés
 - Capture : caméra à utiliser pour prendre la photo (user|environnement)

Les champs cachés

- Élément input avec type="hidden"
- Permet d'inclure dans le formulaire des valeurs
 - Non affichées sur le navigateur
 - Envoyé avec le reste des valeurs lors de la validation du formulaire
 - La valeur n'est pas modifiable une fois le document affiché

```
<input id="numeroEtudiant" name="numeroEtudiant" type="hidden" value="xmj2347q">
```

L'élément output

- Zone dans laquelle on peut insérer le résultat d'une opération émanant d'autres éléments

```
<form oninput="res.value=parseInt(val1.value)+parseInt(val2.value)">
  <input type="range" name="val1" value="0" min="0" max="10" step="1"/>
  +
  <input type="number" name="val2" value="0" min="0" max="10" step="1"/>
  =
  <output name="res" for="val1 val2">0</output>
</form>
```

Champ de saisie multi-ligne

- Élément **textarea**
- Le contenu de l'élément sert de valeur par défaut au champ de saisie
- Déterminer la taille du champ de saisie
 - Attributs *rows* et *cols*
 - Ou, en CSS, avec les propriétés *width* et *height*

Listes déroulantes

- La liste est définie avec l'élément **select**
 - Besoin de l'attribut *name*
- Les items de la liste sont définis avec l'élément **option**
 - Nécessaire d'avoir l'attribut *value*
 - L'attribut *selected* (sans valeur) pour présélectionner un item
- Pour regrouper des items dans la liste : élément **optgroup**
 - L'attribut *label* donne un nom au groupe (non sélectionnable)

```
<select id="classe" name="classe">
  <optgroup label="Seconde">
    <option value="2A">2nde A</option>
    <option value="2B" selected>2nde B</option>
    <option value="2C">2nde C</option>
  </optgroup>
</select>
```

Bouton d'envoi du formulaire

- Bouton permettant à l'utilisateur de valider et envoyer les résultats de son formulaire : Élément input avec `type="submit"`
 - L'attribut *value* permet de changer le libellé du bouton
- `Type="image"` crée également un bouton d'envoi, avec une image à la place du libellé
 - Besoin de l'attribut *src*

Bouton de remise à zéro

- Réinitialiser les champs du formulaire sans l'envoyer
 - Élément input avec type="reset"
- Créer un bouton, autre que pour l'envoi du formulaire
 - Élément input avec type="button"

Naviguer dans un formulaire

- Attribut *tabindex*
 - prend une valeur entière entre 0 et 32767
 - établit la position d'un élément dans le «tabbing order»
 - si deux éléments ont la même valeur, l'ordre est l'ordre d'apparition dans le flux du formulaire.
- Attribut *accesskey*
 - prend une valeur de type caractère
 - raccourci clavier – accessibilité
 - éléments supportant l'attribut : a, input, textarea, label, legend

Paramètres des éléments du formulaire

- Attribut *autofocus* : met le curseur directement sur l'élément
- Attribut *required* : rend le champ obligatoire
- En CSS, possibilité de personnaliser en fonction de l'état :
 - *:required*
 - *:invalid*
 - *:focus*

Regrouper les champs

- Regrouper et encadrer un ensemble d'élément du formulaire
 - Élément **fieldset**
- Donner un titre, sur la bordure, à ce regroupement
 - Élément **legend**

Exemple

Elève		Bulletin de notes	
Nom	<input type="text"/>	Français	<input type="text"/>
Prénom	<input type="text"/>	Mathématiques	<input type="text"/>
Classe	<input type="text" value="2nde B"/>	LV1	<input type="text"/>
		LV2	<input type="text"/>
		Histoire-Géo	<input type="text"/>
		Physique	<input type="text"/>
		SVT	<input type="text"/>
		Bilan	<input type="text"/>
		<input type="button" value="Valider"/>	

Coordonnées

Adresse

Courriel

Téléphone

Options & spécialités

Options

☒ Option Latin

☐ Option Grec

Spécialités

☐ Mathématiques

☒ Informatique

☒ Physique

☐ SVT