HTML : introduction

au programme...

au programme...

HTML

Ca veut dire quoi « **HTML** » ?

HTML

HTML

HTML

Hyper**T**ext **ML**

HTML

Hyper**T**ext **M**arkup **L**

HTML

Hyper**T**ext **M**arkup **L**anguage

HTML

Hyper**T**ext **M**arkup **L**anguage

langage à balises pour hypertextes

HTML

HyperText Markup Language
langage à balises pour hypertextes

exemple – sncf

HTML

HyperText Markup Language
langage à balises pour hypertextes

- un langage
  - de description de documents web, il permet de structurer le contenu de ces documents

exemple – sncf

HTML

HyperText Markup Language
langage à balises pour hypertextes

- un langage
  - de description de documents web, il permet de structurer le contenu de ces documents
- à balises
  - la structure des documents est organisée à l'aide de balises

exemple – sncf

HTML

HyperText Markup Language
langage à balises pour hypertextes

- un langage
  - de description de documents web, il permet de structurer le contenu de ces documents
- à balises
  - la structure des documents est organisée à l'aide de balises
- pour hypertextes
  - les documents contiennent des **hyperliens** permettant un accès direct à une autre partie du document

exemple – sncf

HTML

HTML5

version majeure actuelle du langage HTML

- ajout de balises par rapport aux versions précédentes

XHTML

XHTML

- une « version » d'HTML conforme au langage **XML**
- XML : *« eXtensible Meta Language »*
- impose des règles d'écritures

XHTML5

$$\text{XHTML5} \equiv \text{HTML5}$$

- mêmes balises
- une syntaxe plus rigoureuse

au programme...

au programme...

langage

C'est quoi un langage ?

langage

vocabulaire + syntaxe + sémantique

langage

vocabulaire + syntaxe + sémantique

- syntaxe = grammaire
- sémantique = sens

langage

vocabulaire + syntaxe + sémantique

- syntaxe = grammaire
- sémantique = sens

Le **vocabulaire** permet de construire des textes qui respectent la **syntaxe** et qui ont un **sens**.

vocabulaire

éléments prédéfinis, identifiés par des **balises**

vocabulaire

éléments prédéfinis, identifiés par des **balises**

- ouvrante `<element>`

vocabulaire

éléments prédéfinis, identifiés par des **balises**

- ouvrante/fermante : `<element>` / `</element>`

vocabulaire

éléments prédéfinis, identifiés par des **balises**

- ouvrante/fermante : `<element>` / `</element>`

  `<h1>` `<p>` `<strong>` ...

vocabulaire

éléments prédéfinis, identifiés par des **balises**

- ouvrante/fermante : `<element>` / `</element>`

  `<h1>` `<p>` `<strong>` / `</h1>` `</p>` `</strong>` ...

vocabulaire

éléments prédéfinis, identifiés par des **balises**

- ouvrante/fermante : `<element>` / `</element>`

  `<h1>` `<p>` `<strong>` / `</h1>` `</p>` `</strong>` ...
- balises **auto-fermantes** : `<element/>`

vocabulaire

éléments prédéfinis, identifiés par des **balises**

- ouvrante/fermante : `<element>` / `</element>`

  `<h1>` `<p>` `<strong>` / `</h1>` `</p>` `</strong>` ...
- balises **auto-fermantes** : `<element/>`

  `<br/>` `<img/>`

vocabulaire

éléments prédéfinis, identifiés par des **balises**

- ouvrante/fermante : `<element>` / `</element>`

  `<h1>` `<p>` `<strong>` / `</h1>` `</p>` `</strong>` ...
- balises **auto-fermantes** : `<element/>`

  `<br/>` `<img/>`
- HTML insensible à la casse : `<eLEmENt>` = `<eLeMeNT>`

vocabulaire

éléments prédéfinis, identifiés par des **balises**

- ouvrante/fermante : `<element>` / `</element>`

  `<h1>` `<p>` `<strong>` / `</h1>` `</p>` `</strong>` ...
- balises **auto-fermantes** : `<element/>`

  `<br/>` `<img/>`
- HTML insensible à la casse : `<eLEmENt>` = `<eLeMeNT>`
  mais XHTML ⇒ **minuscules** : `<element>`

liste

On appelle **contenu d'un élément** le code qui se trouve entre ses balises ouvrante et fermante.

 contenu de l'élément p

On appelle **contenu d'un élément** le code qui se trouve entre ses balises ouvrante et fermante.

 contenu de l'élément p

NB : les balises auto-fermantes ont un contenu vide.

au programme...

au programme...

syntaxe

La **syntaxe** définit

- la structure du document
- les règles d'écriture

structure minimale

structure minimale

```
<!DOCTYPE html>
```

← déclaration du **DOCTYPE**

structure minimale

```
<!DOCTYPE html>
```
← déclaration du **DOCTYPE**

```
<html xmlns="http://www.w3.org/1999/xhtml">
```
← un élément **racine**

```
</html>
```

structure minimale

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

  <!-- entête du document -->
  <head>


  </head>




</html>
```

← déclaration du **DOCTYPE**

← un élément **racine**

← l'**entête** du document

structure minimale

```
<!DOCTYPE html>                                    ← déclaration du DOCTYPE

<html xmlns="http://www.w3.org/1999/xhtml">        ← un élément racine

  <!-- entête du document -->
  <head>                                           ← l'entête du document


  </head>

  <!-- corps du document -->
  <body>                                           ← le corps du document
    <!-- on place ici le contenu de la page -->
...
  </body>

</html>
```

structure minimale

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

  <!-- entête du document -->
  <head>
    <title>Document HTML 5 minimal</title>

  </head>

  <!-- corps du document -->
  <body>
    <!-- on place ici le contenu de la page -->
...
  </body>

</html>
```

← déclaration du **DOCTYPE**

← un élément **racine**

← l'**entête** du document
← un **titre**

← le **corps** du document

structure minimale

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

  <!-- entête du document -->
  <head>
    <title>Document HTML 5 minimal</title>
    <meta charset="UTF-8"/>
  </head>

  <!-- corps du document -->
  <body>
    <!-- on place ici le contenu de la page -->
...
  </body>

</html>
```

← déclaration du **DOCTYPE**

← un élément **racine**

← l'**entête** du document
- ← un **titre**
- ← la déclaration de l'**encodage** de caractères utilisé

← le **corps** du document

règles d'écriture

parenthésage

Un document html valide est **bien parenthésé**.

règles d'écriture

parenthésage

Un document html valide est **bien parenthésé**.

1. A toute balise ouvrante `<element>` doit être associée une balise fermante `</element>`.

règles d'écriture

parenthésage

Un document html valide est **bien parenthésé**.

1. A toute balise ouvrante `<element>` doit être associée une balise fermante `</element>`.
2. Les éléments ne doivent pas se chevaucher :
   premier ouvert, dernier fermé

règles d'écriture

parenthésage

Un document html valide est **bien parenthésé**.

1. A toute balise ouvrante `<element>` doit être associée une balise fermante `</element>`.
2. Les éléments ne doivent pas se chevaucher :
   premier ouvert, dernier fermé

```
<p> debut <code>emboité ?</p> suite </code>
```

règles d'écriture

parenthésage

Un document html valide est **bien parenthésé**.

1. A toute balise ouvrante `<element>` doit être associée une balise fermante `</element>`.
2. Les éléments ne doivent pas se chevaucher :
   premier ouvert, dernier fermé

```
<p> debut <code>emboité ?</p> suite </code>
```

! illégal !

règles d'écriture

parenthésage

Un document html valide est **bien parenthésé**.

1. A toute balise ouvrante `<element>` doit être associée une balise fermante `</element>`.
2. Les éléments ne doivent pas se chevaucher :
   premier ouvert, dernier fermé

```
<p> debut <code>emboité ?</code> suite </p>
```

règles d'écriture

emboitement (1)

Le contenu d'un élément peut être constitué d'autres éléments « **emboités** ».

règles d'écriture

emboitement (1)

Le contenu d'un élément peut être constitué d'autres éléments « **emboités** ».

```
<p>
    début du contenu de p
    <code>
        début emboité 1
        <strong>
            contenu emboité 2
        </strong>
        fin emboité 1
    </code>
    suite du contenu de p
</p>
```

règles d'écriture

emboitement (2)

Les emboitements obéissent à des règles.
Tous les emboitements ne sont pas possibles.

règles d'écriture

emboitement (2)

Les emboitements obéissent à des règles.
Tous les emboitements ne sont pas possibles.

exemples :

- un élément <p> ne peut pas contenir un élément <h1>
- un élément <ul> contient nécessairement au moins un élément
- un élément <li> est nécessairement emboité dans un élément
- etc.

au programme...

au programme...

document valide

Un document (X)HTML est **valide** s'il respecte toutes ces règles.

exemple

validation

Les outils de **validation** permettent de vérifier la correction syntaxique d'un document.

`http://validator.w3.org/nu` (extension `xhtml`)

exemple (bis)

validation

Les outils de **validation** permettent de vérifier la correction syntaxique d'un document.

`http://validator.w3.org/nu` (extension `xhtml`)

exemple (bis)

important

La validation d'un document produit doit être **systématique**

au programme...

au programme...

sémantique

A chaque élément (balise) est associée une **sémantique** qui définit son usage.
Elle permet de savoir quand et pourquoi utiliser un élément.

sémantique

A chaque élément (balise) est associée une **sémantique** qui définit son usage.
Elle permet de savoir quand et pourquoi utiliser un élément.

- l'élément <p> sert à représenter un paragraphe *MDN*

sémantique

A chaque élément (balise) est associée une **sémantique** qui définit son usage.
Elle permet de savoir quand et pourquoi utiliser un élément.

- l'élément <p> sert à représenter un paragraphe *MDN*
- l'élément <time> sert à identifier une heure ou une date *MDN*

sémantique

A chaque élément (balise) est associée une **sémantique** qui définit son usage.
Elle permet de savoir quand et pourquoi utiliser un élément.

- l'élément <p> sert à représenter un paragraphe *MDN*
- l'élément <time> sert à identifier une heure ou une date *MDN*
- l'élément <td> représente une « case » dans un tableau *MDN*

sémantique

A chaque élément (balise) est associée une **sémantique** qui définit son usage.
Elle permet de savoir quand et pourquoi utiliser un élément.

- l'élément <p> sert à représenter un paragraphe *MDN*
- l'élément <time> sert à identifier une heure ou une date *MDN*
- l'élément <td> représente une « case » dans un tableau *MDN*
- l'élément sert à donner de l'importance à un texte *MDN*

sémantique

A chaque élément (balise) est associée une **sémantique** qui définit son usage.
Elle permet de savoir quand et pourquoi utiliser un élément.

- l'élément <p> sert à représenter un paragraphe *MDN*
- l'élément <time> sert à identifier une heure ou une date *MDN*
- l'élément <td> représente une « case » dans un tableau *MDN*
- l'élément sert à donner de l'importance à un texte *MDN*
- etc.

sémantique

A chaque élément (balise) est associée une **sémantique** qui définit son usage.
Elle permet de savoir quand et pourquoi utiliser un élément.

- l'élément <p> sert à représenter un paragraphe *MDN*
- l'élément <time> sert à identifier une heure ou une date *MDN*
- l'élément <td> représente une « case » dans un tableau *MDN*
- l'élément sert à donner de l'importance à un texte *MDN*
- etc.

Aux balises s'ajoutent bien évidemment les contenus texte pour donner la sémantique (le sens) global du document.

à suivre ...

HTML : les éléments