

Introduction au HTML



Introduction

Qu'est-ce que l'HTML ?

- Un langage, **standard** du web, créé en 1991
- **HyperText Markup Language**, soit “langage de balisage hypertexte”
- Fichier lisible par les navigateurs web (Google Chrome, Safari, etc.)

A quoi sert-il ?

C'est un **langage de structure**

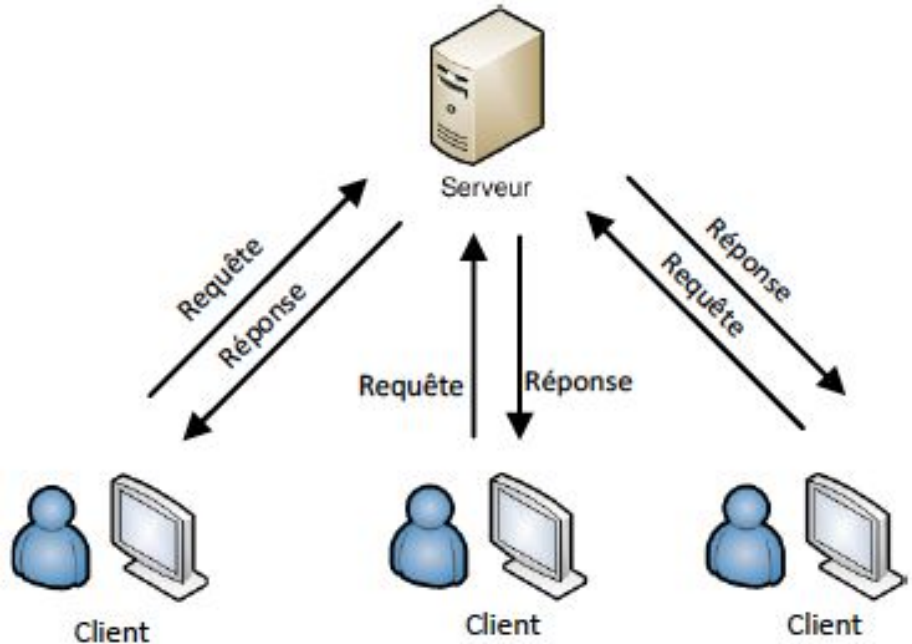
Contexte du HTML

Qu'est-ce qu'un site internet ?

Un ensemble de **fichiers** de code (HTML, CSS, etc.) et de médias (images, vidéos, etc.)

Qu'est-ce qu'un serveur ?

Un ordinateur qui **stocke les fichiers** d'un ou plusieurs site internet



Les éléments

Il permettent de :

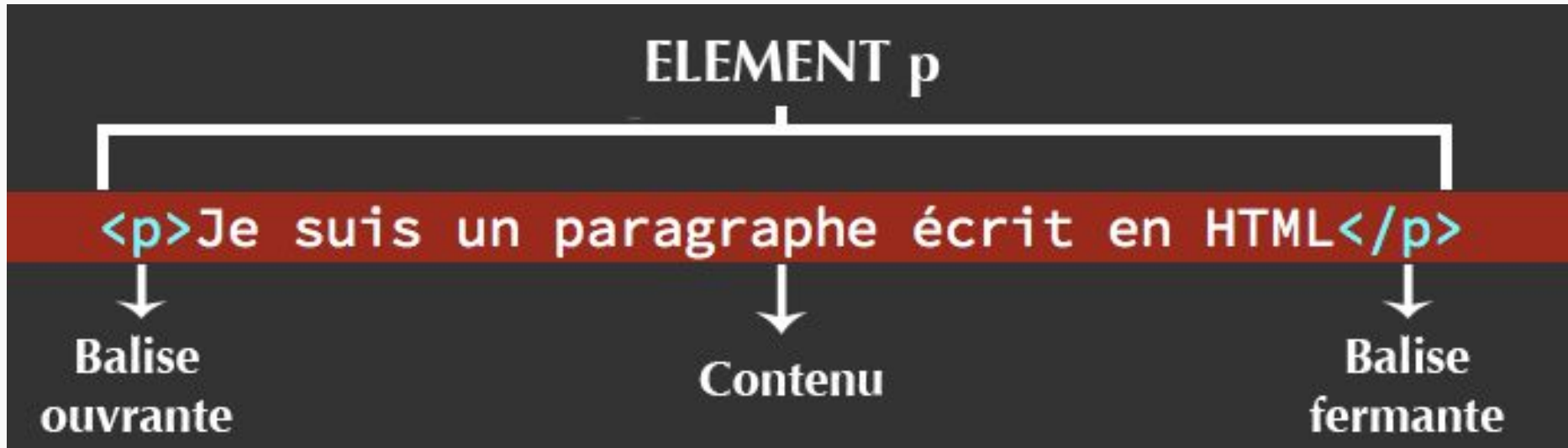
- **créer la structure** de la page HTML
- **définir chaque contenu** de la page
- **passer des informations** au navigateur pour afficher la page

De quoi sont-ils constitués ?

- Soit d'une **paire de balise** (une ouvrante et une fermante) et **son contenu**
- Soit d'une **balise unique**, dite "orpheline"
- Et de 0 à n **attribut(s)**

Les éléments

Premier exemple :



Les éléments

Deuxième exemple :

ELEMENT br



`
`



Balise orpheline

Les attributs

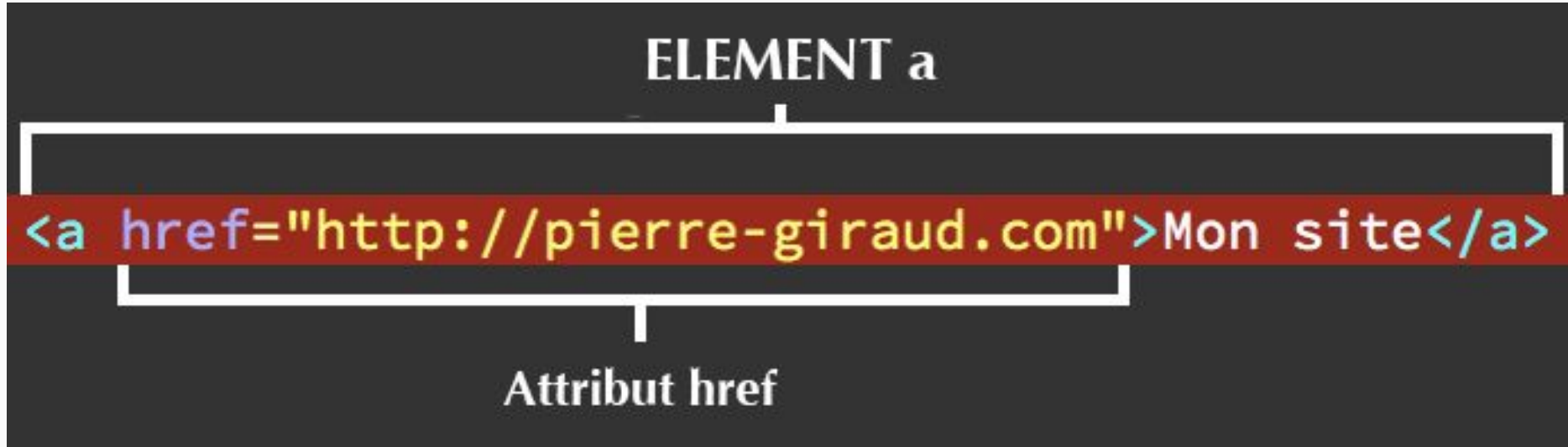
Il servent à **compléter un élément** :

- soit pour le définir plus précisément
- soit pour lui apporter des informations complémentaires sur son comportement

Un attribut contient **toujours** une valeur

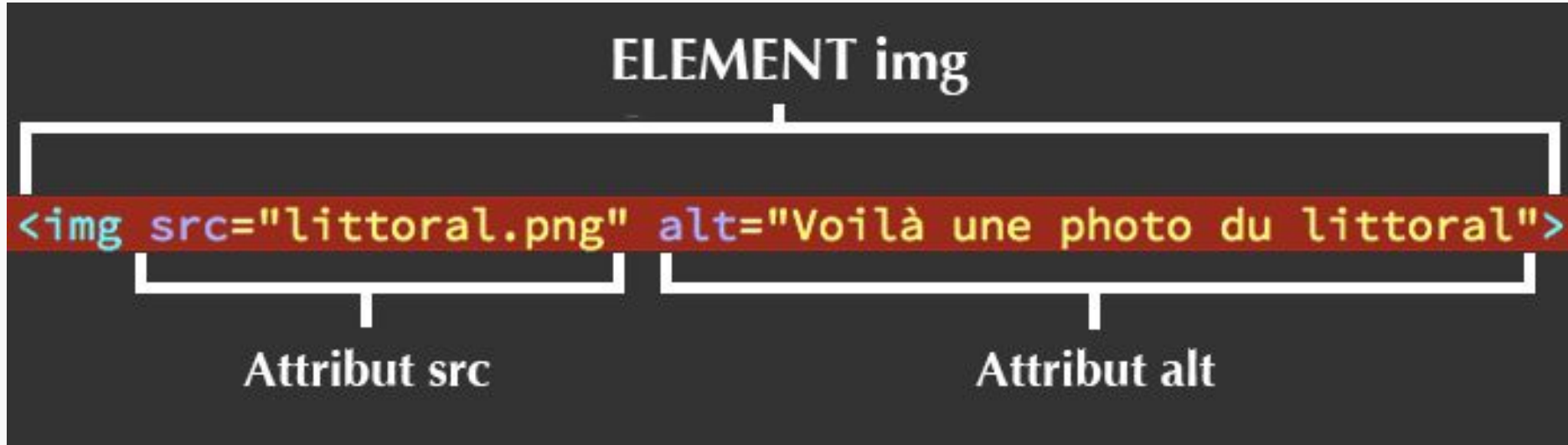
Les attributs

Premier exemple :



Les attributs

Deuxième exemple :



L'imbrication

On peut imbriquer 0 à n élément(s) dans un autre, on parle ici de **dépendance**

Il est important de bien **indenter** son code pour une meilleure lisibilité

Exemple :

```
<nom_element_A>  
  <nom_element_A1></nom_element_A1>  
  <nom_element_A2></nom_element_A2>  
  <nom_element_A3>  
</nom_element_A>
```

Structure minimale d'une page HTML

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Ma première page HTML</title>
    <meta charset="utf-8">
  </head>
  <body>

  </body>
</html>
```

Les commentaires

Ils permettent d'expliquer à quoi correspond un bout de code, donner des indications sur son fonctionnement

3 situations dans lesquelles ils sont utiles :

- Gros projets
- Travailler à plusieurs
- “mettre en standby” une partie du code

```
<!-- Voici mon commentaire -->
```

Les liens

- Pour changer de page
 - Externes : **href="http://le_site_externe.html"**
 - Internes : **href="le_chemin_vers_mon_fichier.html"**
- Naviguer au sein d'une même page : **href="#mon_id"**
- Pour permettre l'envoi d'un mail : **href="mailto:mon.mail@gmail.com"**
- Pour permettre le téléchargement d'un fichier :
href="le_chemin_vers_mon_fichier.extension"
download="le_nom_du_fichier_telecharge"

Autres attributs utiles

Pour tous les éléments : l'attribut **class**

Pour l'élément **ol** :

- l'attribut **type** : soit 1 (valeur par défaut), soit I, soit i, soit A, soit a
- l'attribut **start** : le numéro auquel commencera la liste
- l'attribut **reversed** : une valeur possible qui est reversed (**valeur évidente**)

Pour l'élément **li** dans une liste organisée :

- l'attribut **value** : la valeur de la puce

Autres éléments utiles

Balises ouvrantes/fermantes :

- `pre`
- `audio` et `video`
- `table`

Balises orphelines :

- `hr`

L'élément audio ou vidéo

Une balise ouvrante / fermante **audio** ou **video** est composé d'un attribut **controls** à valeur évidente (affiche les contrôles du lecteur audio)

Dans le contenu de la balise audio ou video, il y a un élément **source** pour chaque fichier

La balise **source** est orpheline et se compose de 2 attributs :

- **src** : le chemin vers le fichier audio ou vidéo
- **type** : le format de codec audio ou vidéo du fichier (audio/mpeg, audio/wave, audio/webm, audio/ogg, video/mp4 ou video/webm)

L'élément audio ou vidéo

A la suite du contenu de la balise audio ou vidéo, ajouter ce texte à la suite des éléments source : **Votre navigateur ne supporte pas ce fichier**

Quelques attributs facultatifs de la balise audio ou video :

- **preload** : avec pour valeur none, metadata ou auto
- **autoplay** : valeur évidente
- **loop** : valeur évidente
- **muted** : valeur évidente
- **poster** (seulement video) : avec pour valeur le chemin vers le fichier image

L'élément table

Un **tableau simple** est constitué d'une balise ouvrante / fermante **table**

Il contient des balises ouvrantes / fermantes **tr** (acronyme pour "table row" ou "ligne de tableau" en français)

Chaque tr contient des balises ouvrantes / fermantes **td** (acronyme pour "table data" ou "donnée de tableau" en français). C'est donc la cellule d'un tableau.

L'élément table

Pour créer des tableaux plus complexes : définir un **thead**, un **tbody** et un **tfoot**

thead : en-tête du tableau

tbody : corps du tableau

tfoot : pied du tableau

Dans le thead, on remplace les cellules td par des **th**

Toutes ces balises sont de type ouvrantes / fermantes

L'élément table

Pour avoir une bordure dans le tableau, il faut ajouter un attribut **border** à la balise table qui a pour valeur 0, 1, 2, etc.

Pour ajouter une légende au tableau, il faut insérer un enfant à l'élément table. C'est une balise ouvrante / fermante nommée **caption**.

Pour fusionner des cellules d'une ligne, il faut ajouter l'attribut **colspan** à une cellule. **Attention**, le nombre de td par ligne diffère donc !

Pour fusionner des cellules d'une colonne, il faut ajouter l'attribut **rowspan** à une cellule. **Attention**, le nombre de td par ligne diffère donc !

Les entités HTML

C'est **une suite de caractères** utilisée pour afficher **un caractère réservé ou invisible**.

- les espaces : ` ` / ` ` / ` ` / ` `
- le & (ou “et commercial”) : `&`
- les chevrons < et > : `<` et `>`

Consultez ces liens :

- [Le tableau des différents caractères et leur code](#)
- [Un décodeur \(tapez le caractère, il vous donne son code, et inversement\)](#)

Les formulaires

Balise ouvrante / fermante **form** composé de 2 attributs : **method** et **action**

L'attribut method n'a que 2 valeurs possible : **"get"** et **"post"**

L'attribut action a pour valeur le chemin vers le fichier php

Les formulaires - Les champs

L'élément **input** s'écrit sous la forme d'une balise orpheline. Elle contient un attribut **type** qui a pour valeur : text, password, email, number, date, radio, checkbox, submit, etc.

Pour afficher un libellé devant votre zone de texte, il faut ajouter une balise ouvrante / fermante nommée **label**.

Pour associer le label avec l'input, il faut attribuer à chaque input un identifiant unique. Puis, ajouter un attribut **for** aux labels avec pour valeur l'identifiant de l'input associé.

Les formulaires - Les champs

Autres éléments de formulaires :

- select
- textarea
- button

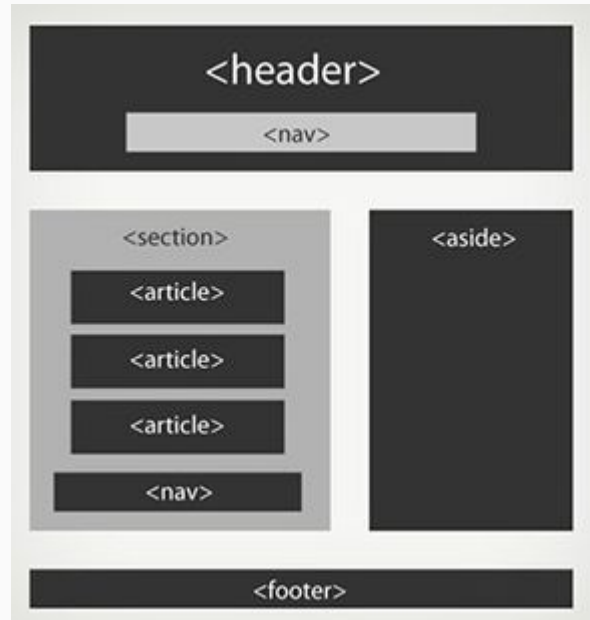
Tous les types de champs d'un formulaire peuvent contenir un attribut **required** (requis en français) à valeur évidente. Le formulaire ne pourra être validé que lorsque tous les champs requis sont remplis.

Pour valider le formulaire il faut un élément input ou button qui contient un attribut **type** qui a pour valeur **submit**.

Structuration d'une page HTML

Afin de bien structurer une page web, on va utiliser les éléments suivants (balises de type ouvrantes / fermantes) :

- header
- footer
- section
- article
- nav
- aside



Eléments link et script

Afin de lier un fichier CSS à notre fichier HTML on utilise l'élément link qui s'écrit avec une balise de type orpheline :

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

Pour lier un fichier JS à notre fichier HTML on utilise l'élément script qui s'écrit avec une balise de type ouvrante / fermante :

```
<script type="text/javascript" src="chemin_vers_mon_fichier.js"></script>
```