

T erm 	<h1>Construction d'un site Web</h1>	
Cours /TD prof	Mots clé : structure, arborescence, squelette, layout, template, framework.	

Centre d'intérêt

CI4 : Gestion de l'information / Structures matérielles et Logicielles

Tutoriels associés :

- "Concevoir un site Web pour mobile"
- "Présenter des données dans une page web"
- "Concevoir le layout d'un site web"



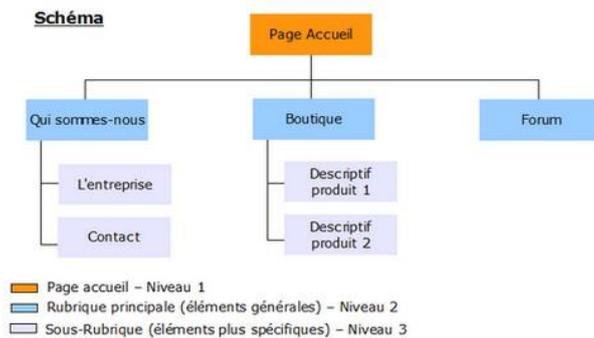
1 - Introduction

« Avant de se lancer la tête la première dans la création d'un site Internet, quel qu'il soit, il est important de se demander à qui l'on souhaite s'adresser, comment et avec quels outils. »
Webdesign HS n°15.

2 - Structure du site

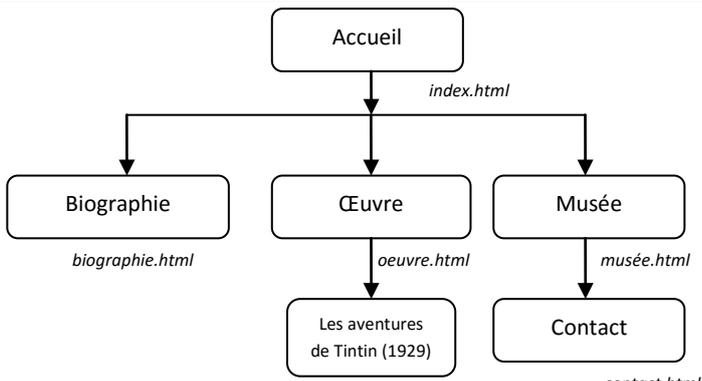
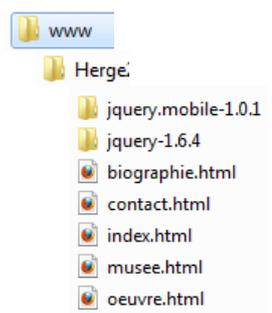
Une fois que les considérations, introduites au paragraphe précédent, ont été prises en compte, il faut penser à la structure du site. Celle-ci pourra être représentée comme un **arbre généalogique**. Il est souhaitable de penser le site en **largeur** (dans l'idéal, ne pas dépasser **sept menus**) et en **profondeur** (dans l'idéal, ne pas dépasser **trois niveaux** pour respecter la règle des **trois clics**).

Exemple :



Le contenu du site doit être **organisé sur le serveur**. Les pages du site seront placées dans le répertoire racine du site. Des sous répertoires accueilleront les images, les fichiers .css, .js etc...

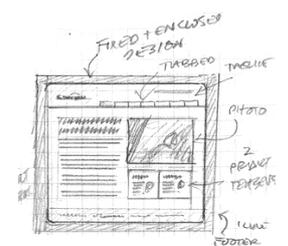
Exemple : Site pour mobile réalisé en TP

Structure du site "Hergé"	Arborescence des répertoires du site "Hergé" sur le serveur
	 <div style="text-align: right; margin-top: 20px;">  </div>

3 – La structure des pages du site

L'utilisation du site sera facilitée lorsque les pages présentent la **même structure**.

3.1 Le squelette = dessin (à la main) de la structure des pages du site



Exemple



3.2 Le template (ou gabarit) des pages du site

Template

« Un **gabarit**, souvent nommé **template**, est un patron de mise en page où l'on place images et textes. Aussi, il est souvent utilisé de manière **répétitive** pour créer des documents présentant une **même structure**. » Wikipédia. Le template intègre la structure, la décoration et le dynamisme de la page.

Exemple : Template correspondant à la structure de la page ci-dessus.

Gabarit (template) téléchargeable sur le site JQuery Mobile

```
<!DOCTYPE html>
<html>

<head>
  <title>My Page</title>
  <meta charset="utf-8"/>
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0.1/jquery.mobile-1.0.1.min.css" />
  <script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
  <script src="http://code.jquery.com/mobile/1.0.1/jquery.mobile-1.0.1.min.js"></script>
</head>

<body>
  <div data-role="page">
    <div data-role="header">
      <h1>Titre affiché dans la page </h1>
    </div><!-- /header -->

    <div data-role="content">
      <p>Le corps de la page</p>
    </div><!-- /content -->

    <div data-role="footer" data-theme="d">
      <h4> TSTI2D SIN PEM BOURGES </h4>
    </div><!-- /footer -->

  </div><!-- /page -->
</body>
</html>
```

Liens relatifs vers des fichiers situés dans les répertoires du site. **Annexe 3 Exercices 1 et 2**

} En-tête de la page à afficher
 } Contenu de la page à afficher
 } Pied de la page à afficher

Framework

Remarque : L'utilisation d'un template mettant en œuvre un **Framework** tel que JQuery permet de développer rapidement un site dynamique présentant un "certain Design".

Le DOCTYPE précise au navigateur la version du code HTML utilisé. Son écriture, complexe avant , se limite aujourd'hui à : **<!DOCTYPE html>** sur chacune des pages du site.

Layout

On parle de **layout** lorsque le gabarit ne contient que la structure (html) et la décoration (css). (voir le lien en page 6)

4 - Quelques balises incontournables lors de l'écriture des pages du site

<meta>

Sans trop entrer dans les détails, nous dirons que la **balise <meta>** donne des indications au navigateur.

Exemple : <meta charset="utf-8 » /> <!--indique l'encodage utilisé dans la page -->

<link>

La **balise <link>** permet de lier une feuille de style (.css) à la page web.

Exemple : <link rel="stylesheet" href="style.css" />

<script>

Les **balises <script>** permettent aux objets de la page d'avoir accès aux fonctions JavaScript des bibliothèques JQuery et JQuery Mobile.

Exemple : <script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>

<div>

Cette balise a été conçue pour **introduire une division (boîte ou conteneur sans passer par une balise de paragraphe)**. Elle est particulièrement utile pour la déclaration de feuilles de style.

Exemple : <div data-role="page"> </div>

5 - Le positionnement en CSS

En HTML, la plupart des balises peuvent se ranger dans l'une ou l'autre des **catégories : block et inline**:

block

- **block** : une balise de type block crée automatiquement un **retour à la ligne avant et après**.

Une page web est constituée d'un ensemble de blocs que l'on peut mettre les uns à l'intérieur des autres.

Exemple de balise de type block : les titres <h></h>, les paragraphes <p></p>

<h1> Titre (block) </h1>

- **inline** : une balise de type inline se trouve **obligatoirement à l'intérieur** d'une balise de type block.

Une balise de type inline ne crée pas de retour à la ligne.

inline

Exemple de balise de type inline.

C'est par exemple le cas des liens : <a>

<p> Paragraphe blablabla <a>
Lien inline </p>

- **Transformation d'un élément avec "display"**

La propriété CSS **display** permet de transformer n'importe quel élément d'un type dans un autre.

Valeur	Exemples	Description
inline	<a>, , ...	Éléments d'une ligne. Se placent les uns à côté des autres.
block	<p>, <div>, <section>...	Éléments en forme de blocs. Se placent les uns en dessous des autres et peuvent être redimensionnés.
inline-block	<select>, <input>	Éléments positionnés les uns à côté des autres (comme les inlines) mais qui peuvent être redimensionnés (comme les blocs).
none	<head>	Éléments non affichés.

Exemple : Positionnement des liens les uns en dessous des autres et redimensionnement possible.

```
a {
  display : block ;
}
```

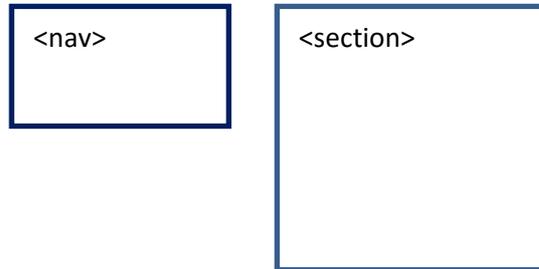
• **Positionnement inline-block**

Les éléments de type **inline-block** se positionnent **les uns à côté des autres** et on peut leur donner des dimensions précises.

Exemple : positionnement d'un **bloc nav** à côté d'un bloc section.

```
.nav {
    display : inline-block ;
    width : 150px ;
    border: 1px solid black;
    vertical-align: top;
}

.section {
    display : inline-block ;
    border: 1px solid black;
}
```



6 – Rappel : le bon usage des balises (règles définies en XHTML et reprises par )

- **Règle 1** : Ecrire les balises en **minuscule**.
- **Règle 2** : Toutes les balises **ouvertes** par `<nom_balise>` doivent être **fermées** par `</nom_balise>` sauf s'il s'agit d'une balise unique `<balise_unique />`.
- **Règle 3** : Les balises doivent être correctement **imbriquées** et **indentées**

Exemple : Dans le code ci-dessous, le `<a>` est **imbriquée** avec `<p></p>`, `<div>` et `<p>` sont **indentés**.

```

<div>
  <p><a href="page2.html" data-role="button">Oeuvre</a></p>
</div>
```

- **Règle 4**: Les valeurs des **attributs** doivent toujours figurer entre des **guillemets**.
Exemple : `< a href="bibliographie.html" data-role="button">`

7 - Résumé sur l'écriture des pages web

Les pages du site sont **structurées en zones** définies par des balises HTML `<div>`.

Dans le cas général, ces balises seront décorées par des css accessibles à travers des **classes**. Les css sont écrites par le développeur (Exercice 2) ou issues d'un Framework (TP4 HTML).

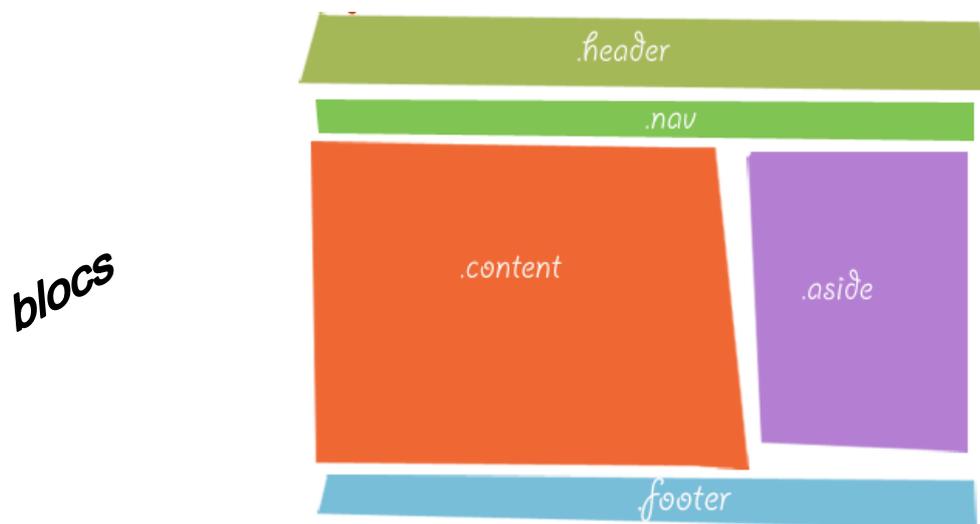
A l'intérieur des zones, des balises HTML permettent :

- de créer des **titres** (`<h> </h>`)
- d'espacer les éléments en introduisant des **paragraphes** (`<p></p>`)
- de créer des **liens vers d'autres pages** (`Nom du lien `)
- de créer des **listes à puces** (` ` et ` `) etc.

Annexe 1 : Structure HTML de base (squelette)

Source : <http://goo.gl/0HtQJ9>¹

« Le **squelette** d'une page web ressemble grosso modo à ça :



Cette structure de base à **cinq blocs principaux** convient dans la majorité des cas, car elle **permet de fabriquer une grande variété** de mises en page. Bien que ce soit conceptuellement très simple, ça reste toujours assez **délicat à mettre en page en HTML et CSS**.

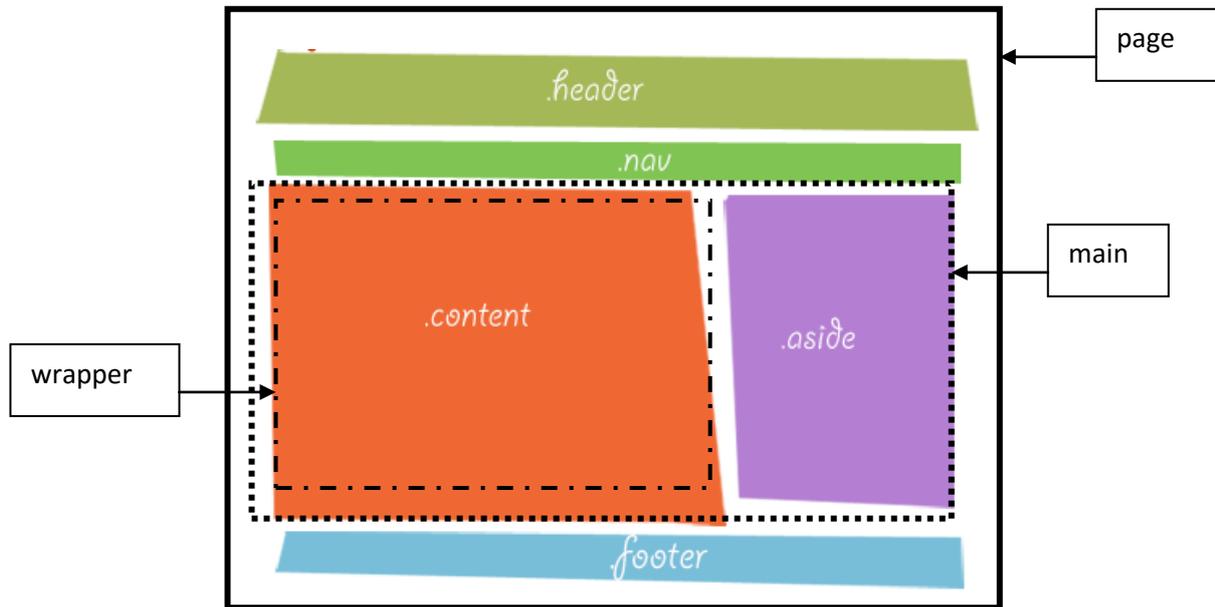
Codage du **layout** de la page ci-dessus :

HTML	CSS
<pre><!DOCTYPE html> <html> <head> <title>Titre de la page</title> <link rel="stylesheet" href="style.css"/> </head> <body> <div class="header">En-tête</div> <div class="nav">Navigation</div> <div class="content">Contenu</div> <div class="aside">Contexte</div> <div class="footer">Pied de page</div> </body> </html></pre>	<pre>/* Mes styles */ .header {...} .nav {...} .wrapper {...} .aside {...} .footer {...}</pre>

Remarque : pour être vraiment à l'aise en intégration CSS, il est utile d'ajouter quelques `<div>` englobantes :

- **page** (parfois appelé `.container`) **englobe toute la page**, comme son nom l'indique
- **main** contient les blocs centraux, ici `.content` et `.aside`, mais qui pourrait être plus nombreux.
- **wrapper** englobe le `.content` seul, juste pour **faciliter la mise en page**. »¹

Représentation de la page web en tenant compte de la remarque précédente.

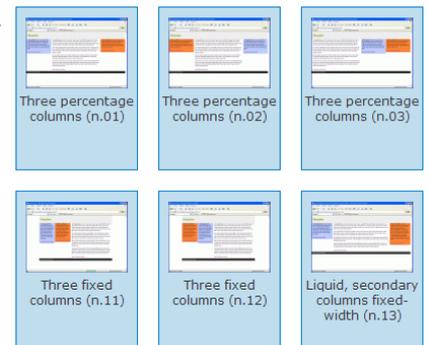


Codage du **layout** de la page en tenant compte de la remarque ci-dessus :

HTML	CSS
<pre> <!DOCTYPE html> <html> <head> <title>Titre de la page</title> <link rel="stylesheet" href="style.css"/> </head> <body> <div class="page"> <div class="header">En-tête</div> <div class="nav">Navigation</div> <div class="main"> <div class="wrapper"> <div class="content">Contenu</div> </div> <div class="aside">Contexte</div> </div> <div class="footer">Pied de page</div> </div> </body> </html> </pre>	<pre> /* Mes styles */ .page {...} .header {...} .nav {...} .main {...} .wrapper {...} .content {...} .aside {...} .footer {...} </pre>

Il existe des collections de modèles de mise en page, à télécharger gratuitement.

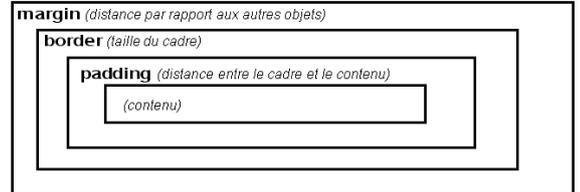
Exemple : « [Layout Gala](#) »



Annexe 2 : Notion de boîte (bloc)

L'**élément boîte** ou « **bloc** » est une notion importante en CSS !

« **Certains éléments** du (x)html peuvent être considérés comme une boîte.² Le rôle du navigateur est alors d'afficher toutes ces boîtes et leur contenu à l'écran selon ses propres spécifications ».



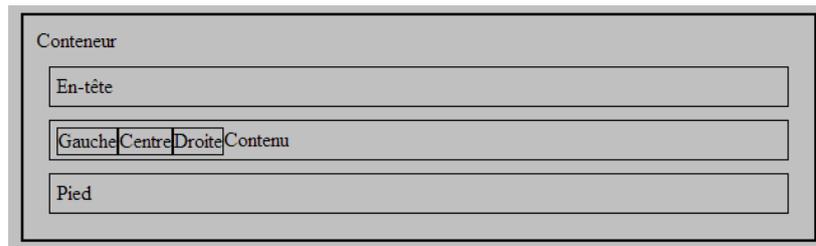
Propriétés de style pour dimensionner la boîte: **width** et **height** (pour le contenu), **border**, **margin**, **padding**.

Les navigateurs modernes disposent d'un outil appelé « Developer tools », permettant d'analyser les éléments d'une page Web. (**F12 avec Chrome**)

Cet outil permet de visualiser les propriétés de style des blocs de la page.

Exemple : balise div

Page Web analysée par « Developer tools » dans chrome



Code HTML

```

Elements Resources Network
<!DOCTYPE html>
<html>
  <head>
    <title> Test CSS3 a </title>
    <link rel="stylesheet" type="text/css" />
  </head>
  <body>
    <div id="container">
      "
      "      Conteneur
      <!-- En-tête -->
      <div id="header">
        En-tête
      </div>
      <!-- header -->
      <!-- Contenu -->
      <div id="content">...</div>
    </div>
  </body>
</html>

```

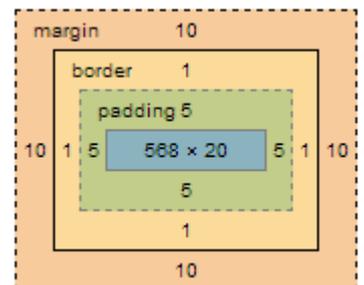
CSS de la boîte "header"

```

Matched CSS Rules
#header {
  border: 1px solid black;
  padding: 5px;
  margin: 10px;
}

```

Dimension de la boîte "header"



Structure HTML et rendu CSS des balises : bloc et en-ligne²

Pour aller plus loin...

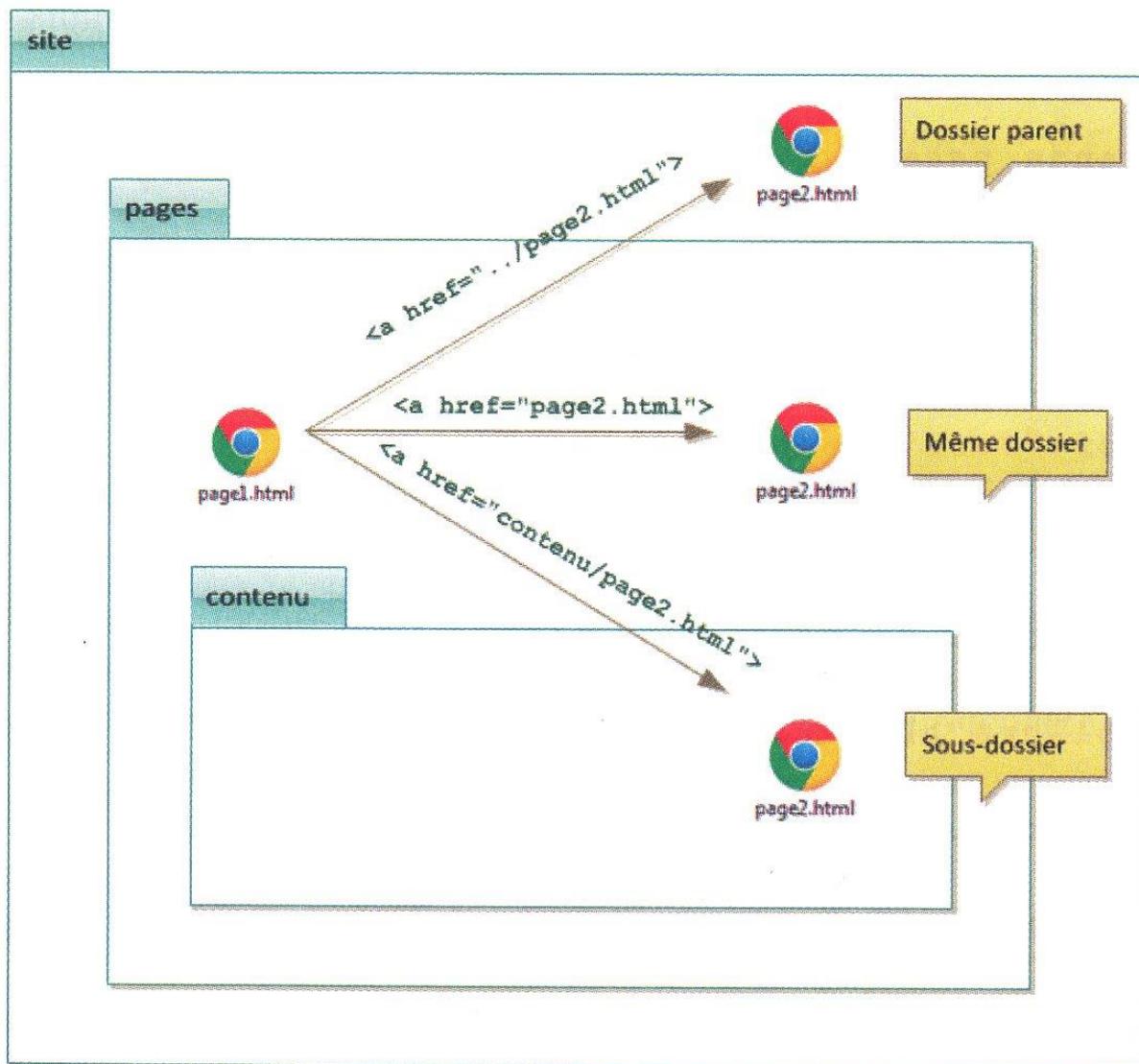


<http://goo.gl/3s8Woy>

Annexe 3 : Les liens relatifs

Un lien avec une adresse relative utilise une adresse par rapport au document en cours.

La figure ci-dessous donne l'écriture du lien relatif lorsque le fichier page1.html veut accéder à page2.html selon que ce dernier se trouve dans un **dossier parent**, dans le **même dossier** ou dans un **sous dossier**.



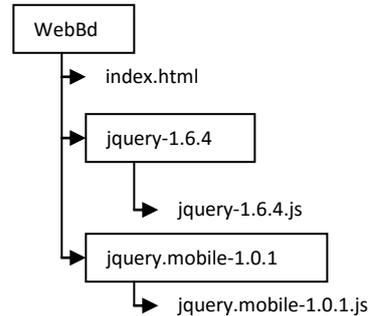
Exercices

Exercice 1 : Dessin de l'arborescence des répertoires d'un site à partir de la lecture de liens relatifs.

Le fichier *index.html*, situé dans le répertoire WebBD contient le code ci-dessous.

```
<link rel="stylesheet" href="jquery.mobile-1.0.1/jquery.mobile-1.0.1.css" />
<script src="jquery-1.6.4/jquery-1.6.4.js"></script>
<script src="jquery.mobile-1.0.1/jquery.mobile-1.0.1.js"></script>
```

Complétez l'arborescence, ébauché ci-contre, des répertoires du site.

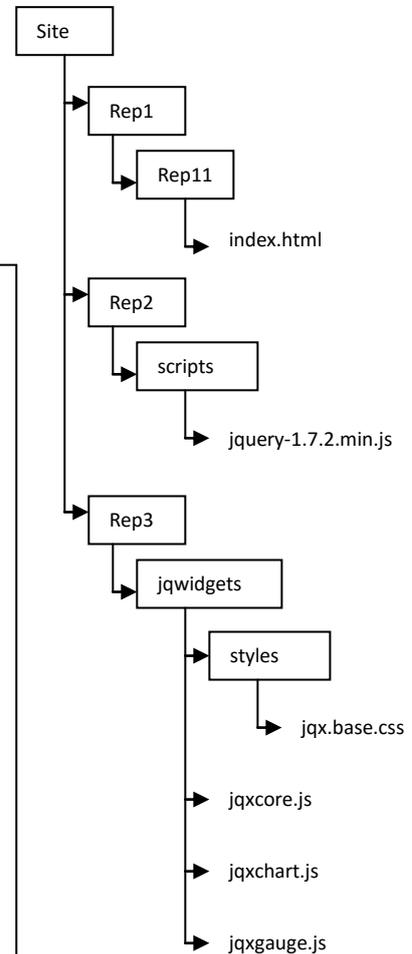


Exercice 2 : Dessin de l'arborescence des répertoires d'un site à partir de la lecture de liens relatifs.

Le code ci-dessous est issu d'exemples téléchargés sur le site jqWidget.com. Le fichier *index.html*, situé dans le répertoire Rep11, contient le code ci-dessous.

```
<link rel = "stylesheet" href = "../jqwidgets / styles / jqx.base.css" type = "text / css" />
<Script type = "text / javascript" src = "../scripts/jquery-1.7.2.min.js" > </ script>
<Script type = "text / javascript" src = "../jqwidgets / jqxcore.js" > </ script>
<Script type = "text / javascript" src = "../jqwidgets / jqxchart.js" > </ script>
<Script type = "text / javascript" src = "../jqwidgets / jqxgauge.js" > </ script>
```

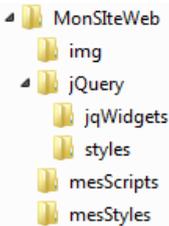
Complétez l'arborescence, ébauchée ci-contre, des répertoires du site.



Exercice 3 : Ecriture de liens relatifs à partir d'une arborescence.

Écrivez, les liens permettant au fichier *index.html* (situé à la racine de MonSiteWeb) d'accéder aux fichiers :

- jquery-1.7.2.min.js situé dans « jQuery »,
- jqxcore.js et jqxchart.js situés dans « jqWidgets »
- jqx.base.css situé dans « styles »



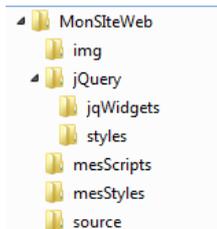
```
<Link rel = "stylesheet" href = "jQuery/ styles / jqx.base.css" type = "text / css" />
<Script type = "text / javascript" src = jQuery / jquery-1.7.2.min.js" > </ script>
<Script type = "text / javascript" src = " jQuery / jqWidgets / jqxcore.js" > </ script>
<Script type = "text / javascript" src = " jQuery / jqWidgets / jqxchart.js" > </ script>
```

Exercice 4 : Ecriture de liens relatifs à partir d'une arborescence.

Le fichier *index.html* de l'exercice précédent a été déplacé dans le répertoire « source ».

Comment doit-on modifier les liens dans ce fichier ?

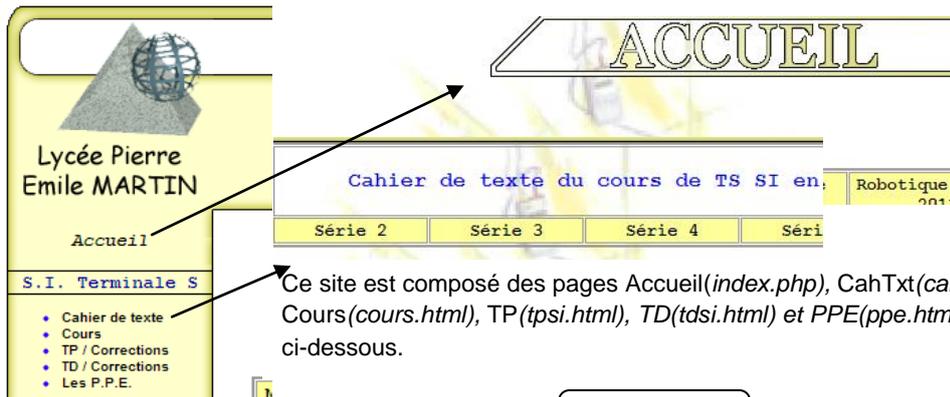
Rajouter ../ au début des liens



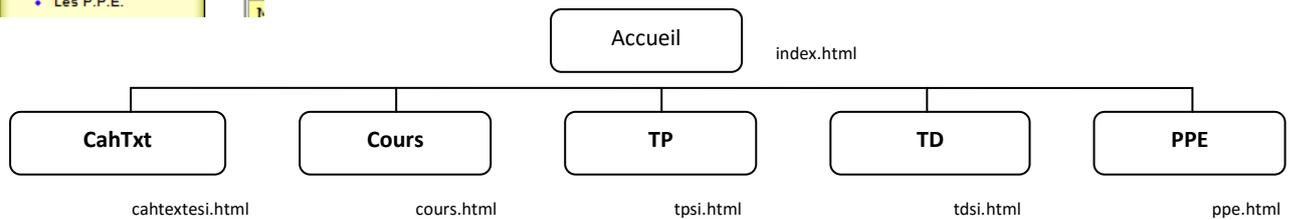
Ex :

```
<Link rel = "stylesheet" href = "../jQuery/ styles / jqx.base.css" type = "text / css" />
```

Exercice 5 : Architecture d'un site existant



Ce site est composé des pages Accueil(*index.php*), CahTxt(*cahtextesi.html*), Cours(*cours.html*), TP(*tpsi.html*), TD(*tdsi.html*) et PPE(*ppe.html*). **Complétez** son architecture ci-dessous.



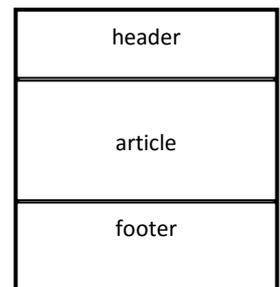
Exercice 6 : balise div et sélecteur id pour créer un **layout**

Cet exercice illustre le principe utilisé pour créer le « layout » des pages d'un site.

On souhaite coder le site dont les pages auront une structure correspondant au **squelette** ci-contre :

Le design des zones *header*, *article* et *footer* est décrit par le fichier *style.css*.

Complétez le fichier *layout.html* ci-dessous pour qu'il traduise le squelette ci-contre. Vous placerez le texte : « En-tête de page, Corps de page et Pied de page » Afin d'identifier les différentes zones de la page.



style.css	layout.html
<pre>body{ background-color: #eee; } #header{ background-color: #aaa; } #article{ background-color: #ccc; } #footer{ background-color: #fff; }</pre>	<pre><!DOCTYPE html> <html> <head> <title>Ma Page</title> <link rel="stylesheet" href="style.css"/> </head> <body> <div id="header"> <p>En-tête de page</p> </div> <div id="article"> <p>Corps de page</p> </div> <div id="footer"> <p> Pied de page </p> </div> </body> </html></pre>

Liens à consulter

L'ASCII et l'Unicode (UTF-8)
UTF-8 avec ou sans BOM

<http://goo.gl/Cvok4a>
<http://goo.gl/eQnCBf>