Technologies du Web



Tutoriel « Conception d'un site Web pour mobile »

CONNAISSANCES ABORDÉES

Architecture client-serveur.

OBJECTIF

Réaliser un site web pour mobile (soigné et fonctionnel) et le rendre accessible sur un réseau local.

CONDITIONS DE DÉROULEMENT DE L'ACTIVITÉ

Etapes	*Objectifs pédagogiques*	*Activités pratiques*
A - Mise en situation	***Comprendre** le cahier des charges.*	***Lecture** de la mise en situation.*
B - Réaliser un site pour mobile	***Concevoir** les pages du site.*	***Écriture** du code HTML5 des pages du site. Test du site avec un navigateur pour mobile.*
C - Synthèse	***Modifier** le site existant et le rendre accessible à un équipement disposant du wifi.*	***Modification** du site. Réglage du serveur et d'un équipement disposant du wifi.*

SIGLES UTILISES

: Durée conseillée	: Important	: Information	: Éditeur de code (voir prof)

PRÉPARATION DU DOSSIER DE TRAVAIL



Organisez votre dossier de travail en suivant les indications données sur la **dernière page** de ce document.

LOGICIELS ET MATERIELS

PC, navigateur (Chrome, Opéra, Mozilla, etc.), éditeur de code, serveur Apache, guide des balises HTML

Table des matières

1. Mise en situation [🕐 10mn] 3
1.1. Introduction 3
1.2. Cahier des charges 3

2. Réaliser un site pour mobile [🕐 2h15] 4
2.1. Code minimum du " gabarit" des pages d'un site pour mobile 4
2.2. Affichage du template des pages du site en "mode mobile" 5
2.3. Analyse du fichier "template.html" 6
2.4. Construction du site 7
2.4.1. Nécessité d'un serveur 7
2.4.2. Etape 1 : codage de la page « Accueil » (fichier index.html) 7
2.4.3. Etape 2 : codage de la page « Biographie » (fichier biographie.html) 8
2.4.4. Etape 3 : codage de la page « Œuvre » (fichier oeuvre.html) 8
2.4.5. Etape 4 : codage de la page « Les aventures de Tintin (1929) » (DANS le fichier oeuvre.html) 9
2.4.6. Etape 5 : codage de la page « Musée » (fichier musee.html) 10
2.4.7. Etape 6 : synthèse - codage de la page « Contact » (fichier "contact.html") 10

3. Synthèse [🕐 20mn] 10

Annexe 1 : Gabarit (*template.html*) des pages du site à réaliser 11
Annexe 2 : Rendu attendu pour chacune des pages 12

Pour aller plus loin… 13
Installation du dossier du TP dans votre compte sur le serveur NasSIN 14
Glossaire 14

Table des illustrations

Figure 1 : Page "Accueil" 3
Figure 2: Géolocalisation 3
Figure 3 : Choix de la cible 5
Figure 4 : Template avec jQuery 5
Figure 5: Template sans jQuery 5
Figure 6 : Style personnalisé 5
Figure 7 : Page "Accueil" 7
Figure 8 : Page "Biographie" 8
Figure 9: Aide en ligne jQuery mobile 8
Figure 10 : Page "Œuvre" partielle 8
Figure 11: Page "Œuvre" 9
Figure 12: Page "Les aventures de Tintin (1929) " 9
Figure 13: Page "Musée" 10
Figure 14: Page "Contact" 10

Dans ce document, vous reverrez les notions abordées précédemment (**balise, attribut**, **sélecteur, classe**) sur les langages **HTML** et **CSS**. Vous appliquerez ces notions à la réalisation d'un **site web pour mobile**. Celui-ci sera déployé sur le **serveur NAS_SIN.** Il sera testé avec un **navigateur** de site pour mobile, un smartphone ou une tablette.

Aucune notion de programmation en JavaScript n'est nécessaire.
Notre objectif étant de réaliser un site **soigné et fonctionnel**, les animations et le rendu des pages seront obtenus en faisant appel à **JQuery Mobile**.

Les compétences que vous allez acquérir seront réinvesties dans **les projets**.

Elles vous permettront également de commencer à développer **un site mobile personnel**.

1. Mise en situation [🕐 10mn]

1.1. Introduction

« Impossible aujourd'hui de négliger l'internet mobile lors de la création d'un outil de communication ! Sa maîtrise est une plus-value, aussi bien pour se vendre que pour vendre ». Webdesign (2012) - Extrait.

Bibliothèque JavaScript

Réaliser une application web pour mobile, c'est-à-dire : une application capable d'accéder à des pages web à partir d'un **écran tactile**, dont les dimensions peuvent varier dans de grandes proportions d'un type d'appareil à l'autre, serait un travail complexe sans l'existence de kits de composants logiciels ou **des bibliothèques** intégrant des fonctionnalités écrites en **JavaScript** [1] et des styles de présentation **CSS**. Les **bibliothèques JavaScript** pour mobiles ont pour nom : Swiper.js, Zepto.js, **JQuery Mobile**, etc.

jQuery Mobile permet aux développeurs de réaliser simplement des **applications web attrayantes** pour mobiles (smartphone, tablette). Il est basé sur JQuery et JQuery UI[2] (deux outils de référence pour le développement d'un site web).



1.2. Cahier des charges

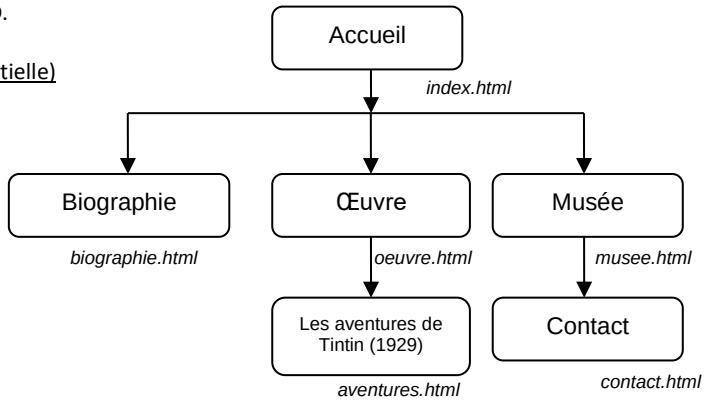


Figure 1 : Page "Accueil"

Le site à réaliser est destiné à des appareils mobiles (smartphones, tablettes).
Il doit permettre de présenter rapidement l'auteur de bandes dessinées « Hergé » (sa jeunesse, son œuvre, la liste des albums de Tintin, etc.). Il doit également permettre de géolocaliser le musée qui lui est consacré.

Nom du site : **WebBD**.

Structure du site (partielle)



Le rendu attendu pour les différentes pages est présenté en **annexe 2** à la fin de ce document.

- La page « ***Accueil »*** (figure 1) contient **trois liens** :
  - Un lien vers la **biographie** de l'auteur,
  - Un lien vers la présentation de son **œuvre**,
  - Un lien vers les informations permettant d'accéder au site du **musée**.

- La page « ***Biographie »*** *(figure 8)* doit être organisée en zones de **texte pliables** (**collapsible**).

- La page « ***Œuvre »*** *(figure 11)* doit afficher une partie de la chronologie de l'œuvre de l'auteur dans des **listes** (**listview**). La liste des albums de Tintin « **Les aventures de Tintin (1929)** » (*figure 12*) doit apparaitre sur une nouvelle page.

- La page « ***Musée »*** *(figure 13)* doit présenter des informations générales et géographiques dans des zones de texte pliables (**collapsible**). Elle contient **deux liens** :
  - Un lien vers le site du musée consacré à l'auteur,
  - Un lien vers les moyens d'accès au musée (plan, mail).

- La page « ***Contact »*** (figure 14) contient **deux liens** :
  - Un lien vers le plan d'accès au musée (**Google Maps),**
  - Un lien permettant d'envoyer un mail au musée.



Figure 2: Géolocalisation

[1] **JavaScript** est le langage de programmation de scripts utilisé dans les pages web interactives (**côté client**) ou côté serveur (**Node.js**). PHP est le langage de programmation de scripts utilisé côté serveur.
[2] **UI** : User interface (interface utilisateur).

2. Réaliser un site pour mobile [🕐 2h15]

Objectif : *Concevoir les pages du site.*

2.1. Code minimum du " gabarit" des pages d'un site pour mobile

Gabarit ou template

Dans cette première partie, vous allez étudier l'organisation de la **page de référence** du site puis vous serez guidé dans sa construction **étape par étape**.

Le site devant être constitué de **plusieurs pages**, nous allons utiliser un **gabarit** (***template.html***). Il suffira de le **dupliquer** et de le compléter pour créer les pages du site.

Le **gabarit** le plus élémentaire, retenu pour les pages du site, correspond au code ci-dessous :

Il est disponible sur le site « JQuery Mobile »

```
<!DOCTYPE html>
<html>
 <head>
        <title>Page Title</title>
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <link rel="stylesheet" href="https://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css" />
        <script src="https://code.jquery.com/jquery-1.11.1.min.js"></script>
        <script src="https://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
 </head>

 <body>
   <div data-role="page">
        <div data-role="header">
                <h1>My Title</h1>
        </div><!-- /header -->

        <div role="main" class="ui-content">
                <p>Page content goes here.</p>
        </div><!-- /content -->

        <div data-role="footer">
                <h1>Page Footer</h1>
        </div><!-- /footer -->
   </div><!-- /page -->
 </body>
</html>
```

meta indispensable pour l'affichage sur un mobile

En-tête de la page

Corps de la page

L'**organisation** de cette page html est plutôt **classique**. Elle se compose d'un **en-tête** <head> et d'un **corps** <body> comme celles que vous avez écrites précédemment.

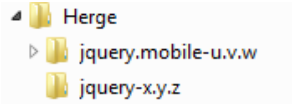
Outre la balise **<title>** que vous connaissez, l'en-tête contient une balise <meta>, une balise <link> et deux balises <script>.



- Sans trop entrer dans les détails, nous dirons que la **balise <meta>** donne des indications au navigateur.
  Dans le cas présent cette balise s'appelle « *viewport* ». Elle précise que la page web est optimisée pour les appareils mobiles ; l'attribut « *width* » affecté de la valeur « *device-width* » indique au navigateur qu'il doit s'adapter à l'écran du mobile ; l'attribut « initial-scale » contrôle le zoom.
- La **balise <link>** permet de lier une feuille de style (.css), spécialement développée pour les appareils mobiles, à la page web.
- Les **balises <script>** permettent aux objets de la page d'avoir accès aux fonctions JavaScript des bibliothèques JQuery et JQuery Mobile.

Remarque : Dans le code ci-dessus, ces balises font référence à des fichiers situés sur le site du projet JQuery. On dispose ainsi de la version actualisée et compressée des fichiers. Cette technique est principalement utilisée "**en production**".

Lors du **développement d'un site** il est préférable de télécharger les fichiers .css et .js à partir du site "JQuery Mobile" pour les installer "**en local**" (sur son PC).
C'est ce qui a été réalisé pour ce TP. Le répertoire du projet **« Herge »** regroupe ces fichiers dans deux sous-répertoires comme sur la copie d'écran ci-contre. (Les chiffres correspondant à u, v, w et x, y, z représentent la version des bibliothèques)



Le fichier « *template.html* », situé dans le répertoire **« Herge »,** contient le code du gabarit des pages du projet. Il a été construit à partir du code de la page précédente. L'en-tête a été modifié afin d'accéder aux fichiers « **en local** ». Des éléments ont été ajoutés pour répondre à notre cahier des charges. Le code de ce template est disponible en **annexe 1** à la fin de ce document.

Le DOCTYPE précise au navigateur la **version** du code **HTML** utilisé. Son écriture, complexe avant HTML 5, se limite aujourd'hui à : **<!DOCTYPE html>** sur chacune des pages du site.

STOP Votre dossier de travail doit être organisé comme demandé sur la **dernière page** de ce document.

Ouvrez le répertoire WEB4 dans **Visual Studio Code et affichez** le code du fichier **« *template.html* ».**

⚠ **Affichez** la page **« *template.html* »** dans un navigateur à l'aide de **IPduServeur/~votrelogin/WEB4/Herge/template.html**

Le mode plein écran du navigateur ne donne pas une bonne représentation du rendu d'un site pour mobile.

Nous allons passer l'écran du navigateur en mode mobile**.**

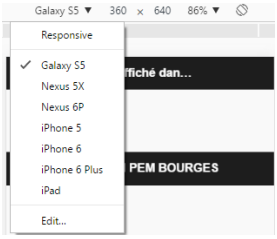


Figure 3 : Choix de la cible

2.2. Affichage du template des pages du site en "mode mobile"

- Dans Mozilla, appuyez sur la touche **F12** pour faire apparaître les outils du navigateur. Cliquez sur (dans la zone des "outils" pour passer l'écran du navigateur en "mode mobile".
- **Sélectionnez** le mobile Samsung Galaxy comme sur la figure 3 ci-contre.
- L'écran du mobile doit ressembler à la figure 4 ci-contre.



Figure 4 : Template avec jQuery

Le design par défaut n'est pas exceptionnel, mais à quoi ressemblerait cette page **sans** les styles CSS et les scripts de JQuery Mobile ?



Figure 5: Template sans jQuery

Pour le savoir, il suffit de placer les lignes sur lesquelles se trouvent les balises <link> et <script> à l'intérieur de balises de commentaires.

- **Entrez** **< !--** avant <link et **-->** après le dernier < /script>.

- **Sauvegardez** le fichier (**ctrl-s**) et actualisez l'affichage dans le navigateur.

Vous devriez obtenir une page ressemblant à la copie d'écran ci-dessus (figure 5). **Pas terrible** !

On comprend mieux la devise du projet **JQuery : « write less, do more ». En effet,** avec JQuery, nous avons un « **Design** » sans écrire une ligne de code CSS !

Supprimez les balises **< !-- -->** écrites précédemment et **sauvegardez** la page.

Remarque : Dans ce TP, nous utiliserons le « **Design par défaut** ».

ThemeRoller

Figure 6 : Style personnalisé

2.3. Analyse du fichier "template.html"

En observant le rendu de la page *template.html* dans le navigateur, nous pouvons identifier **trois zones** :

- **Un en-tête de page,**
- **Un contenu**
- **Un pied de page**

Ces zones étant destinées à **contenir des objets** (boutons, texte, etc.) on les appelle : **boîte (ou conteneur)**.

La page *template.html* est représentée sous la forme d'un **schéma comprenant quatre boîtes (*page, en-tête, contenu* et *pied*)**.



La page dans le navigateur **Maquette graphique**

** data-***

Le fichier *template.html* contient les **balises complétées par des attributs data-*** (ou * est remplacé par un mot reconnu par JQuery Mobile).

Exemple : < !-- Correspond à la définition de la boîte (Page) en trait gras ci-dessus -->

Les balises ** créent les différentes boites** (page, en-tête, contenu et pied). En analysant le contenu de la balise <body> dans le fichier *template.html,* on constate qu'une balise , à laquelle on attribue le rôle de "page", **contient** trois autres dont le rôle est **header** (en-tête de page), **main** (contenu) et **footer** (pied de la page).



<**div**> : balise spécialement conçue pour **introduire une division (boîte ou conteneur sans passer par une balise de paragraphe)**. Cette balise est particulièrement utile pour la déclaration des feuilles de style.
data-* : Cet attribut fait partie des « **custom data attributs** » introduits par **HTML5**. Ils permettent d'ajouter une donnée ou une information sur l'élément HTML sans qu'elle soit vue par l'utilisateur. Ce sont ces données que JQuery Mobile utilise pour transformer la page web en une application mobile.

Les termes "role", "page", "header", "main" et "footer" sont reconnus par JQuery Mobile pour appliquer des styles à la page. Vous pouvez faire beaucoup avec jQuery Mobile : ajouter des listes, mettre des boutons, des formulaires, etc. Tout passe par les attributs **data-***. Leur description est disponible sur le site de JQuery Mobile.

Résumé et rappels

La réalisation du site va être grandement facilitée par l'utilisation de **JQuery et JQuery Mobile**. Les différents **comportements et rendus** attendus seront obtenus sans écrire une ligne de code **JavaScript** puisqu'il suffira de "décorer" les balises ** ** (divisions appelées blocs ou **conteneur**), assurant la structure de la page, avec des attributs **data-***.

Seul le concept d'attribut, vu dans les précédents TP doit avoir été bien compris !

Le bon usage des balises (règles définies en xHTML et reprises par HTML5 **)**

- **Règle 1** : Ecrire les balises en **minuscule**.
- **Règle 2** : Toutes les balises **ouvertes** par **<***nombalise***>** doivent être **fermées** par **</***nombalise***>** 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 , et sont **indentées.**

```
<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">

Il est temps de commencer à coder…

2.4. Construction du site

2.4.1. Nécessité d'un serveur

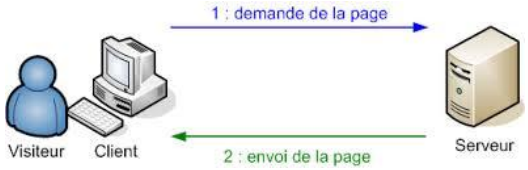
Pour des **raisons de sécurité**, les navigateurs **n'exécutent pas les liens gérés par jQuery** si les pages qui les contiennent ne se situent pas sur un serveur web.

Le répertoire **www** du **NAS SIN** contient un **serveur web** ! Votre code ayant été placé dans ce répertoire, il s'exécutera correctement si vous y accédez avec l'url :

http://IPduServeur/~votrelogin/chemin/ressource

Serveur Web

Un **serveur web** est un logiciel installé sur un ordinateur permettant à des **clients** d'accéder à des **pages web**. Ces pages sont créées par un navigateur (browser) à partir de fichiers au format HTML.
Les principaux serveurs web sur le marché sont : **Apache** et Microsoft IIS



Apache

Le logiciel libre Apache HTTP Server (**Apache**) est un **serveur HTTP** créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache

Le serveur web Apache a été créé en **1995** par un groupe de huit développeurs dont **Brian Behlendorf** et **Cliff Skolnick** à partir du code du serveur HTTP NCSA HTTPd développé par **Rob McCool** à l'université de l'Illinois. Depuis 1999 il est développé sous la direction de la fondation « Apache Software Foundation » créé par l'équipe Apache.

Apache fonctionne principalement sur les systèmes d'exploitation **UNIX** (Linux, Mac OS X, Solaris, BSD et UNIX) et **Windows**.

2.4.2. Etape 1 : codage de la page « Accueil » (fichier index.html)

Format agrandi disponible en annexe 2.



lien vers la page « Biographie » décoré avec un aspect « **button** »

Figure 7 : Page "Accueil"

Pour l'instant, le contenu de "*index.html*" est identique à celui de "*template.html*".

Vous allez modifier *index.html* afin d'obtenir l'affichage ci-contre.

Revoyez la description de la page « Accueil » dans le **cahier des charges**.

Indications

- La **taille du texte** se règle avec des balises **<hx></hx>** (x ∈[1,6] ⇒ la valeur de x **règle la taille du texte**).
- Les paragraphes sont obtenus en utilisant des balises ** **.
- Pour transformer **un lien** en **bouton,** on utilise les propriétés **data-role="button"** ou **class="ui-btn ui-corner-all ui-shadow".**

Exemples :
Biographie ou
Biographie

Activité 1 : Modifiez le fichier *index.html* pour que la page « Accueil » corresponde à la copie d'écran ci-dessus (**figure 7**).
Testez-le dans le navigateur.

Appel prof **Pour faire vérifier la justesse du code**

2.4.3. Etape 2 : codage de la page « Biographie » (fichier biographie.html)

La page **« Biographie »** présente **deux nouveautés** par rapport à la page « Accueil » :

- Une icône (home), située dans l'en-tête (un lien vers *index.html*), permet à l'utilisateur de revenir sur la page « Accueil »,
- Trois zones de **texte pliables** (**collapsible**) optimisent la place occupée sur l'écran.

Faire une copie du fichier **« *template.html* »** dans l'éditeur et la **sauvegarder** sous le nom **« *biographie.html* ».**

Format agrandi disponible en annexe 2.

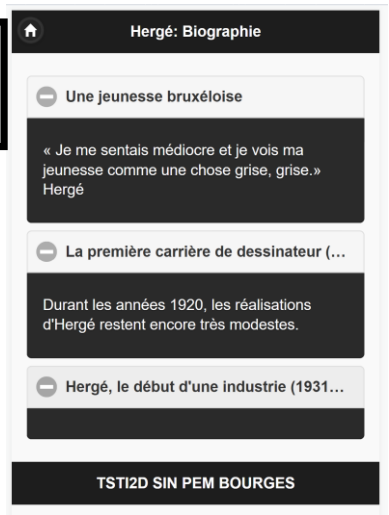


Figure 8 : Page "Biographie"

Indications

- Pour placer l'icône , il suffit d'entrer le code ci-dessous :

```
<a href="index.html" data-theme="d" data-icon="home" data-iconpos="notext">home</a>
```

- Pour créer la première zone de **texte pliable,** il suffit d'entrer le code ci-dessous :

```
<div data-role="collapsible" data-theme="d" data-content-theme="b">
    <h3>Une jeunesse bruxéloise</h3>
    <p>« Je me sentais médiocre et je vois ma jeunesse comme une chose grise, grise.» Hergé </p>
</div><!-- /collapsible -->
```

Documentation

La documentation sur le site **JQuery Mobile est accessible ici** : https://jquerymobile.com/demos/

Utilisez la version jQuery Mobile 1.4.5 Demos .

Consultez la page associée au widget « **Collapsible** ». Vous devez retrouver le code proposé dans les indications ci-dessus.

La documentation en ligne vous donne toutes les informations utiles au codage de vos pages !

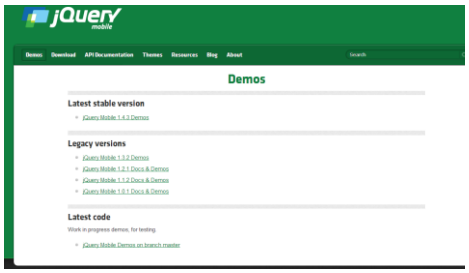


Figure 9: Aide en ligne jQuery mobile

Activité 2 : Modifiez le fichier *biographie.html* pour que la page « Biographie » corresponde à la copie d'écran ci-dessus (**figure 8**). **Testez-le** dans le navigateur.

Appel prof **Pour faire vérifier la justesse du code**

2.4.4. Etape 3 : codage de la page « Œuvre » (fichier oeuvre.html)

La page **« Œuvre »** est composée de **listes à puces**. La décoration de ces listes par le jQuery produit l'affichage ci-contre (fig10).

<ul>
<li>

Dans un premier temps, vous allez découvrir le comportement des listes à puce, et voir ou revoir l'utilisation des balises : <**ul> et **.

Vous les utiliserez ensuite avec ***jQuery mobile*** pour coder la page **« Œuvre »**.

Liste à puces

Il est souvent nécessaire de **mettre en forme des listes d'éléments**. La **liste à puce** permet de faire une énumération.

Deux balises sont nécessaires pour obtenir une liste à puce et . **
** ouvre la liste et **
** la ferme.

Chaque élément de la liste doit être encadré par ** **

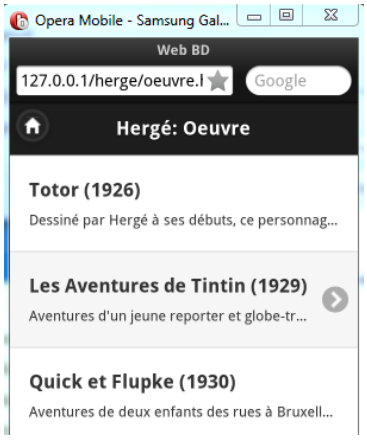


Figure 10 : Page "Œuvre" partielle

Exemple : La page *œuvre.html* va être présentée sous la forme d'une **liste à puces** comme ci-dessous.

Code HTML	**Résultat dans un navigateur sans jQuery**
** ** ** **Totor (1926)** ** ** **Les Aventures de Tintin (1929)** ** ** **Quick et Flupke (1930)** ** ** **	• Totor (1926) • Les Aventures de Tintin (1929) • Quick et Flupke (1930)

Avec les balises HTML seules, le rendu est "minimum" ! Là encore, jQuery Mobile assure un certain « **Design » à la page**.

Faire une copie du fichier **« *template.html* »** dans l'éditeur et la **sauvegarder** sous le nom **« *oeuvre.html* ».**



Format agrandi disponible en annexe 2.

Figure 11: Page "Œuvre"

Indications

Pour présenter le premier élément sous la forme d'une liste, il suffit d'entrer le code ci-dessous :

```
<ul data-role="listview">
    <li>
        <h3>Totor (1926)</h3>
        <p>Dessiné par Hergé à ses débuts, ce personnage est
       l'ancêtre de Tintin.</p>
    </li>
    <li>
      …
    </li>
</ul>
```

Remarques : Le deuxième élément doit contenir un lien (#page2) vers une page interne au fichier *oeuvre.html*. Le bouton est automatiquement créé. ()

Consultez la documentation en ligne (Listview -> Linked)**. La page "page2" sera développée à partir des indications de l'étape 4.**

Activité 3 : Modifiez le fichier "*oeuvre.html*" pour que la page « Œuvre » corresponde à la copie d'écran ci-dessus (**figure 11**). **Testez-le** dans le navigateur. N'oubliez pas le lien vers la page *« Accueil » (fichier index.html)*.

2.4.5. Etape 4 : codage de la page « Les aventures de Tintin (1929) » (DANS le fichier oeuvre.html)

La réalisation de la page **« Les aventures de Tintin (1929) »** nécessite de **reprendre la démarche adoptée précédemment**. Il est donc nécessaire d'avoir bien compris les activités précédentes !

Créez une deuxième page après la page 1 dans le fichier œuvre.html (attribuez-lui l'id="page2").



Figure 12: Page "Les aventures de Tintin (1929) "

```
<div data-role="page" id="page2">

        // Zone à compléter

</div>
```

Activité 4 : Modifiez le fichier "*oeuvre.html*" pour que la page « Les aventures de Tintin (1929) » corresponde à la copie d'écran ci-dessus (**figure 12**).

Testez-le dans le navigateur.

Appel prof **Pour faire vérifier la justesse du code**

2.4.6. Etape 5 : codage de la page « Musée » (fichier musee.html)

La réalisation de la page **« Musée »** nécessite de **reprendre la démarche adoptée pour une partie du travail réalisé précédemment**. En effet, il s'agit maintenant de présenter des **listes à puces à l'intérieur de zones pliables** et d'assurer les liens vers le site du musée et vers la page « Contact » (fichier *contact.html)*.

Faire une copie du fichier **« *template.html* »** dans l'éditeur et la **sauvegarder** sous le nom **"*musee.html*"**

Format agrandi disponible en annexe 2.

Liste à puce contenue dans une zone pliable !

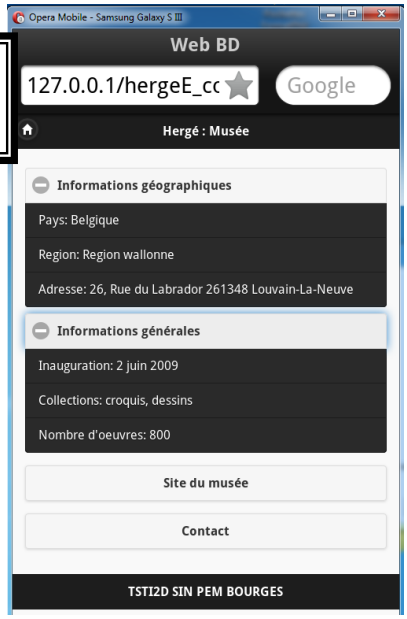


Figure 13: Page "Musée"

Revoyez la description de la page « Musée » dans le **cahier des charges**.

Indications

L'url du musée Hergé est : https://museeherge.be/

Pour réaliser le lien vers la page contact, il faut entrer le code ci-dessous :

Contact

Remarque : data-transition = "pop" permet de réaliser une **pop-up**.

Activité 5 : Modifiez le fichier "*musee.html*" pour que la page « Musée » corresponde à la figure 13. **Testez-le** dans le navigateur.

N'oubliez pas le lien vers la page « Accueil ».

Appel prof **Pour faire vérifier la justesse du code.**

2.4.7. Etape 6 : synthèse - codage de la page « Contact » (fichier "contact.html")



Figure 14: Page "Contact"

On donne ci-dessous (figure 14) la représentation de la **pop-up** « **Contact** » (fichier *contact.html*).

Pour information

- La liste des icônes est sur la page « Icons » du site jQuery mobile.
- Liens pour le plan d'accès : https://edurl.fr/11zRhs33
- Le mail : href="mailto:info@museeherge.com?subject=Demande d'informations"

Activité 6 : Faire le lien de la page « **Contact** ».

Appel prof **Pour faire vérifier la justesse du code.**

3. Synthèse **[🕐 20mn]**

***Objectif** : réaliser une version du site dans un seul fichier.*

Activité 7 : dupliquez *index.html* sous ***index2.html*** et **créez** dans ce fichier une nouvelle version du site contenant toutes les pages. Les liens entre les pages doivent être des **liens d'ancrage** (comme celui réalisé dans la page œuvre). Le fichier ne doit pas contenir de **liens internes**. **Testez** l'affichage de cette nouvelle version sur un mobile.

Appel prof **Pour faire vérifier la justesse du code.**

Annexe 1 : Gabarit (*template.html*) des pages du site à réaliser

```
<!DOCTYPE html>
<html>
<head>
    <title>Ma page</title>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1 /">

    <!-- Liens pour le développement en local -->
    <link rel="stylesheet" href="jquery.mobile-1.4.5/jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1/jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5/jquery.mobile-1.4.5.min.js"></script>
</head>
<body>

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

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

    <div data-role="footer" data-theme="b" data-position="fixed">
        <h4>TSTI2D SIN PEM BOURGES</h4>
    </div><!-- /footer -->
</div><!-- /page -->

</body>
</html>
```

Annexe 2 : Rendu attendu pour chacune des pages

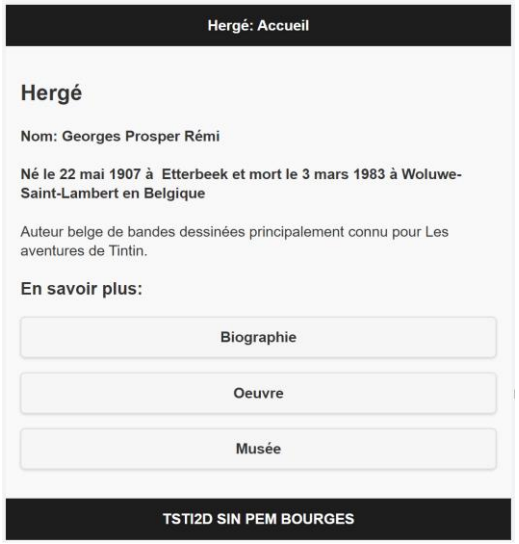


Figure 1 et Figure 7

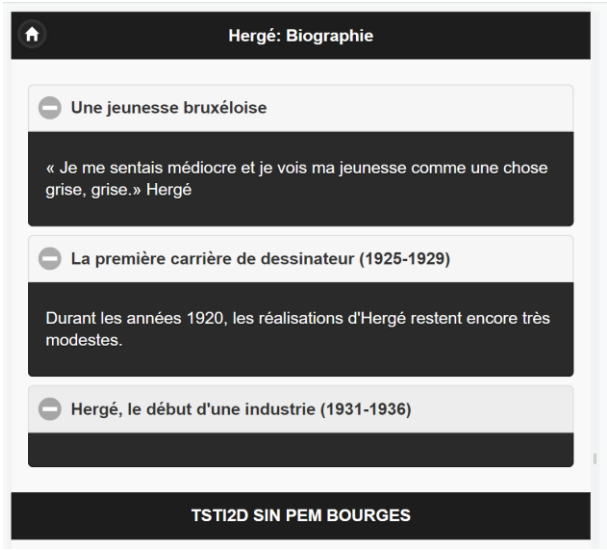


Figure 8

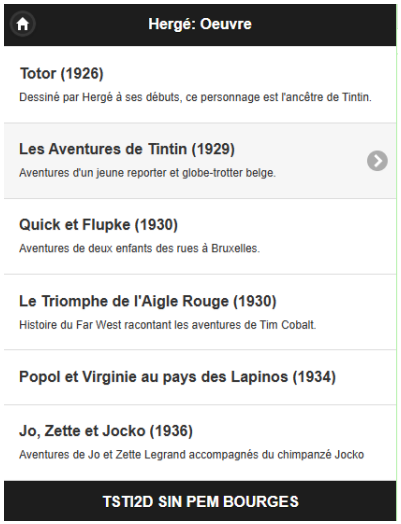


Figure 11



Figure 12



Figure 13



Figure 14

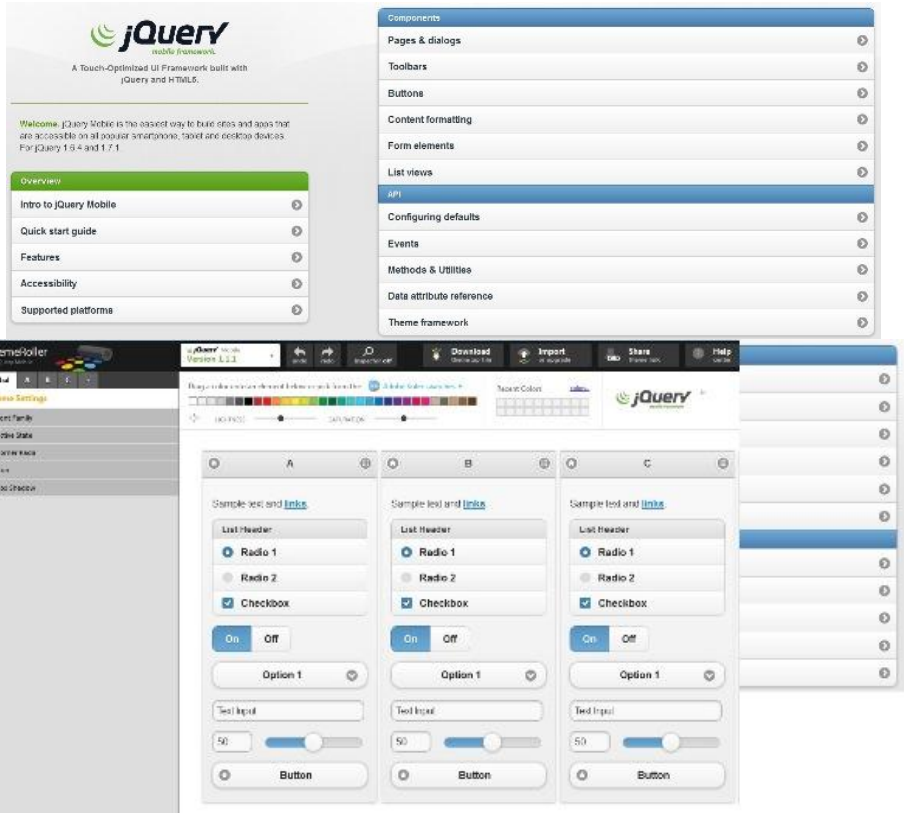


Pour aller plus loin….

Informations complémentaires (pour améliorer et personnaliser le projet)

La documentation JQuery mobile est disponible à l'adresse :

https://jquerymobile.com/



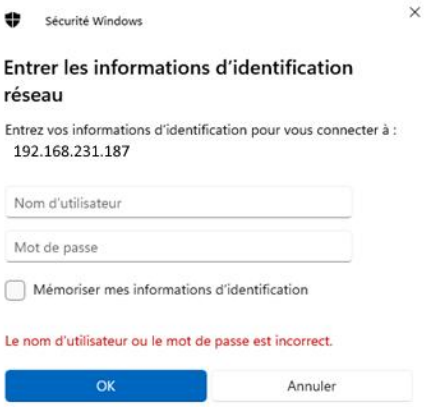
Le générateur de styles JQuery mobile permet de personnaliser les différents thèmes d'un site mobile à l'aide d'une interface graphique.

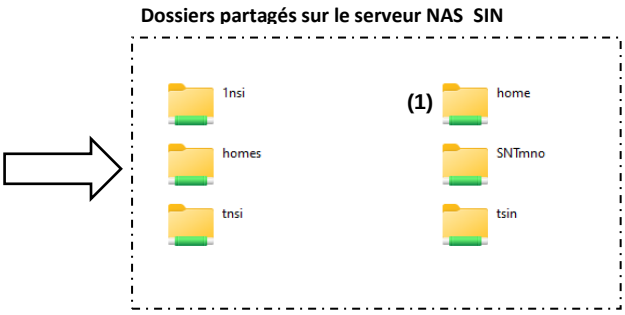
https://themeroller.jquerymobile.com/

Installation du dossier du TP dans votre compte sur le serveur Nas_SIN

1. Connexion au serveur NAS_SIN pour accéder au dossier partagé du cours.

- Sélectionnez  et entrez **\\192.168.231.187** dans l'invite de commande. 
- Complétez la fenêtre de sécurité Windows représentée ci-dessous avec l'**identifiant** et le **mot de passe** de votre compte sur le serveur **NAS_SIN pour accéder aux dossiers partagés.**



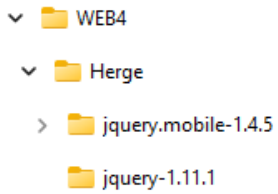


(1) Votre dossier personnel

- Double-cliquez sur le dossier **du cours.**

2. Dossier à utiliser dans le TP

Copiez le dossier **WEB4** situé dans ***DOSSIER_du_COURS/5_IHM_WEB/*** et collez-le dans ***home/www/***.



Glossaire

Gabarit
Un **gabarit**, souvent nommé à tort avec les anglicismes **template** ou **layout**, 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)

UTF-8 (*UCS transformation format* 8 bits) est un codage de caractères informatique conçu pour coder l'ensemble des caractères internationaux d'Unicode en restant compatible avec la norme ASCII limitée à l'anglais, mais très largement répandue depuis des décennies. **(**Wikipédia)

Lien interne, Lien externe
Les liens internes pointent vers une ressource du même site, les liens externes vers une ressource d'un autre site. Dans les deux cas, c'est la balise <a> avec l'attribut href, mais l'écriture de l'adresse diffère.
Exemples
<!--lien interne --> Contact
<!--lien externe --> Arduino sur Wikipédia
