

Chapitre 2 : Utilisation du framework JQuery

JQuery est un framework écrit en JavaScript, dont la devise est « *jQuery. Write Less, Do More* ». Cette "trousse à outils" permet non seulement de manipuler les éléments du DOM des pages web mais également d'interagir avec des éléments d'interface. Les utilisateurs pourront par exemple déplacer des éléments, afficher des calendriers interactifs, saisir des formulaires avec une gestion immédiate des erreurs, utiliser des galeries d'images, choisir des variantes de produits dans des boutiques en ligne... Les exemples sont innombrables.

Le framework JQuery propose deux bibliothèques principales : JQuery et JQuery UI.

- **JQuery** offre aux développeurs une **API** (*Application Programming Interface*) de programmation, constituée de méthodes, d'événements, d'effets, qui va permettre de développer des applications web dynamiques et interactives. Les développeurs vont pouvoir s'appuyer sur JQuery pour développer et proposer leurs propres plugins. Actuellement il existe des milliers de plugins JQuery interactifs.
- **JQuery UI** (*User Interface*) offre des composants d'interface (des **widgets**) prêts à l'emploi : comme des accordéons, des panneaux à onglets, des sélecteurs de dates, des boutons radio, des cases à cocher...

En résumé, le framework JQuery permet :

- de manipuler le DOM très facilement,
- de gérer des événements,
- de créer des effets graphiques,
- et d'implémenter des moteurs Ajax avec une syntaxe très simple et concise.
Exécuter un script PHP et récupérer le résultat en JSON et modifier le DOM

I) Comment utiliser le framework jQuery ?

Pour profiter de jQuery, il faut lier toutes les pages web de votre site à la bibliothèque. Pour cela deux solutions:

- Télécharger la bibliothèque dans le dossier de votre site et puis lier chacune des pages au fichier JavaScript de jQuery.
- Lier chacune des pages de votre site à la bibliothèque jQuery en ligne, sur le site de jQuery ou sur d'autres serveurs CDN (**Content Delivery Network**.)

II) La classe jQuery

La plupart des instructions jQuery commencent par la création d'un objet jQuery auquel on appliquera par la suite une ou plusieurs méthodes.

L'instanciation d'un objet jQuery peut être réalisée avec la syntaxe suivante :

```
jQuery()
```

Cependant, en pratique c'est l'alias de l'appel à la classe jQuery qui est utilisé :

```
$()
```

a) Les sélecteurs

C'est l'objet jQuery qui va permettre de sélectionner les éléments du DOM sur lesquels on souhaite agir. Pour cela il suffit d'insérer entre les parenthèses un sélecteur semblable à ceux utilisés pour le CSS pour définir l'élément (ou les éléments) désiré(s).

Exemple :

Objectif	Sélecteur jQuery
Sélectionner toutes les balises <p> de la page	\$('p')
Sélectionner la classe « menu »	\$('.menu')
Sélectionner la balise dont l'identifiant est « info »	\$('#info')

b) Les méthodes

Une fois l'objet jQuery créé et l'élément sélectionné, il est possible d'appliquer les méthodes de la classe jQuery avec la syntaxe pointée :

\$(sélecteur).NomDeLaMethode()

Exemple : la méthode `attr()` permet à la fois de récupérer un l'attribut d'une balise et de le redéfinir.

```
var cheminImage = $('img').attr('src'); // récupération du chemin d'une image
$('img').attr('src', 'nouveauChemin/photo.png');
```

Attention cependant, par souci de performance, les multiples appels d'un même objet jQuery sont à éviter, car ils sont gourmands en mémoire et les internautes risquent de ne pas apprécier si l'attente se révèle trop longue.

Pour pallier ce problème, il est possible de donner un objet en tant que paramètre à la méthode `attr()`, qui contiendra chaque attribut à modifier avec leurs nouvelles valeurs respectives.

Exemple :

```
// mauvaise méthode
$('img').attr('src', 'nouveauChemin/photo.png');
$('img').attr('alt', 'Ceci est une nouvelle photo');
$('img').attr('title', 'Nouvelle photo');

// bonne méthode
$('img').attr({
  src : 'nouveauChemin/photo.png',
  alt : 'Ceci est une nouvelle photo',
  title : 'Nouvelle photo'
});
```

Quelques exemples de méthode, bien utiles :

- **La méthode « html » pour les balises html**

Exemples d'utilisation de la méthode	Nom de ma méthode
<pre>var recupHTML=\$('#info').html() ; \$('#info').html('Voici son nouveau contenu');</pre>	La méthode <code>html</code> permet ici de récupérer et de modifier le contenu HTML de la balise englobante nommée <code>info</code>

- **La méthode « val » pour les balises zone de texte**

Exemples d'utilisation de la méthode	Nom de ma méthode
<pre>var recupContenu=\$('#zoneTexte').val() ; \$('#zoneTexte').val('Nouveau contenu') ;</pre>	La méthode val permet de récupérer et de modifier le contenu d'une zone de texte

- **Les méthodes « addClass » et « removeClass » pour l'ajout et la suppression de classes css sur un composant**

Exemples d'utilisation de la méthode	Nom de ma méthode
<pre>\$('#info').addClass('important') ; \$('#info').removeClass('important') ;</pre>	La méthode addClass permet d'appliquer la classe CSS qui s'appelle « important » au composant info. La méthode removeClass permet de la supprimer.

III) La gestion des événements

Comme vu précédemment, jQuery permet non seulement de manipuler le DOM mais également de configurer des gestionnaires d'événements.

Cependant avant d'appliquer les méthodes jQuery aux éléments du document, il convient de s'assurer que le DOM est complètement chargé.

Pour cela jQuery propose la méthode ready() qui permet de vérifier que le DOM est intégralement chargé. Cette méthode permet de détecter la fin du chargement des éléments du DOM sans attendre le chargement des images ou autres composants qui pourraient freiner leur configuration. Cette méthode s'applique par contre à l'élément document comme l'illustre le code ci-dessous :

```
$(document).ready(function(){
```

```
// mettre ici le code jQuery à tester.
```

```
});
```

a) Les événements

Parmi les nombreux événements disponibles, les plus célèbres sont sans aucun doute ceux pouvant être déclenchés par l'utilisateur. Ce sont eux qui permettent une très grande interactivité avec l'internaute et qui sont les plus utilisés en général.

Voici la liste des événements associés à la souris :

Action	Fonction
Clic	click()
Double-clic	dblclick()
Passage de la souris	hover()
Point d'entrée dans un élément	mouseenter()
Quitter un élément	mouseleave()
Presser un bouton de la souris	mousedown()
Relâcher un bouton de la souris	mouseup()
Scroller (utiliser la roulette)	scroll()

Voici une liste non exhaustive des événements jQuery associés au clavier :

Action	Fonction
Action activée lorsqu'une touche du clavier est enfoncée	<code>keydown()</code>
Action exécutée lorsqu'on maintient une touche enfoncée	<code>keypress()</code>
Action qui se lance lorsqu'on relâche une touche préalablement enfoncée	<code>keyup()</code>

Pour finir sur les événements, les éléments de formulaire possèdent eux aussi leur lot d'événements associés. Par exemple, lorsqu'un champ est focalisé, on dit qu'il « prend le *focus* », car c'est sur lui que se trouve notre pointeur. Un événement permet de contrôler ce genre de chose.

Action	Fonction
Focalisation	<code>focus()</code>
Sélection (point d'entrée dans une liste)	<code>select()</code>
Changement de valeur	<code>change()</code>
Envoi du formulaire	<code>submit()</code>

L'exemple ci-dessous illustre l'appel de l'événement `click()` pour la balise dont l'identifiant est `monLien` :

```
$(document).ready(function() {
    $("#monLien").click(function() {
        window.alert("Merci d'avoir cliqué sur ce lien") ;
    }) ;
}) ;
```

b) Les gestionnaires d'événements

En jQuery, et même plus globalement en JavaScript, on peut faire appel aux **gestionnaires d'événements**. Ce sont des fonctions auxquelles on donne un type d'événement à écouter, ainsi qu'une fonction à exécuter à chaque fois que l'événement est déclenché.

La création de gestionnaires d'événements est possible grâce à la méthode `on()`. Cette méthode s'applique sur l'objet jQuery créé auquel on passe en paramètre deux arguments : le type (ou les types) d'événement puis la fonction de retour à exécuter.

Exemple :

```
$('button').on('click dbclick', function() {
    window.alert('Ce code fonctionne !');
});
```

Il est également possible d'associer un traitement différent pour chaque événement comme indiqué dans l'exemple ci-dessous :

```

$('button').on({
  click : function(){
    window.alert('Vous avez cliqué !');
  },
  mouseup : function(){
    window.alert('Vous avez relâché le clic !');
  }
});

```

Faire l'application 1 Sur feuille ou machine

IV) La création de moteurs Ajax

AJAX (Asynchronous JavaScript and XML) désigne une combinaison de technologies (XHTML, DOM, XML, JavaScript et plus particulièrement son objet XmlHttpRequest).

Les applications Ajax permettent de disposer de fonctionnalités avancées mais aussi d'améliorer l'interactivité et l'ergonomie des interfaces Web. L'auto complétion ou le contrôle en temps réel des données de formulaire, l'introduction de Widget sont possibles grâce à la technologie Ajax.

Les applications Ajax se caractérisent principalement par un nouveau mode d'échange de données entre le navigateur et le serveur Web. Dans les sites traditionnels, l'envoi d'une requête vers le serveur impose au navigateur d'attendre sa réponse, le privant du coup de tout type d'activité pendant ce délai. C'est ce que l'on appelle un transfert synchrone. Les applications Ajax permettent au contraire, d'émettre une requête et d'en réceptionner la réponse d'une manière différée (transfert Asynchrone) sans interrompre l'activité de l'utilisateur. En plus de cet avantage, la réponse du serveur ne contient que les données sollicitées par sa requête et non toute la page HTML comme c'est le cas lors d'une réponse classique.

Le principe de fonctionnement de la technologie Ajax est simple et il est résumé dans les grandes lignes ci-dessous :

- Ecoute de l'évènement, clic par exemple, sur le bouton d'envoi.
- Envoi d'une requête HTTP en POST vers un script PHP depuis jQuery.
- Exécution du script PHP qui renvoie un résultat dans un format léger (JSON ou XML).
- Ajout de ce résultat au DOM avec jQuery.

L'utilisation de la technologie AJAX nécessite l'instanciation d'un objet XmlHttpRequest. Cet objet est créé implicitement lors de l'appel à la méthode ajax() de la bibliothèque jQuery. Deux syntaxes sont possibles pour appeler cette méthode.

jQuery.ajax({...}) ;

ou

\$.ajax({...}) ;

Les accolades contiennent les options nécessaires à la configuration de la méthode

Quelques options de la méthode ajax() sont présentées ci-dessous :

- type: pour indiquer la méthode http utilisée pour la requête comme POST ou GET. La valeur par défaut est GET
- url : pour nommer le fichier qui assurera le traitement côté serveur. C'est la seule option obligatoire.
- data : pour stocker les données au format d'url, à envoyer avec la requête

- `dataType` : pour indiquer le type des données renvoyé par le serveur. Les types possibles sont : json, xml, html, text ou script.
- `success`: pour stocker le nom de la fonction javascript appelée en cas de succès
- `error`: pour stocker le nom de la fonction javascript à exécuter en cas d'erreur

Exemple : Insertion d'un étudiant

```
function inserer(){
$.ajax({                                     // appel à la méthode ajax sur l'objet jquery instancié par $
  type:'POST',                             // les données seront envoyées selon la méthode POST
  url: './insererEtudiant.php',            // composant PHP à exécuter
  data: "nom="+$('#txtnom').val()+"&prenom="+$('#txtprenom').val(), // chaine à envoyer au
  // composant format classique HTML
  dataType:'json',                         // on récupèrera le résultat au format JSON
  success: actualiserPage,                 // si succès appel de la fonction actualiserPage
  error: function(){alert('Erreur Serveur');} // fonction anonyme
});
}
```

**Faire l'application 2
Sur feuille ou machine**

IV) La création d'effet graphiques grace à JQUERY UI

JQuery UI (UI pour *User Interface*) propose de nombreux widgets pour enrichir l'interface visuelle de vos pages ou applications HTML comme les fenêtres de dialogue, les infobulles, les menus en accordéon, les menus avec onglets, les boutons, le glisser et le déposer (*drag/drop*), le redimensionnement et la réorganisation des éléments.

Outre ces éléments devenus classiques, jQuery UI intègre également les nouvelles balises du HTML5 avec les curseurs, les calendriers, les barres de progression, les compteurs numériques et les formulaires avec suggestion. Ces widgets permettent d'incorporer dans la conception des pages HTML des balises HTML5 innovantes qui deviennent ainsi enfin opérationnelles pour tous les navigateurs du marché. Même les plus anciens !

En raison du nombre et de la richesse de ces modules externes proposés, jQuery UI se positionne comme le complément indispensable de jQuery. Que vous souhaitiez des applications web hautement interactives ou que vous désiriez simplement ajouter un widget comme un calendrier, jQuery UI s'impose comme une excellente solution.

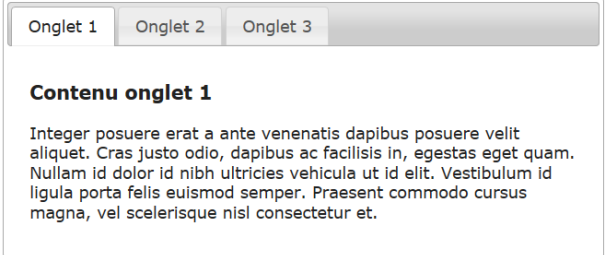
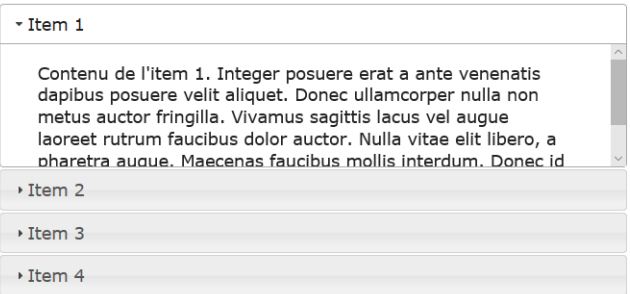
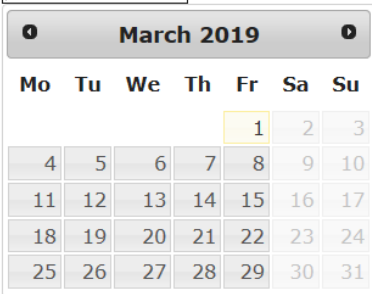
a) Installation

JQuery UI (<http://jqueryui.com/>) est une extension du framework jQuery dédiée à l'interface de vos applications. Ainsi, jQuery UI a besoin de jQuery pour fonctionner. Cette couche de jQuery prend en charge toute la gestion des fonctionnalités JavaScript mises en œuvre par les multiples widgets introduits par jQuery UI.

Exemple :

```
<script src="http://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
```

b) Quelques exemples de widgets

Widget	Méthode JQuery	Visuel
Le menu avec des onglets	Le menu par onglets est une option de mise en page qui permet de regrouper l'information d'un site par sujets ou par thèmes. C'est ce que réalise la méthode jQuery UI tabs().	 <p>Contenu onglet 1</p> <p>Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Nullam id dolor id nibh ultricies vehicula ut id elit. Vestibulum id ligula porta felis euismod semper. Praesent commodo cursus magna, vel scelerisque nisl consectetur et.</p>
Le menu en accordéon	La présentation dite en accordéon regroupe du contenu dans des panneaux séparés qui peuvent être ouverts ou fermés par l'utilisateur. Ce qui permet un gain de place appréciable en maximisant la quantité d'informations disponibles. Les accordéons sont mis en place par la fonction jQuery UI accordion().	 <p>Item 1</p> <p>Contenu de l'item 1. Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Donec ullamcorper nulla non metus auctor fringilla. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Nulla vitae elit libero, a pharetra augue. Maecenas faucibus mollis interdum. Donec id</p> <p>Item 2</p> <p>Item 3</p> <p>Item 4</p>
Les calendriers	<p>La méthode jQuery UI à utiliser est datepicker(). Par défaut, le calendrier affiche les mois et jours en anglais. Il est nécessaire pour nous francophones de faire appel à un script supplémentaire pour un affichage en français.</p> <pre>\$("#calendrier").datepicker({ firstDay: 1, beforeShowDay: \$.datepicker.noWeekends });</pre> <p>Les widgets de jQuery UI comportent de nombreuses options de configuration. Ainsi dans ce script firstDay: 1 permet de faire débuter la semaine le lundi et beforeShowDay: \$.datepicker.noWeekends d'exclure de la sélection les samedis et dimanches.</p>	<p>Date : <input type="text"/></p> 

Exemple 1 : menu avec des onglets

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>jQuery UI</title>
<link rel="stylesheet" href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css">
<script src="http://code.jquery.com/jquery-3.1.0.js"></script>
<script src="http://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
<script>
    $(document).ready(function(){
        $("#onglets").tabs();
    });
</script>
</head>
<body>
<div id="onglets">
    <ul>
        <li><a href="#onglet-1">Onglet 1</a></li>
        <li><a href="#onglet-2">Onglet 2</a></li>
        <li><a href="#onglet-3">Onglet 3</a></li>
    </ul>
    <div id="onglet-1">
        <h3>Contenu onglet 1</h3>
        <p>Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Cras justo odio, dapibus ac
facilisis in, egestas eget quam. Nullam id dolor id nibh ultricies vehicula ut id elit. Vestibulum id ligula porta felis euismod
semper. Praesent commodo cursus magna, vel scelerisque nisl consectetur et.</p>
    </div>
    <div id="onglet-2">
        <h3>Contenu onglet 2</h3>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>
    </div>
    <div id="onglet-3">
        <h3>Contenu onglet 3</h3>
        <p>Etiam porta sem malesuada magna mollis euismod. <p>
    </div>
</div>
</body>
</html>
```

Exemple 2 : menu en accordéon

```
<!doctype html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>jQuery UI</title>
<link rel="stylesheet" href="http://code.jquery.com/ui/1.12.0/themes/smoothness/jquery-ui.css">
<script src="http://code.jquery.com/jquery-3.1.0.js"></script>
<script src="http://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
<script>
    $(document).ready(function(){
        $("#accordeon").accordion();
    });
</script>
</head>
<body>
<div id="accordeon">
    <h3><a href="#">Item 1</a></h3>
    <div>Contenu de l'item 1. Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Donec ullamcorper
nulla non metus auctor fringilla. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Nulla vitae elit
```



```
libero, a pharetra augue. Maecenas faucibus mollis interdum. Donec id elit non mi porta gravida at eget metus. Etiam porta
sem malesuada magna mollis euismod.</div>
<h3><a href="#">Item 2</a></h3>
<div>Contenu de l'item 2. Curabitur blandit tempus porttitor. Aenean eu leo quam. Pellentesque ornare sem
lacinia quam venenatis vestibulum.</div>
<h3><a href="#">Item 3</a></h3>
<div>Contenu de l'item 3. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. </div>
<h3><a href="#">Item 4</a></h3>
<div>Contenu de l'item 4. Nullam id dolor id nibh ultricies vehicula.</div>
</div>
</body>
</html>
```

Exemple 3 : Utilisation d'un datepicker

```
<!doctype html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>jQuery UI</title>
<link rel="stylesheet" href="http://code.jquery.com/ui/1.12.0/themes/smoothness/jquery-ui.css">
<script src="http://code.jquery.com/jquery-3.1.0.js"></script>
<script src="http://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
<script src="jquery.ui.datepicker-fr.js"></script>
<script>
$(document).ready(function(){
    $("#calendrier").datepicker({
        firstDay: 1,
        beforeShowDay: $.datepicker.noWeekends
    });
});
</script>
</head>
<body>
<p><label for="calendrier">Date :</label>
<input type="text" id="calendrier"></p>
</body>
</html>
```

**Faire l'application 3
Sur feuille ou machine**

Applications

Application n°1 .Transformation d'une application écrite en JavaScript grâce au framework JQuery

L'application Web IMC écrite en JavaScript et dont le code est présenté en annexe 1 a pour objectif de calculer l'Indice de Masse Corporelle, d'une personne, en fonction de son poids et de sa taille.

Deux exemples d'exécution sont présentés ci-dessous :

Exemple n°1	
Indice de masse corporelle	
Taille <input type="text" value="1.65"/> mètres	
Poids <input type="text" value="53"/> kg	
<input type="button" value="Calculer"/>	
Après un clic sur le bouton Calculer	
Taille <input type="text" value="1.65"/> mètres	19.5 <i>Corpulence Normale</i>
Poids <input type="text" value="53"/> kg	
<input type="button" value="Calculer"/>	

Exemple n°2	
Indice de masse corporelle	
Taille <input type="text" value="1.70"/> mètres	
Poids <input type="text" value="43"/> kg	
<input type="button" value="Calculer"/>	
Après un clic sur le bouton Calculer	
Taille <input type="text" value="1.70"/> mètres	14.9 <i>Anorexie</i>
Poids <input type="text" value="43"/> kg	
<input type="button" value="Calculer"/>	

TRAVAIL A FAIRE	
1	Observer et comprendre le code du fichier indexJS.html se trouvant en annexe 1
2	Quelle balise faut-il intégrer dans le code pour utiliser le framework JQuery ? A quel endroit ?
3	Modifier la balise <code><input type="button" id="btnCalculer" onclick="afficheIMC(txtPoids.value, txtTaille.value);" value="Calculer"/></code> et fournir l'instruction JQUERY qui permettra d'appeler la fonction afficheIMC lors d'un clic sur le bouton btnCalculer en utilisant le framework JQuery..
4.	Transformer le code contenu entre les balises <code><script></script></code> pour que l'application fonctionne Remarque la partie script devra être réécrite sur votre copie.

Application n°2 Création d'un petit lexique informatique en utilisant la technologie Ajax

L'objectif est ici de créer un petit lexique informatique. Un clic sur un lien représentant une lettre affiche, trois définitions.

Exemple

Petit lexique informatique

A

B

C

D

E

F

G

Etc

Après un clic sur la lettre A,
l'affichage suivant se produit.

Petit lexique informatique

A **ANSII**

B Agence Nationale de la Sécurité des Systèmes
d'information

C **AUDIT D'ARCHITECTURE**

D Il s'intéresse au SI dans son ensemble et cherche à
E valider la pertinence des technologies et le respect des
bonnes pratiques de sécurité.

F **AZERTY**

G Nom du clavier français qui vient de l'ordre des six
premières lettres de la première rangée. Mot de passe à
Etc éviter

TRAVAIL A FAIRE

- | | |
|----|--|
| 1. | A partir de l'arborescence présentée à la figure 1, du code html du fichier lexiqueAjax.html fourni en annexe 2, et sachant que toutes les définitions des mots pour la lettre A se trouvent dans le fichier « lettre_a.html », compléter la méthode ajax ci-dessous |
|----|--|

	<pre>\$(document).ready(function() { \$('#lettre_a').click(function() { \$.ajax({ url: // à compléter success :// à compléter error:// à compléter }) }); });</pre>
2	Compléter le code pour que le lexique fonctionne lors d'un clic sur la lettre B

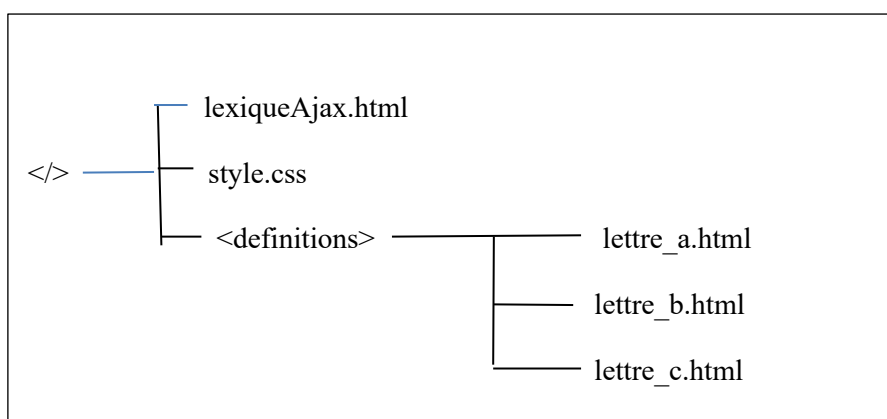


Figure1 : Arborescence du Site « Lexique »

Application n°3 Insertion d'un DatePicker dans un formulaire

TRAVAIL A FAIRE	
1.	Lancer le logiciel UWamp, copier dans le répertoire « www » le répertoire « Proj-Application3 » récupéré sur Elearn puis lancer votre IDE NetBeans en créant un projet PHP avec des sources existantes. Dans le répertoire « application3 » ouvrir le fichier « formReservation.html ».
2	Dans le fichier « formReservation.html », rajouter les deux balises permettant d'intégrer le framework JQuery et sa bibliothèque jquery-ui. Pour cela vous utiliserez un serveur CDN.
3.	<p>Dans l'entête du document HTML saisir le code ci-dessous, puis recharger le document HTML pour vérifier que la boîte de dialogue contenant le message « Je suis bien appelée » apparaît bien.</p> <pre><script> \$(function(){ window.alert(« Je suis bien appelée »); }); </script></pre>
4.	<p>Remplacer à présent l'instruction <code>window.alert(« Je suis bien appelée »);</code> par la création d'un datepicker.</p> <pre>\$('#txtdate').datepicker({ firstDay: 1, dateFormat : 'yy-mm-dd', numberOfMonths: 1, minDate : 0, maxDate : 31 });</pre>

TRAVAIL A FAIRE

5.	<p>Recharger l'affichage de la page « formReservation.html ». et constater l'affichage du datepicker.</p> <p>Rechercher sur l'Internet la signification de chacune des lignes écrites à la question 4) puis commenter votre code en conséquence. Vous noterez également le but de l'option beforeShowDay qui sera utilisée plus tard.</p>
6	Ajouter, dans le composant frmreservation.html le lien vers le fichier Aristo.css. Remarquez le changement de design du datepicker.
7	<p>Enrichir votre datepicker comme indiqué ci-dessous :</p> <pre> \$('#txtdate').datepicker({ firstDay: 1, dateFormat: 'yy-mm-dd', numberOfMonths: 1, minDate: 0, maxDate: 31, beforeShowDay: creerCalendrierSoiree }); </pre>
8	<p>Rajouter la fonction creerCalendrierSoiree, juste avant l'instruction \$('#txtdate').datepicker.</p> <pre> function creerCalendrierSoiree(date) { var jour; var mois; var annee; var result; var reste; jour = date.getDate()+1; mois = date.getMonth() + 1; annee = date.getFullYear(); reste = parseInt(jour) % 2; if (jour % 2) { result = [false, 'gris', "jour pair pas de sélection possible"]; } else { result = [true, 'rose', "jour impair on peut cliquer"]; } return result; } </pre>
9	Rechercher sur Internet l'objectif des méthodes getDate(), getMonth() et getFullYear().
10	Modifier la fonction creerCalendrierSoiree pour que seuls les jours pairs soient cliquables et qu'ils apparaissent en jaune.

L'objet DatePicker créé n'est pas satisfaisant car il permet de réserver une date de soirée pour laquelle il n'y a pas forcément de soirées étoilées prévues officiellement. On souhaite corriger l'application afin de ne visualiser que les dates où l'internaute peut éventuellement réserver et cela jusqu'à la fin de l'année civile. Autre point à améliorer, il serait judicieux de permettre à l'internaute de visualiser le nombre de places encore disponibles pour chaque soirée.

Ce travail sera réalisé en deux temps :

- Temps n°1 : les dates ainsi que les places restantes pour la soirée seront dans un premier temps stockées de manière fictive dans un dictionnaire avant d'être affichées dans le DatePicker.
- Temps n°2 : une fois que le DatePicker sera fonctionnel avec le dictionnaire pré-rempli, il faudra récupérer les données non plus à partir du dictionnaire mais depuis la table Soirée de la base de données mySQL.

Tâche n°1 : Création du DatePicker à partir des données stockées dans un dictionnaire.

TRAVAIL A FAIRE	
11.	<p>Dans la partie javaScript:</p> <ul style="list-style-type: none"> à la suite de l'instruction <code>\$(function(\$){</code>, définir les variables globales <code>aujourd'hui</code> et <code>nbmois</code> comme ci-dessous: <code>var aujourd'hui = new Date();</code> <code>var nbmois = 12 - aujourd'hui.getMonth();</code> puis modifier les propriétés du datePicker comme indiqué ci-après : <code>\$('#txtdate').datepicker({</code> <code> dateFormat : 'yy-mm-dd',</code> <code> numberOfMonths: nbmois,</code> <code> minDate : 0,</code> <code> maxDate : nbmois * 31,</code> <code> beforeShowDay: creerCalendrierSoiree</code> <code>});</code>
12.	a) L'objectif est de transformer la fonction creerCalendrierSoiree afin que l'affichage soit celui défini en annexe 3.
13.	Créer dans le fichier styles.css, les trois classes : vert, orange et rouge avec leur couleur de fond associée.
14.	Ajouter immédiatement après la déclaration et l'initialisation de la variable globale <code>nbmois</code> , la variable nuits_dispos . Cette variable est un tableau qui va contenir toutes les dates des soirées ainsi que leur nombre de places disponibles.

	<p>Les deux instructions à ajouter sont :</p> <pre>var nuits_dispos;</pre> <pre>nuits_dispos={"2022-02-14": 16, "2022-02-20": 5, "2022-03-13": 18, "2022-03-09": 30};</pre> <p><u>Exemple d'utilisation du tableau</u> : pour afficher le nombre de places disponibles pour la soirée du 2022-02-20 il suffira d'écrire :</p> <pre>alert(nuits_dispos["2022-02-20"]);</pre>
15.	<p>Modifier la fonction creerCalendrierSoiree pour que seules les dates présentes dans la variable <code>nuits_dispos</code> puissent être sélectionnées dans le datepicker.</p> <p>De plus si le nombre de places restantes pour la nuit est strictement inférieur à 10, la date du datepicker doit apparaître en fond rouge. Si le nombre de places est compris entre 10 et 20 la couleur de fond doit être orange et si le nombre de places est strictement supérieur à 21, la couleur de fond doit être verte.</p> <p>Aide à la conception : pour réaliser ce travail vous aurez besoin :</p> <ul style="list-style-type: none"> • de la fonction <code>parseInt(lecara)</code> qui convertit en entier le caractère passé en paramètre • de la propriété <code>undefined</code> <p><u>Exemple d'utilisation de la propriété undefined:</u></p> <pre>var x, y; y=2; if (x === undefined) { // ces instructions seront exécutées } if (y !== undefined) { // ces instructions seront exécutées }</pre>

Tâche n°2 : Récupération des données provenant de la base pour alimenter la variable globale `nuits_dispos`

16.	Importer la base <code>bdetoileL3.sql</code> qui se trouve sur le serveur Elearn.
17.	Rechercher sur Internet en quoi consiste les technologies Ajax et JSON qui vont être utilisées dans cette troisième tâche.
18.	Créer le fichier php " obtenirDates.php ". Ce composant doit valoriser un tableau associatif nommé resultat .

	<p><u>Rappel de cours</u> : Dans un tableau associatif, chaque élément est constitué de deux informations : la clé qui permet de repérer la valeur et la valeur elle-même.</p> <p>La clé qui permet de repérer la valeur doit être la date de la soirée à venir, et la valeur le nombre de places encore disponibles.</p>
9.	<p>Rajouter en dernière ligne du programme "obtenirDates.php", l'instruction :</p> <pre>echo json_encode(\$resultats);</pre> <p>Exécuter le programme " obtenirDates.php" et constater le bon affichage de la chaine de caractères :</p> <pre>{"2022-02-14": 6, "2022-02-20": 5, "2022-03-13": 28, "2022-05-09": 30};</pre>
10.	<p>Juste avant l'instruction <code>\$.datepicker.setDefaults(\$.datepicker.regional["fr"]);</code></p> <p>Insérer le code ci-dessous :</p> <pre>var req = \$.ajax({ url: "./obtenirDates.php", }); req.done(function(msg) { alert(msg); nuits_dispos = JSON.parse(msg);</pre>
11.	<p>Après l'extrait de code ci-dessous :</p> <pre>beforeShowDay: creerCalendrier });</pre> <p>Insérer les instructions suivantes:</p> <pre>}); req.fail(function() { alert("Erreur lors du traitement de la requête Ajax"); });</pre>
12.	<p>Mettre en commentaire l'instruction</p>

	<pre>//nuits_dispos = {"2022-02-14": 16, "2022-02-20": 5, "2022-03-13": 18, "2022-03-09": 30};</pre> <pre>//et ajouter la ligne</pre> <pre>nuits_dispos = "";</pre>
13.	Exécuter le fichier index.html et constater le bon affichage du datepicker comme présenté en annexe 3.

Annexe 1 imcJS.html

```
<html>
<head>
  <title>Calcul de L'IMC</title>
  <meta charset="UTF-8">
  <style type="text/css">
    .normale{
      background-color:greenyellow;
      font-size:40px;
    }
    .autre{
      background-color:red;
      font-size:40px;
    }
    .italique{
      font-style:italic;
    }
    .maigreur{
      background-color:#ff9933;
      font-size:40px;
    }
    #BlocSaisie{
      width: 50%;
      float:left;
    }

    #BlocRésultat{
      width: 50%;
      float:left;
    }

  </style>
  <script>
  function calculeIMC(poids, taille) {
    var imc = poids / (taille * taille);
    return imc;
  }
  function afficheIMC(poids, taille) {
    var imc = calculeIMC(poids, taille);
    var leRes = document.getElementById("indice");
    var lExplication = document.getElementById("explication");
    // méthode toFixed(n) renvoie n chiffres après la virgule
    leRes.innerHTML = imc.toFixed(1);
    if (imc < 16.5) {
```

```

        leRes.setAttribute("class", "autre");
        lExplication.innerHTML = "Anorexie";
    } else {
        if (imc < 18.5) {
            leRes.setAttribute("class", "maigreur");
            lExplication.innerHTML = "Maigreur";
        } else {
            if (imc < 25) {
                leRes.setAttribute("class", "normale");
                lExplication.innerHTML = "Corpulence Normale";
            } else {
                leRes.setAttribute("class", "autre");
                lExplication.innerHTML = "Obésité";
            }
        }
    }
}
}
}
</script>
</head>
<body>
    <fieldset>
    <legend>
        Indice de masse corporelle
    </legend>
    <div id="BlocSaisie">
        <p>Taille <input type="text" id="txtTaille" /> mètres</p>
        <p>Poids <input type="text" id="txtPoids" /> kg</p>
        <input type="button" id="btnCalculer"
        onclick="afficheIMC(txtPoids.value, txtTaille.value);" value="Calculer"/>
    </div>
    <div id="BlocRésultat">
        <p> <span id="indice"> </span> </p>
        <p> <span id="explication" class="italique"> </span></p>
    </div>
    </fieldset>
</body>
</html>

```

ANNEXE 2 : Code HTML du fichier LexiqueAjax.html

```
<!doctype html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>jQuery Lexique HTML</title>
<link href="style.css" rel="stylesheet" type="text/css"/>
  <script src="http://code.jquery.com/jquery-3.1.0.js"></script>
<script>
// A compléter

$(document).ready(function() {
  $('#lettre_a').click(function() {
    $.ajax({
      url:"????????????????????????????????", // à compléter
      success:                                , // à compléter
      error:                                  //à compléter
    })
  });

});

function actualiserPage(reponse){
$('#lexique').html(reponse);
}

</script>
</head>
<body>
<div id="container">
  <div id="header">
    <h2>Petit lexique informatique</h2>
  </div>
  <div id="alphabet">
    <div >
      <h3><a id="lettre_a" href="#">A</a></h3>
    </div>
    <div id="lettre_b">
      <h3><a href="#">B</a></h3>
    </div>
    <div>
      <h3>C</h3>
    </div>
    <div>
```

```

    <h3>D</h3>
  </div>
  <div>
    <h3>Etc</h3>
  </div>
</div>
<div id="lexique"></div>
</div>
</body>
</html>

```

Annexe 3

Les soirées étoilées au pic du midi de Bigorre
Les soirées étoilées au pic du midi de Bigorre
Les soirées étoilées au pic du midi de Bigorre

Les soirées étoilées au pic du midi de Bigorre
Les soirées étoilées au pic du midi de Bigorre

LES SOIREES AU PIC DU MIDI DE BIGORRE

Effectuer une réservation

Effectuer une réservation:

Date Soirée

janvier 2022							fevrier 2022							mars 2022						
L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D
					1	2							1	2	3	4	5	6		
3	4	5	6	7	8	9	7	8	9	10	11	12	13	7	8	9	10	11	12	13
10	11	12	13	14	15	16	14	15	16	17	18	19	20	14	15	16	17	18	19	20
17	18	19	20	21	22	23	21	22	23	24	25	26	27	21	22	23	24	25	26	27
24	25	26	27	28	29	30	28	29	30	31				28	29	30	31			
31																				