

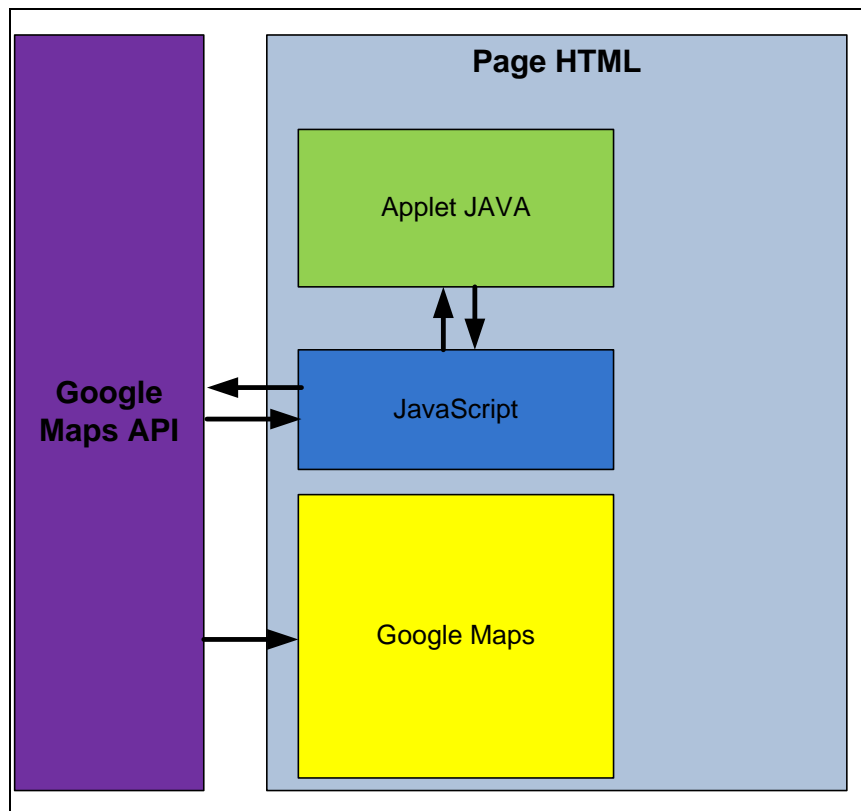
Utilisation de l'API Google Map (2 séances)

Raksmey PHAN

Ce TP a pour objectif de vous faire découvrir l'utilisation d'une Application Programming Interface (API) du web. A travers l'exemple de Google Map API nous verrons :

- La programmation du JavaScript
- L'utilisation de l'API Google Map à travers requêtes distantes faites en JavaScript
- L'interfaçage de Java comme surcouche à JavaScript

Le schéma suivant montre le flux d'informations entre les différentes composantes de la page HTML que nous allons créer.



I - Introduction à l'HTML et au JavaScript (1h30)

Nous allons créer une page HTML et utiliser quelques balises courantes en JavaScript.

1 - Présentation du code HTML (10min)

Le langage Hypertext Markup Language (HTML) est un langage de balises, conçu pour représenter les pages web. C'est-à-dire qu'elle utilise des <balise> </balise> pour organiser le contenu d'une page internet. Ci-dessous une l'organisation d'une page HTML :

```
<html>
  <head>
    // L'entête qui sert par exemple à déclarer certaine fonction utiliser dans la page.
  </head>

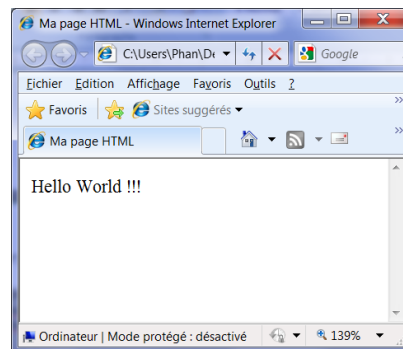
  <body>
    // Le corps du text où on met le contenu de la page
  </body>
</html>
```

Tester par exemple le code suivant :

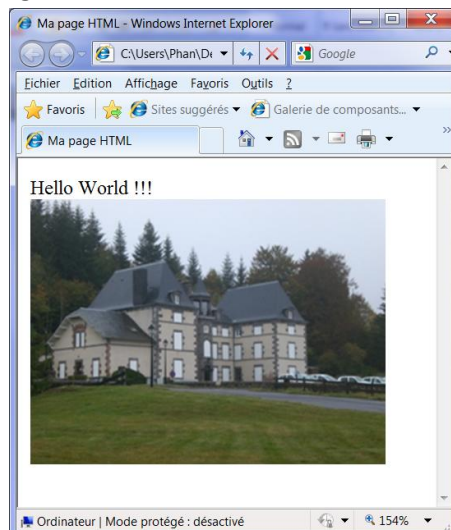
```
<html>
  <head>
  </head>

  <body>
    Hello World !!!
  </body>
</html>
```

- 1) A l'aide de cette page (<http://www.codeshttp.com/baliseh.htm>) qui regroupe les balises html, faire une page *Hello World* qui aura pour titre « Ma page HTML » avec comme message « Hello World !!! ». Le résultat doit ressembler à ceci :



- 2) Insérer une image dans la page. Le résultat final doit ressembler à ceci :



2 - Introduction à JavaScript (20min)

« JavaScript est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur¹. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de générer leurs propriétés, et notamment une propriété de prototypage qui permet d'en générer des objets héritiers personnalisés. » Wikipedia.

L'intégration de code JavaScript dans une page web se fait de cette manière.

```
<html>
  <head>
    <script type="text/javascript">
      function FonctionJavaScript()
      {
        // code ...
      }
    </script>
  </head>

  <body></body>
</html>
```

- 1) Le code Java permet d'ajouter des fonctions à du code HTML. Par exemple on peut ajouter des zones de texte et des boutons.

Utiliser le tutoriel de la page http://www.deluxeavenue.com/tutoriels_js03.php#binput pour dessiner la page ci-dessous.

Afficher Paris

Puis modifier la page html pour que le clic sur le bouton fasse afficher « Paris » dans la zone de texte.

III – Création d'un formulaire (1h) (à rendre)

Ainsi on peut aisément créer un formulaire sur une page web. Créer le formulaire suivant :

Afficher Paris

Choix 1 : ☒ - Choix 2 : ☐ - Choix 3 : ☐

Choix 1 ok

Afficher

Choix 1 : ☒ - Choix 2 : ☒ - Choix 3 : ☐

Choix 1 ok Choix 2 ok

Afficher

Choix 1 : ☒ - Choix 2 : ☒ - Choix 3 : ☒

Choix 1 ok Choix 2 ok Choix 3 ok

Afficher

Le bouton `Afficher_Paris` fait la même chose que dans l'exercice précédent.

Les boutons `Afficher` affichent les choix de l'utilisateur dans une zone texte (assez grande pour tout afficher).

II – Utilisation de l'API Google Map (1h30)

Nous allons insérer une carte Google Map dans une page web et la commander avec le langage JavaScript puis avec une application Java.

1 – API Google Map en Java Script (45min)

- 1) Obtenir une clé pour utiliser Google Map API
- 2) Lire la présentation de Google Map API
- 3) Copier le code suivant dans un fichier *.html et le lancer dans un navigateur.

```
<html>
  <head>
    <title>Test Google Geocoder</title>

    <script type="text/javascript"
src="http://www.google.com/jsapi?key=ABQIAAAA8TST_2HKKoGznxDawi66qhQmYiKzZDYKg5FRQqPLSc0VqIw2V
RQ5knpR5iws1A-O-pvtYsznA_QIGQ">
    </script>

    <script type="text/javascript">

      google.load("maps", "2.x") ;
      var map ;

      function Afficher_Paris()
      {
        text_AfficherParis.value = 'Paris' ;
      }

      function mapsLoaded()
      {
        map = new GMap2(document.getElementById("map"));
        map.enableGoogleBar();

        var Point = new GLatLng(45.777461, 3.082008);
        map.setCenter(Point, 5);

        var MapControl = new GMapTypeControl();
        map.addControl(MapControl);

        var LargeMapControl = new GLargeMapControl();
        map.addControl(LargeMapControl);

        // Zoom a la souris autorise
        map.enableScrollWheelZoom();
      }

      function loadMaps()
      {
        google.load("maps", "2", {"callback":mapsLoaded});
      }

      function start()
      {
        google.setOnLoadCallback(loadMaps);
      }

    </script>
  </head>

  <body>
    <script type="text/javascript">
      <!-- Démarrage de google map-->
      start() ;
    </script>

    <div id="secondaire" align="left">
      <input id='text_AfficherParis' value=''>
      <input id='button_AfficherParis' type='button' value='Afficher Paris'
onclick='Afficher_Paris();'>

      <!-- Carte google-->
```

```

<div id="map" style="width: 590px; height: 400px;"></div>
</div>
</body>
</html>

```

- 4) Expliquer ce que signifie chacune des lignes suivantes du fichier en-dessous.

```

<script type="text/javascript"
src="http://www.google.com/jsapi?key=ABQIAAAA8TST_2HKKoGznxDaWi66qhQmYiKzZDYKg5FRQq
PLSc0VqIw2VRQ5knpR5iws1A-O-pvtYsznA_QIQQ">
</script>
google.load("maps", "2.x") ;

```

```
map = new GMap2(document.getElementById("map")) ;
```

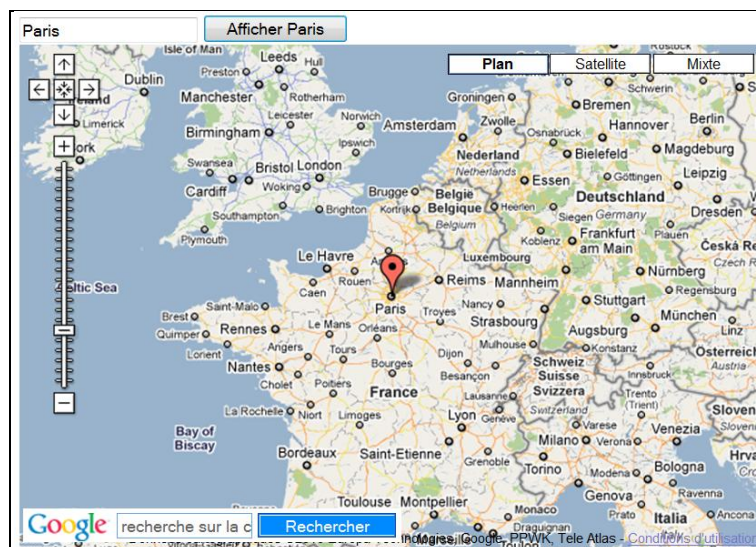
```
var Point = new GLatLng(46.98025,3.66943) ;
map.setCenter(Point, 5) ;
```

```
var MapControl = new GMapTypeControl() ;
map.addControl(MapControl) ;
```

```
var LargeMapControl = new GLargeMapControl() ;
map.addControl(LargeMapControl) ;
```

```
map.enableScrollWheelZoom() ;
```

- 5) Modifier le bouton pour que le clique centre sur Paris et affiche le pointeur.



II - API Google Map avec Java et JavaScript (15min)

- 1) Télécharger le projet vide d'Applet Java à cette adresse : fc.isima.fr/~phan/tp/web_service/tp2/JavaJSGoogleAPI.zip.
- 2) Ajouter un bouton qui aura la même fonction que celui en JavaScript. Voici le contenu du bouton :

```

Object[] args = {};
JSObject win = JSObject.getWindow(this) ;
win.call("Afficher_Paris", args) ;

```

- 3) Expliquer chaque ligne du code ci-dessus.

- 4) Pour qu'une application java puisse être lancée sur une page web, il faut le « certifier ». Cela implique que l'on doit créer un certificat avec différentes informations.

On compile avec NetBeans et on certifie le .jar créer donne la certif avec keylogger.exe.

Certifier un applet java :

- Il faut d'abord créer le profil du certificat, par exemple :
keytool -genkey -alias whatevercle -validity 365
- Puis il faut l'appliquer au .jar que l'on veut certifier, par exemple :
jarsigner Archive.jar whatevercle

A la place de Archive.jar, mettre le chemin exacte de votre .jar. Le keytool et jarsigner sont deux exécutables qui se trouve dans le dossier "java/bin/" selon la version, le dossier peut être "jdk1.6.0_19/bin/..."

III – Recherche d'itinéraire avec Google Map API (30min) (à rendre)

Faire une page html avec du JavaScript et une carte de google map qui permet de :

- Faire la recherche d'une ville (hint : GClientGeocoder::getLatLng)
- Trouver l'itinéraire entre deux villes