



zenika  
ARCHITECTURE INFORMATIQUE

# HTML 5 En pratique

## Exercices

# TP1 : Structure et sémantique

Dans ce premier exercice, nous allons tout d'abord mettre en forme un contenu existant (une liste d'articles) en tirant parti des nouvelles balises sémantiques de HTML5, puis nous développerons un formulaire d'inscription moderne et robuste.

Le style graphique n'est pour le moment pas important, il fera l'objet du prochain exercice.

## 1. Mise en forme d'une liste d'articles

Le fichier `news.txt` contient une liste d'articles que nous souhaitons structurer en vue de leur présentation sur Internet.

Pour cela, développez une page `index.html` contenant

- un en-tête,
- une barre de navigation,
- une section centrale contenant les articles,
- un pied de page affichant le logo Zenika
- un menu flottant proposant des liens "Actualités" et "Localiser l'événement".

Chaque article est composé de

- un titre,
- des méta-données comme l'auteur et la date de publication,
- d'un contenu rédactionnel déjà formaté

Prêtez une attention particulière au doctype et à l'encodage de la page.

## 2. Conception d'un formulaire

Le visiteur de notre site doit pouvoir s'inscrire pour participer aux événements.

Dans la page `inscription.html`, développez un formulaire d'inscription contenant les champs suivants :

- nom (requis, type texte )
- prénom (requis, type texte )
- date de naissance (type date )
- société (type texte)
- email (requis, type email )
- email2 (requis, type email , et doit correspondre au premier email)

- téléphone (type tel )
- nombre de jours (requis, nombre entre 1 et 2 inclus)
- nombre de places (requis, nombre entre 1 et 5)

N'oubliez pas d'ajouter les propriétés nécessaires pour la validation des champs, ainsi que pour améliorer l'expérience utilisateur (placeholders, messages...)

Ajoutez également un composant de type output (type text), qui permettra d'afficher le prix total de la réservation.

# TP2 : CSS

Dans le précédent exercice, nous avons mis en place la structure des pages d'accueil et d'inscription. Il est maintenant temps de les rendre présentables, en leur appliquant un style graphique sympathique.

Les captures d'écran fournies avec le TP vous serviront de guide.

## 1. Quelques pistes

Pour la page `index.html`

- La barre de navigation est une liste de liens ; chaque item est flotté à gauche.
- Article : Définir une marge
- Section : flotte à gauche
- Menu latéral : flotte à droite
- Figure : flotte à gauche et a une bordure
- body : Largeur 90% avec un légère ombre
- Header : Intègre un dégradé
- Logo dans Header : Animation de l'effet lumineux (une "ombre" blanche)

Pour la page `inscription.html`

- Le formulaire est centré
- Légère ombre sur les input
- Animation du bouton de soumission du formulaire

# TP3 : Géolocalisation et Drag'n'Drop

## 1. Géolocalisation

Maintenant que l'utilisateur peut s'inscrire à la conférence, nous allons l'aider à s'y rendre en lui proposant une carte et un itinéraire (voir les captures d'écran).

Actuellement, notre page d'accueil `index.html` n'affiche que les actualités. Les liens du menu flottant (Actualités / Localiser l'événement) vont désormais nous permettre d'afficher au choix les actualités ou la carte Google Maps.

- Créez un div d'id "`mapCanvas`" après la section contenant les actualités
- Développez une fonction Javascript pour basculer l'affichage entre le mode "actualités" et le mode "carte".
- Appelez cette fonction depuis les liens du menu flottant.

Affichons maintenant la carte.

Dans un premier temps, nous nous contenterons d'une simple carte statique, sans contrôles interactifs, centrée sur notre localisation géographique.

- Lors du premier affichage de la carte, utilisez l'API de géolocalisation pour déterminer votre position
- Affichez la carte centrée sur cette position, en renseignant la propriété `src` de la balise `<img>` affichant la carte.
- Pensez à gérer les différents cas d'erreur

Nous allons maintenant remplacer la carte statique par une carte dynamique indiquant non seulement l'emplacement de la conférence, mais également l'itinéraire conseillé pour s'y rendre à partir de notre position.

Il nous faudra pour cela utiliser l'API Google Maps ; le script `geoloc.js` contient déjà un squelette de code facilitant sa prise en main.

- Remplacez la carte statique par une carte dynamique en appelant la méthode `initialize()` lors du premier affichage de la carte
- Ajoutez un bouton au-dessus de la carte, intitulé "Se rendre à la conférence". Appeler la méthode `requestDirection()` lors du clic.
- Complétez la méthode `startGeolocation()`, pour initier une géolocalisation. Les méthodes `findPosition()` et `handleError()`, à compléter également, servent de *callback* respectivement en cas de succès et d'erreur de la géolocalisation.

## 2. Drag & Drop

La conférence propose un certain nombre de sujets, parmi lesquels l'utilisateur doit choisir. Pour cela, nous allons lui proposer une interface de type drag&drop, très intuitive.

Nous allons créer 2 pages :

- `agenda.html`, qui récapitulera les sessions sélectionnées.
- `agenda-edit.html`, qui permettra de choisir les sessions

### Page `agenda.html`

- Mettre en forme la page avec la même structure que la page d'accueil
- Ajouter un bouton permettant d'accéder à la page `agenda-edit.html`
- Sur toutes les pages, configurer le lien "Agenda" de la barre de navigation afin qu'il pointe sur cette page.

### Page `agenda-edit.html`

Cette page se présente sous la forme de deux listes (conférences disponibles, conférences sélectionnées). L'utilisateur peut glisser les conférences d'une liste à l'autre pour établir son programme.

Le code permettant de charger la liste initiale des conférences est fourni.

- Mettre en forme la page avec la même structure que la page d'accueil
- Ajuster les styles CSS pour que les deux listes soient présentées l'une à côté de l'autre (gauche/droite)
- Mettre en œuvre les API Drag&Drop pour autoriser le déplacement des conférences entre les deux listes.

Astuces :

- N'oubliez pas de gérer les événements `ondragenter` et `ondragover` !
- Passez l'ID de la conférence en méta-donnée du drag&drop, afin de pouvoir rattacher la puce correspondante à la liste de destination
- Ecouter les événements liés au Drag&Drop afin de fournir un *feedback* visuel à l'utilisateur (modification des styles graphiques)

# TP4 : Local Storage et mode offline

Maintenant que l'utilisateur a choisi les conférences auxquelles il souhaite assister, nous allons pouvoir afficher le récapitulatif dans la page `agenda.html`.

L'utilisateur apprécierait de retrouver sa sélection en revenant sur l'application, et aimerait pouvoir consulter les informations des conférences hors ligne. Pour cela, nous ferons appel aux API de stockage du navigateur, et au cache applicatif.

## 1. LocalStorage

La liste des conférences devant persister entre deux sessions de navigation, nous allons utiliser l'API `LocalStorage`.

La clé de stockage d'une conférence sera son ID, et la valeur associée à cette clé sera sa catégorie (cette information nous servira lors du prochain exercice).

Au niveau de la page `agenda-edit.html`

- Complétez les fonctions gérant le drag&drop afin de sauvegarder la liste des conférences sélectionnées (ajout/suppression en fonction de la liste cible)
- Au chargement de la page, appelez la méthode `loadData()` (à compléter) afin de peupler correctement les deux listes (lors de la boucle sur les conférences, il suffit de vérifier si leur ID fait partie des IDs sauvegardés)

Dans la page `agenda.html`

- Au chargement de la page, appelez la méthode `loadDataInAgenda()` (à compléter) afin d'afficher la liste des conférences sélectionnées (même technique que précédemment).

## 2. Cache de ressources

Mettre les pages `agenda.html`, `agenda-edit.html` ainsi que les scripts et ressources nécessaires en cache.

## 3. Bonus

Ajouter la possibilité d'enregistrer les valeurs du formulaire de la page inscription. (Ajouter un bouton `submit` qui ne propose pas de validation de contenu)

# TP5 : Multimédia et Canvas

Pour terminer, nous allons ajouter une dimension multimédia à notre application. Tout d'abord en intégrant une vidéo au sein d'une page (sans plugin), puis en tirant parti du Canvas pour visualiser la popularité des différentes catégories de conférences.

## 1. Intégration d'une vidéo

Dans la page `index.html`, intégrez une vidéo dans un article.  
Le fichier vidéo est fourni dans le répertoire `ressources/vidéo`.

## 2. Manipulation du Canvas

Dans la page `agenda.html`, nous allons créer un graphique à barres pour représenter la répartition des conférences choisies par catégorie.  
Par exemple, la catégorie web peut représenter 40% des conférences sélectionnées.

Dans le fichier `loadData.js`, la variable `liste` contient le nombre de conférences sélectionnées par l'utilisateur, par catégorie. Une simple règle de trois suffira donc à calculer la popularité relative de chaque catégorie.  
Utilisez l'API Canvas pour tracer un graphique en barres à partir de ces valeurs.

Bonus : animer le graphique lors du chargement de la page.