

## TP1 : Initiation à JavaScript

Le but de ce TP est de réaliser un petit site web en utilisant différentes techniques Web, mais en se concentrant particulièrement sur les interactivités des pages web par l'utilisation des scripts : JavaScript.

Ce TP sera divisé en plusieurs pages, chaque page présente un exercice parmi ceux cité ci-dessous (page authentification, page accueil, des pages qui présentent le cours JavaScript sous forme de tutoriaux).

Toutes les pages sont présentées par la même **maquette**, sauf la première page qui présente une fenêtre de connexion (sert d'entrée à notre site).

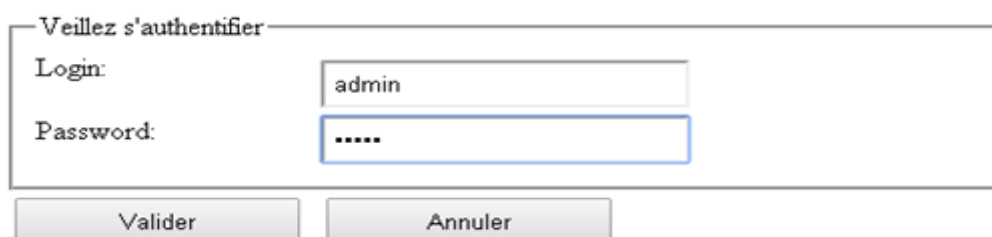
**Astuce** : Vous pouvez faire une maquette générale, et après vous faites copier/coller au niveau de chaque page, en ajoutant après les spécificités correspondantes à chacune.

Pour la réalisation de notre site, nous aurons besoin à une feuille de style « style.css », certaines images, et des pages en HTML 5. Le projet aura l'arborescence finale suivante :



### La page d'authentification

La page à réaliser nommée : « auth.html », et elle aura l'allure suivante :

L'image présente une maquette d'une page d'authentification. Elle est encadrée par un rectangle gris. En haut à gauche, le titre « Veuillez s'authentifier » est écrit. À l'intérieur, il y a deux champs de saisie. Le premier est précédé de l'étiquette « Login: » et contient le texte « admin ». Le second est précédé de l'étiquette « Password: » et contient cinq points « ..... ». En bas de la page, il y a deux boutons rectangulaires : « Valider » à gauche et « Annuler » à droite.

1. Écrire le code html correspondant, qui permettra d'afficher cette page.
2. Ajouter les styles suivants à la page de style « style.css »

Balise <form>	Nom=auth, marge extérieur=30px et marge intérieur=30px.
Balise <fieldset>	Largeur = 500px , hauteur = 100 px , Flottement à gauche
Balise <label>	Largeur : 150px , affichage : en block, Flottement à gauche
Balise <input>	Affichage en block, largeur : 180px, marge extérieur=5px et marge intérieur : 5px

3. Ecrire le code javascript nécessaire pour vérifier à la validation si login=admin et pwd=admin, il passe à la page index.html, sinon il affiche un message en rouge au niveau du document avant le formulaire.

Le code nécessaire pour le passage à une autre page :

```
Document.location.href="index.html"
```

### Page index.html

La page à réaliser nommée : « **index.html** », qui correspond à la page d'accueil, et elle aura l'allure finale suivante :



1. Commencez par la préparation de la maquette suivante en se basant sur les styles définis ci-après (à mettre au niveau du fichier style.css).

La maquette comprend essentiellement 4 zones (header, nav, section et footer). L'entête « header » va contenir une division pour le logo de l'ISET, avec un titre de « Bienvenue ». La partie « nav » va contenir un Menu déroulant qui présente les thèmes généraux à traiter. La partie « section » qui va contenir les « articles » nécessaires pour chaque page. La partie « footer » qui sert à spécifier « copyright ».



Pour l'allure générale de la maquette, c'est à vous à mettre en évidence les styles en se basant sur vos connaissances en HTML5 et CSS.

(Proposition pour les couleurs : body : #9DBDBD, header, nav et footer:#FF8306, et pour section : #FF9B37) et l'image du logo présente un lien vers la page « index.html ».

Pour le Menu déroulant, les styles sont détaillés dans le tableau suivant (c'est à vous d'en déduire la structure du menu) :

<b>#menu-accordeon</b> : identifiant à affecter à la balise <code>&lt;ul&gt;</code> qui va contenir notre Menu.	<code>padding:0; margin:0; list-style:none; text-align: center; width:280px;</code>
<b>#menu-accordeon ul</b>	<code>padding:0; margin:0; list-style:none; text-align: center;</code>
<b>#menu-accordeon li</b>	<code>background-color:black; border-radius: 6px; margin-bottom:2px; box-shadow: 3px 3px 3px #999; border:solid 1px #333A40</code>
<b>#menu-accordeon li li</b>	<code>max-height:0; overflow: hidden; border-radius:0; background: #444; box-shadow: none; border:none; margin:0</code>
<b>#menu-accordeon a</b>	<code>display:block; text-decoration:none; color: #fff; padding:8px; font-family: verdana; font-size:20px;</code>
<b>#menu-accordeon ul li a, #menu-accordeon li: hover li a</b>	<code>font-size:1em;</code>
<b>#menu-accordeon li: hover</b>	<code>background: silver;</code>
<b>#menu-accordeon li li: hover</b>	<code>background: #999;</code>
<b>#menu-accordeon ul li: last-child</b>	<code>border-radius: 0 0 6px 6px; border:none;</code>
<b>#menu-accordeon li: hover li</b>	<code>max-height: 15em;</code>

Les astuces utilisées à ce niveau pour rendre le Menu déroulant, sont ::

#menu-accordeon li li → max-height:0; overflow: hidden;

Et si on a survol de souris sur le menu principal on aura :

#menu-accordeon li: hover li → max-height: 15em;

### Les Menus Principaux

Pour les Menus principaux : la partie « **section** » va contenir une description détaillée du thème spécifié en se basant sur les définitions citées au niveau de notre cours, et respectant les mise en page nécessaires (listes, tableaux, lien hypertexte, ...)

- **Math** → lien vers la page « math.html », qui détaille la bibliothèque Math.
- **String** → lien vers la page « string.html », qui détaille la bibliothèque String.
- **Date** → lien vers la page « date.html », qui détaille la bibliothèque Date.
- **Etude de cas** → lien vers la page « etude.html », qui détaille un peu l'exercice de l'étude de cas à traiter.

### Les Menus Secondaires

Pour les Menus secondaires, chacun d'eux présente un lien vers une page de l'exercice spécifié.

**Exemple** : Ex1 → lien vers la page « Ex1.html ».

Vous trouvez ci-après la liste d'exemples d'exercices à traiter, vous essayez de les ranger au niveau des menus selon leur thème traiter (Math, String, Date, Etude de cas).

### Exemple d'Exercices à implémenter

**Exercice 1** : Transformer une durée exprimée en secondes en heures, minutes, secondes.

**Exercice 2** : Effectuer les 5 opérations mathématique (+, -, \*, /, %) sur 2 nombres entrés au clavier.

**Exercice 3** : Calculer la tangente de A en n'utilisant que les fonctions sin & cos.

**Exercice 4** : Calculer l'hypoténuse d'un triangle de côtés A et B.

**Exercice 5** : Calculer la moyenne de 3 nombres fournis en entrées.

**Exercice 6** : Ecrire un programme qui échange les valeurs de deux variables.

**Exercice 7** : Ecrire un programme qui transforme un nombre entier de jours en données en nombres de jours, de mois, d'années (1 mois = 30 jours).

**Exercice 8** : Ecrire un programme qui détermine la somme de trois mesures de temps données en heures, minutes et secondes et qui donne le résultat en jours, heures, minutes et secondes.

**Exercice 9** : On donne une somme exprimée en Euro. On demande de déterminer le nombre de billets de 1000 et de 100 Euro et le nombre de pièces 50, 20, 5 et 1 Euro dont il faut disposer pour reconstituer cette somme.

**Exercice 10** : On donne deux nombres x et y. On demande d'écrire un programme qui range le plus petit dans x et le plus grand dans y.

**Exercice 11** : Ecrire le code permettant de saisir le prénom et la classe d'un étudiant. Si le prénom et la classe sont différents de chaîne vide, alors afficher le message :

Bonjour prénom votre classe est classe !

**Exercice 12** : Ecrire le code permettant de Saisir un prix hors taxe, et Calculer et afficher le prix hors taxe et le prix TTC (toute taxe comprise), sachant que :  $\text{PrixTTC} = \text{prixHT} * 1.2$

**Exercice 13** : Ecrire le code permettant de saisir l'âge d'une personne. Et d'afficher le message correspondant.

- 1- Si l'âge est inférieur ou égal à zéro, alors afficher un message d'erreur,
- 2- Si l'âge est inférieur ou égal à 13, alors afficher "vous êtes encore trop jeune",
- 3- Si l'âge est compris entre 13 et 18, alors afficher "Vous êtes encore mineur(e)",
- 4- Si l'âge est compris entre 18 et 25, alors afficher "Vous êtes déjà majeur(e),
- 5- Sinon afficher "Ne vous vieillissez donc pas !".

**Exercice 14** : Ecrire le code permettant de calculer et d'afficher la somme des entiers naturels compris entre 1 et 100.

**Exercice 15 :** Ecrire le code permettant de remplir un tableau Année contenant les 12 mois de l'année, et de demander à l'utilisateur de saisir un mois et de chercher sa position dans le tableau et à quel trimestre il appartient.

**Exercice 16 :** Ecrire le code permettant de Saisir deux chaînes de caractères, Les concaténer, Déterminer la longueur de la chaîne résultante, Accéder à un caractère dont la position sera saisie par l'utilisateur (prévoir un test dans le cas où la position n'est pas un entier ou qu'elle dépasse la longueur de la chaîne) , Extraire une partie de la chaîne.

**Exercice 17 :** Ecrire le code permettant d'afficher la date d'aujourd'hui et l'heure actuelle en détail.

**Exercice 18 :** On veut calculer le poids idéal d'une personne. Pour cela, on doit avoir trois informations : sa taille T en cm, son âge A en années, et son sexe S (qui sera donné par un champ texte dans lequel on écrira "homme" ou "femme"). La formule est :

- pour un homme :  $(3 * T - 250) * (A + 270) / 1200$
- pour une femme :  $(T / 2 - 30) * (180 + 1) / 200$

**Exercice 19 :** On a deux champs texte. Dans le premier on entre un texte et dans le second on entre un caractère. Écrire par 2 manières la fonction qui affiche vrai si le caractère est compris dans le texte et faux sinon. La première en utilisant `indexOf` et la 2ème en écrivant une boucle en utilisant la méthode `charAt()`.

**Exercice 20 :** Ecrire un script permettant de faire apparaître la table de multiplication pour les entiers compris entre 0 et 9 dans et présenter le résultat dans un tableau [11x 11] dans la page HTML.

**Exercice 21 :** On se propose de développer un formulaire HTML, composé d'une zone de texte, de trois objets liste déroulante et d'un bouton de type *submit*. La zone de texte sera créé avec du code HTML, alors que les trois listes seront générées avec du code JavaScript, qui contient respectivement les numéros des jours d'un mois (de 1 à 31), les mois de l'année (de Janvier à Décembre) et les années (de 1975 jusqu'à cette année).

## ETUDE DE CAS

Un restaurant vous demande de réaliser un formulaire pour qu'un client puisse commander son repas à la carte par internet. Vous calculerez au fur et à mesure le tarif qui sera affiché dans un champ du formulaire et remis à jour lorsqu'un plat sera sélectionné. Si le tarif dépasse une certaine somme, le client aura droit à un dessert gratuit, un message d'alerte le préviendra. Lorsque la commande est prête, un clic sur le bouton Commander postera la commande au restaurant.

### Étape 1 : dessin du formulaire

Le formulaire que vous devez créer comporte :

- ♣  une adresse de livraison,
- ♣ des listes de choix pour sélectionner une entrée, un plat, un dessert (prévoir dans chaque liste un item "aucun" car il n'est pas obligatoire de prendre une entrée, un plat et un dessert).

- ♣ un champ pour donner le nombre de personnes pour chaque liste de choix (par exemple, 2 personnes peuvent prendre une entrée et une seule un dessert).
- ♣ un champ désactivé pour afficher le tarif (l'utilisateur ne peut pas entrer de données dedans, par contre votre script JavaScript pourra afficher les données dans le champ).

N'oubliez pas de nommer les champs du formulaire afin de pouvoir y accéder dans vos scripts (attribut name dans INPUT). Un clic sur le bouton Commander provoquera l'appel de la fonction **VerifierCommande()** qui retournera **true** si l'adresse est valide et que le nombre de **personnes** est correct, **false** sinon. Dans le cas d'un retour false le mail ne sera pas envoyé.

## Étape 2 : Gestion du nombre de personnes

**Par défaut**, le nombre de personnes est à 0 lorsque l'item **aucun** est sélectionné dans la liste de choix. S'il y a 0 et qu'un plat est sélectionné il passe à 1, sinon le nombre de personnes reste inchangé.

## Étape 3 : calcul du prix du repas

Calculez le **prix du repas au fur et à mesure de la commande**. Lorsqu'un plat ou une entrée ou un dessert est sélectionné affichez le nouveau prix. Vous devez tester le nombre de personnes pour calculer le prix afin d'éviter de faire des multiplications avec des données non valides.

## Étape 4 : vérification des données

### Nombre de personnes

Il doit être compris entre 0 et une borne supérieure que vous fixerez.

### Adresse

Il faut vérifier si l'adresse est remplie et valide. Pour cela, créez une fonction **AdresseValide(chaine)** qui teste si la chaîne passée en paramètre est valide. La fonction tester:

- ♣ si le nombre de caractères de la chaîne est **non nul** ;
- ♣ si la chaîne contient un code postal suivi d'une ville

Cette fonction retourne false s'il y a un problème, true sinon. En cas de problème elle affiche un **message d'alerte**.

## Commande

Il est possible de ne pas prendre de dessert, mais on ne peut pas passer une commande si on ne prend ni entrée, ni plat, ni dessert.

## Étape 5 : envoyer le mail

Lorsque vous cliquez sur le bouton commander la commande est envoyée au restaurant, à condition que les champs soient remplis correctement. (Vous pouvez utiliser mailto)