

Séance n° 5	Évaluation finale – Groupe TP4
-------------	--------------------------------

Objectif :

Déposez sur UniversiTICE dans le dépôt correspondant à votre groupe trois fichiers contenant les codes sources résolvant les 3 exercices ci-dessous. Les fichiers devront être nommés par l'intitulé de votre groupe, votre nom et le numéro de l'exercice, cad TP4-Nom-1.html, TP4-Nom-2.html et TP4-Nom-3.html.

Les supports de cours accessibles directement sur le site du cours et la documentation sont autorisés.

Vous accorderez un soin particulier à la présentation de votre code : respectez les conventions de nommage des variables, l'indentation, commentez votre code quand nécessaire.

Assurez-vous de travailler avec un navigateur à jour.

1 Une histoire de moyenne (5 points)

Vous allez réaliser un script qui va calculer la moyenne de tous les champs de saisie par l'utilisateur et l'afficher sur la page.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Une histoire de moyenne</title>
  </head>

  <body>
    <label>Valeur 1 : </label>
    <input type="text"> <br>
    <label>Valeur 2 : </label>
    <input type="text"> <br>
    <label>Valeur 3 : </label>
    <input type="text"> <br>
    <label>Valeur 4 : </label>
    <input type="text"> <br>
    <label>Valeur 5 : </label>
    <input type="text"> <br>
    <label>Valeur 6 : </label>
    <input type="text"> <br>
    <label>Valeur 7 : </label>
    <input type="text"> <br>
    <label>Valeur 8 : </label>
    <input type="text"> <br>

    <button id="go">Calculer</button>

    <p>
      Moyenne = <span id="result"></span>
    </p>

    <script type="text/javascript">
    </script>
  </body>
</html>
```

Le document HTML est téléchargeable [ici](#) (Clic droit > télécharger la cible du lien sous).

1. Créez le fichier ci-dessus et enregistrez-le
2. Créez la fonction `inputsMean` qui va :
 - récupérer les champs de saisie de la page **1 pt**
 - calculer la somme de leurs valeurs **1 pt**
On attend des nombres, vous penserez à faire les vérifications nécessaires
 - afficher la moyenne des valeurs dans l'élément d'identifiant `result` **1 pt**
3. Ajoutez un écouteur d'évènements sur le bouton d'identifiant `go` qui appellera la fonction `inputsMean` lors d'un clic **1 pt**

Voici ce que vous pouvez obtenir :

Valeur 1 :

Valeur 2 :

Valeur 3 :

Valeur 4 :

Valeur 5 :

Valeur 6 :

Valeur 7 :

Valeur 8 :

Moyenne = 45

Présentation du code : **1 pt**

2 Bacheliers et étudiants (5 points)

Vous allez écrire des classes permettant de gérer des bacheliers et des étudiants. Un bachelier est caractérisé par son nom et sa mention au bac. Un étudiant est un bachelier qui prépare un diplôme.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Études supérieures</title>
  </head>

  <body>

    <p>Caractéristiques des étudiants :</p>

    <script type="text/javascript">
    </script>
  </body>
</html>
```

Le document HTML est téléchargeable [ici](#) (Clic droit > télécharger la cible du lien sous).

1. Créez le fichier ci-dessus et enregistrez-le.
2. Écrivez la classe `Bachelier` qui possède un constructeur prenant en paramètre un nom et une mention au bac et initialise les propriétés `nom` et `mention`. **1 pt**
3. Ajoutez une méthode `affiche` à votre classe `Bachelier` qui affiche une description dans le document HTML (Exemple : Bachelier X a obtenu la mention Y). **1 pt**

4. Un étudiant est un bachelier qui prépare un diplôme. Écrivez la classe `Etudiant` et son constructeur. **0.5 pt**
5. Ajoutez une méthode `affiche` à votre classe `Etudiant` qui affiche une description dans le document HTML. **0.5 pt**
6. Créez un objet de type `Bachelier` et un objet de type `Etudiant`. Affichez leur description. **1 pt**

Présentation du code : **1 pt**

3 TamDead (10 points)

Tama n'a pas bien supporté ses dernières aventures en TP... Vous allez écrire un nouveau jeu avec sa version zombie : `TamDead`. `TamDead` a besoin de cervelles fraîches et surtout d'une barrière pour éviter qu'il ne s'échappe. Si la barrière s'écroule ou que `TamDead` meurt de faim (façon de parler), il fuira. Et on est tous d'accord, on n'aimerait pas que ça arrive...



Le document HTML est téléchargeable [ici](#) (Clic droit > télécharger la cible du lien sous).

3.1 Évolution de `TamDead`

Dans la première partie, vous allez créer la classe `TamDead` qui gèrera la vie de `TamDead` et son évolution : son besoin en cervelles, l'état de la barrière qui l'empêche de s'échapper...

1. Créez le fichier ci-dessus et enregistrez-le.
2. Déclarez la classe `TamDead` et son constructeur qui ne prend aucun paramètre. **0.5 pt**
3. Initialisation des propriétés dans le constructeur : **1,5 pt**
 - (a) La propriété `hunger` servira à stocker l'élément HTML indiquant l'appétit de `TamDead`. La propriété `fencePower` servira à stocker l'élément HTML indiquant la puissance de la barrière vous protégeant de `TamDead`. La propriété `tamDeadImg` servira à stocker l'élément HTML image représentant `TamDead`. Stockez dans les propriétés `hunger`, `fencePower`, `tamDeadImg` les éléments HTML d'identifiant `hunger`, `fence`, `tamdead`.
 - (b) Donnez les conditions de départ de `TamDead` : le contenu textuel de `hunger` est de 0, celui de `fencePower` de 100% et la source de l'image est http://chloecabot.com/mmi/M3203/img/tamDead_alive.png.
 - (c) Initialisez la propriété `here` à `True`. Elle indiquera si `TamDead` s'est enfui ou non.
 - (d) Initialisez la propriété `intervalID` à 0.
4. Écrivez la méthode `life` qui gèrera l'évolution de `TamDead` lors d'une étape de sa vie : **1 pt**
 - (a) Son appétit gagne 5 points
 - (b) La puissance de la barrière perd 5 points
 - (c) Si l'appétit de `TamDead` est supérieur ou égal à 50, il change d'apparence (url : http://chloecabot.com/mmi/M3203/img/tamDead_angry.png)
 - (d) Sinon il a une apparence normale (url : http://chloecabot.com/mmi/M3203/img/tamDead_alive.png)

- (e) Si son appétit est de 100 ou la puissance de la barrière est nulle, TamDead s'enfuit (url : http://chloecabot.com/mmi/M3203/img/tamDead_gone.png)
5. Écrivez la méthode `start` qui lance la méthode `life` toutes les 500ms. **1 pt**
6. Ajoutez un appel à la méthode `start` dans le constructeur de TamDead. **0.5 pt**
7. Écrivez la méthode `stop` qui va : **1 pt**
- (a) Stopper l'intervalle.
 - (b) Donner la valeur `False` à la propriété `here`.
 - (c) Créer un nouvel élément HTML paragraphe qui sera le message de fin de partie. Le paragraphe possède la classe CSS `message` et un contenu textuel (choisissez ce que vous voulez).
 - (d) Ajouter ce paragraphe à son parent dans le document HTML : l'élément HTML d'identifiant `end`.
8. Ajoutez un appel à la méthode `stop` dans la méthode `life`. **0.5 pt**

3.2 Mise en place des écouteurs d'évènements

1. Dans le script principal, ajoutez un écouteur d'évènements au bouton d'identifiant `start` lors d'un clic. La fonction appelée devra effacer le contenu textuel de l'élément d'identifiant `end` et créer un nouvel objet de type TamDead. **1 pt**
2. Dans la classe TamDead, ajoutez une méthode `feed` qui diminue l'appétit de TamDead de 5 points seulement si TamDead ne s'est pas enfui. **0.5 pt**
3. Dans la classe TamDead, ajoutez une méthode `reEnforce` qui augmente la puissance de la barrière de 20 points seulement si TamDead ne s'est pas enfui. **0.5 pt**
4. Dans le constructeur de la classe TamDead, ajoutez des écouteurs d'évènements aux boutons d'identifiant `feed_button` et `reenforce_button` qui appelleront respectivement les méthodes `feed` et `reEnforce` lors d'un clic. **1 pt**

Présentation du code : **1 pt**