

TP noté (Web avancée LA1)

6 Juin 2017

Consignes générales

- La durée de l'épreuve est de 4h. Une archive, sous format zip contenant les fichiers sources, doit donc être envoyée à l'adresse `Yacine.bouzidi@inria.fr` (avec comme objet CTPWEB), avant 12h00¹. Dépassé ce délai, le travail est jugé irrecevable.
- Ce travail est à réaliser seul. En outre, il est strictement interdit d'utiliser du code trouvé sur internet sinon de manière très marginale (inférieur à 10%). Toute entorse à cette règle sera aisément repérable et sanctionnée par une note de zéro à l'épreuve.
- Le barème des exercices est fourni à titre indicatif.

1 Exercice 1 : Module QCM (7 points)

On souhaite mettre au point un module de questionnaire à choix multiples (QCM) que l'on pourra intégrer à nos applications web. Un tel module propose une liste de questions à l'utilisateur, chacune accompagnée d'une liste de propositions de réponse. Après avoir sélectionné une ou plusieurs réponses pour chaque question, l'utilisateur soumet le résultat et une note globale est calculée sur la base suivante : une réponse correcte vaut un point, une réponse incorrecte vaut zéro.

On suppose que les questions, récupérées depuis un serveur distant, sont données sous le format JSON (Javascript Object Notation) suivant :

```
{
  "question" : "Lesquels de ces nombres ne sont pas algébriques ?",
  "propositions" : [" $\pi$ ", " $\sqrt{2}$ ", "e"],
  "reponses" : [1,3]
}
```

où les champs *question*, *propositions* et *réponses*, contiennent respectivement la question, la liste des réponses proposées et les indices des réponses correctes.

De plus, l'ensemble des questions est donné sous la forme d'un objet JSON contenant un tableau de questions.

¹En cas d'incapacité d'envoyer le travail par email, prière de contacter l'enseignant qui viendra le récupérer par d'autres biais.

```

{
  "questions" :
  [
    {
      "question" : "ques1"
      "propositions" : "[...]"
      "reponses" : "[...]"
    },
    {
      "question" : "ques2"
      "propositions" : "[...]"
      "reponses" : "[...]"
    },
    :
  ]
}

```

- Écrivez une fonction qui convertit une question en format JSON en un objet Javascript *question* et une autre qui convertit une liste de questions en un tableau d'objets *question* (on pourra utiliser la fonction `JSON.parse()`).
- Écrivez une fonction qui parcourt le tableau des questions et affiche un questionnaire HTML contenant : les questions, les proposition de réponses, des boutons pour sélectionner les réponses à chaque question, ainsi qu'un bouton **valider** pour soumettre les réponses.
- Écrivez une fonction qui calcule la note globale de l'utilisateur après soumission du QCM.
- On souhaite mettre en place un barème de notation. Pour cela, on associe à chaque question un coefficient (entier compris entre 1 et 5) qui détermine son poids dans la note finale. Modifiez votre code afin de prendre en compte cette nouvelle notation.

2 Exercice 2 : calculatrice (7 points)

On souhaite mettre au point une petite calculatrice pour effectuer des opérations binaires sur des entiers. Pour cela, à l'aide des directives `ng-app` et `ng-Controller`, on créera une application Angular `Calc` ainsi qu'un contrôleur `CtrCalc` associé.

- Ajouter deux champs `input` permettant de saisir les deux opérandes.
- Ajouter quatre boutons pour effectuer les opérations de base sur les entiers saisis.
- Ajouter un troisième champs invisible qui sert à stocker le résultat d'un calcul en vue de le réutiliser ultérieurement. On utilisera pour cela la directive `ng-show`.
- Proposez une variante de cette calculatrice dans laquelle les deux opérandes sont saisies sur un même champs texte (à la manière d'une calculatrice standard).

3 Exercice 3 : ToDo liste (6 points)

L'objectif de cet exercice est de mettre au point une application de planification de tâches (**ToDo liste**). Le principe d'une **ToDo liste** est le suivant :

Dans une zone de saisie de texte, l'utilisateur peut ajouter l'intitulé d'une tâche à effectuer. Celle-ci est alors ajoutée à la liste des tâches représentée par un menu déroulant. A chaque tâche présente dans la liste est associée une case que l'utilisateur peut cocher pour indiquer que celle-ci a été réalisée. Dans ce cas, l'intitulé de la tâche est barré dans le menu déroulant.

- Proposez un code permettant de créer une **ToDo liste** respectant les spécifications indiquées plus haut.
- Rajoutez trois boutons **tout**, **fait** et **en cours** permettant à l'utilisateur de visionner respectivement : toutes les tâches, celles déjà effectuées ainsi que celles restantes.
- A présent, lors de la saisie de chaque tâche, sera également saisie une date limite de réalisation. Rajoutez un bouton qui permet de trier les tâches dans le menu déroulant en fonction de l'urgence de leurs dates de réalisation.