

## Examen HAI203I - Programmation événementielle et web

28/06/2022 - 16h-18h - TD 36.309 (154) TD 36.402 (118)

Partie Pierre Pompidor / JavaScript / Aucun document autorisé

### Question 1. (1 point)

Soit une liste *maliste* quelconque, donnez le code JavaScript qui affiche ses éléments un par un, du dernier au premier.

La liste de listes suivante est utilisée dans les questions 2 et 3 (il est inutile de la recopier dans vos codes) :

```
var peintres = [['Monet', 'Cézanne', 'Gauguin', 'Renoir'],
                ['Picasso', 'Dali', 'Vélasquez'],
                ['Léonard De Vinci', 'Michel-Ange', 'Raphaël']];
```

Les valeurs des listes contenues dans la liste *peintres* pourraient bien sûr être différentes...

### Question 2. Fonction de rappel (callback) (3 points)

Donnez l'implémentation de la fonction *peintresParIndice()* de telle sorte qu'appelée avec une liste de listes et un indice, elle affiche les valeurs de cet indice dans chacune des sous-listes (si la liste a un élément à cet indice).

Voici deux exemples de mises en œuvre :

```
peintresParIndice(peintres, 0, function(v) { console.log(v); });
```

produit cet affichage :

```
Monet
Picasso
Léonard de Vinci
```

```
peintresParIndice(peintres, 3, function(v) { console.log(v); });
```

produit cet affichage :

```
Renoir
```

### Question 3. Serveur Node.js (3 points)

Complétez le code du service web suivant :

```
app.get('/peintresdememenationalité/:peintre', function(request, response) {
  // EMLACEMENT DU CODE A ECRIRE
});
```

Lors de son appel avec une route comportant un nom de peintre, le service web renvoie la liste sérialisée le contenant (il n'y a pas d'homonymes et si le nom est inconnu, une liste vide est renvoyée).

Par ex., `localhost:8888/peintresdememenationalité/Dali` renvoie `"['Picasso', 'Dali', 'Vélasquez']"`

Vous pouvez utiliser la méthode *includes()* qui renvoie vrai (true) quand son paramètre est une valeur de la liste sur laquelle elle s'applique.

### Question 4. Ajax et modification du DOM (3 points)

Les quatre listes HTML à items (avec des sous-balises `<li>`) suivantes ont déjà été créées dans le DOM :

```
<ul id="peintres0"></ul>
<ul id="peintres1"></ul>
<ul id="peintres2"></ul>
<ul id="peintres3"></ul>
```

Ecrire le code JavaScript qui rajoute à chacune de ces listes une des valeurs de la liste renvoyée par le serveur Node.js sur la route `localhost:8888/peintresdememenationalite/Dali`

L'item *Picasso* sera donc rajouté à la liste d'identifiant `peintres0`, l'item *Dali* à celle d'identifiant `peintres1` et l'item *Vélasquez* à la liste d'identifiant `peintres2`. La route pourrait être différente et donc l'appel au serveur renverrait une autre liste.