

Planning

Utilisez l'**index.html** et le fichier javascript **lesSoutenances.js**.

Comme vous pouvez le constater vous devez saisir votre code dans le fichier **afficherSoutenances.js** référencé par l'**index.html**.

Est-ce que vous avez des questions ???

Exo 1 : Construire une classe Planning / les colonnes

- le constructeur prend en argument
 - la balise dans laquelle construire le planning
 - un tableau de la liste des jours
 - la largeur de colonne pour chaque jour col-lg-X
 - ces arguments deviennent des propriétés de l'instance
- une méthode construitColonnesJours
 - aucun argument
 - on veut une colonne / div pour chaque journée avec un identifiant "Jour"+jour dans la baliseMere
 - on veut une propriété qui permet de retrouver facilement une colonne pour un jour donnée

Pour affecter une valeur à un attribut d'une balise HTML

```
> balise.setAttribute ( "nomAttribut", valeur );
```

Pour affecter une classe à une balise HTML

```
> balise.className = valeur;
```

Exo 2 : afficher basiquement

- Ajouter un argument au constructeur - creneaux - qui sera ensuite une propriété de l'instance
- Ajouter une méthode trierCreneaux qui triera les creneaux selon l'horaire de passage
- Ajouter une méthode afficherCreneaux qui
 - ajoute une balise Div pour chaque créneau dans la colonne correspondante au jour
 - Y mettre comme texte : l'horaire de passage et le nom de l'étudiant

Exo 3 : Afficher les créneaux avec la structure suivante

```
<div class="panel panel-default">
  <div class="panel-heading">
    <b>08:30</b>
  </div>
  <div class="panel-body">
    <b>Ludovic Goldak</b><br>
    Proxiad (Cure)<br>
    Tuteur : Tarby
  </div>
  <div class="panel-footer">
    M5 - A6 / [e-Services]
  </div>
</div>
```

Exo 4 : remplacer le contenu d'afficherCreneaux en implémentant la classe PanelCreneau de manière à ce que le constructeur suivant fonctionne

>

```
function PanelCreneau ( baliseMere, creneau ) {
  this.baliseMere=baliseMere;
  this.creneau = creneau;
```

```

this.creerPanel();
this.entete=creneau.debut;
this.corps=creneau.etudiant;
this.corps=creneau.societe;
this.corps=creneau.tuteur;
this.pied=Salles[creneau.specialite][creneau.jour]+" / "+creneau.specialite;
}

```

Exo 5 : refactorer tout le code pour passer en Javascript 6

>

Exo 6 :

>E

Rajouter à Planning les méthodes suivantes :

- razCreneaux > pour supprimer tous les créneaux affichés
- afficherCreneauxACondition > qui prend en argument une fonction. Cette fonction doit être faite pour avoir un argument de type creneau.

AfficherCreneauxACondition va afficher les créneaux pour lesquels la fonction en paramètre renverra true.

Au final, on veut que le code suivant fonctionne en n'affichant que les créneaux e-Services

```

window.onload = function () {
    planning = new Planning(
        document.getElementById("Planning"),
        ["04", "05", "06", "07"],
        2,
        lesSoutenances
    );
    planning.razCreneaux();
    planning.afficherCreneauxACondition(
        (creneau) => {
            return (creneau.specialite==Specialites.eServices);
        }
    );
}

```

```

    }
    );
}

```

Exo 7 : rajouter les méthodes ajouterBalise (nom, className) et ajouterTexte à Element (classe mère de toutes les classes “HTML”) pour qu’on puisse ajouter facilement une balise ou du texte à n’importe quelle balise HTML.
> EEE

Exo 8 : ajouter un attribut ID = “Outils” comme suit dans l’index.html
>EEE

```

<div class="row">
  <div id="Outils" class="col-lg-2">
  </div>
  <div id="Planning" class="col-lg-10">
  </div>
</div>

```

Ensuite créer une classe SelectionSpecialite dont le constructeur prend les paramètres suivants - planning et baliseMere - et qui va créer la liste comme suit (ceci se trouve dans la balise Outils précédente si pour baliseMere on a mis comme valeur la balise Outils) :

```

<div class="list-group">
  <a class="list-group-item" href="#">e-Services</a>
  <a class="list-group-item" href="#">IAGL</a>
  <a class="list-group-item" href="#">IVI</a>
  <a class="list-group-item" href="#">MOCAD</a>
  <a class="list-group-item" href="#">TIIR</a>
  <a class="list-group-item" href="#">Recherche</a>
</div>

```

Où un click sur un des items résultera sur l'affichage des créneaux ne concernant que la spécialité "cliquée".
Utiliser pour cela - aussi - l'objet Specialites (lesSoutenances.js).

Ce qui change le démarrage de la page :

```
window.onload = function () {  
    planning = new Planning(  
        document.getElementById("Planning"),  
        ["04", "05", "06", "07"],  
        2,  
        lesSoutenances  
    );  
  
    new SelectionSpecialite(planning, document.getElementById("Outils"));  
}
```

Exo 9 : Faites de la même manière la classe SelectionSalle
>EEEE

Exo 10 : Créer une seule classe Selection - où le constructeur prend les arguments suivants (planning, baliseMere, objetValeurs, fonction) - et où on pourrait avoir le comportement suivant :

> EEEEE

```
new Selection(planning, document.getElementById("Outils"), Specialites,  
    (creneau, specialite) => (creneau.specialite == specialite)  
);  
new Selection(planning, document.getElementById("Outils"), {A2: "M5 - A2", A6: "M5 - A6", A7: "M5 - A7"},  
    (creneau, salle) => (Salles[creneau.specialite][creneau.jour] == salle)
```

);

Terminé: EEEEE