

TP 4 : EcmaScript6

N'est pas lisible partout

Comme on l'a vu en cours, Javascript 2015 n'est pas lisible par tous les navigateurs. En vous référant à la [table de compatibilité](#), vous pourrez essayer au fil de ce TP d'utiliser le code brut sur différents navigateurs et de le compiler si besoin pour qu'il reste accessible sur les plus anciens.

Différentes solutions s'offrent à nous pour maintenir cette portabilité

Installer Babel

Compte tenu de vos droits, on peut installer Babel sur vos machines en suivant la procédure ci-dessous:

```
monPrompt@maMachine-$ mkdir "${HOME}/.npm-packages"
monPrompt@maMachine-$ ls ~/.npm-packages/^C
monPrompt@maMachine-$
monPrompt@maMachine-$ vi ~/.npmrc
monPrompt@maMachine-$ NPM_PACKAGES="${HOME}/.npm-packages"
monPrompt@maMachine-$ PATH="$NPM_PACKAGES/bin:$PATH"
monPrompt@maMachine-$ unset MANPATH
monPrompt@maMachine-$ export MANPATH="$NPM_PACKAGES/share/man:$(manpath)"
monPrompt@maMachine-$ npm install --global babel
/home/infoetu/MeMyself/.npm-packages/bin/babel ->
```

```
/home/infoetu/MeMyself/.npm-packages/lib/node_modules/babel/babel
+ traceur@0.0.111
added 18 packages in 1.644s
```

```
monPrompt@maMachine-$
```

Ou passer par un transpileur online

Votre navigateur n'est pas compatible ES6 ? Ce n'est pas un problème pour aujourd'hui ! Effectivement nous allons utiliser deux scripts qui vont nous transformer à la volée le code ES6 en code compatible avec votre navigateur.

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8" />
  <title>ECMAScript 6 - Demos</title>
</head>
<body>
  <script src="https://google.github.io/traceur-compiler/bin/traceur.js"></script>
  <script src="https://google.github.io/traceur-compiler/src/bootstrap.js"></script>
  <script type="module">
    // Notre code ES6 ici
  </script>
</body>
</html>
```

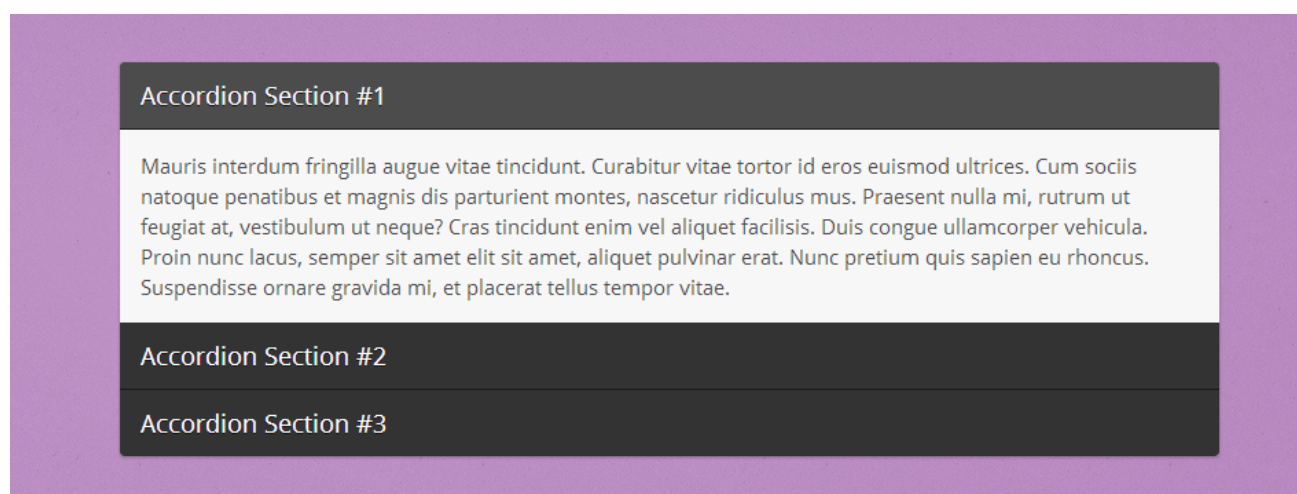
Ici Google convertit le JavaScript à la volée. Ainsi, tout le code JavaScript placé dans la balise *script type="module"* sera automatiquement converti.

Ces éléments seront de moins en moins utiles au fur et à mesure de l'harmonisation des navigateurs. Cela étant, il ne faut pas négliger la vétusté de certains équipements.

Un petit tour d'accordéon

Dans un premier temps, nous allons développer un composant simple de page web qui permet de faire apparaître ou pas du contenu lorsque l'on clique sur cet élément (très utile pour les spoilers ou pour les solutions de TP).

L'accordéon, permet ainsi d'afficher plusieurs rubriques et de n'en faire apparaître qu'une seule à la fois.



Pour réaliser cet accordéon, vous devez utiliser :

- les sélecteurs DOM vus en jQuery,
- la manipulation de styles et/ou classes CSS en JavaScript,
- la capture d'événements `onclick`,
- et la gestion du scope.

Exercice 1

Commençons par créer un accordéon

En vous basant sur la page `pagePourTp4.html` fournie, modifier le bloc JS pour faire apparaître ou non les blocs de texte au clic de l'utilisateur.

Dans ce premier exercice, nous avons créé un accordéon où chaque élément de l'accordéon est identifié. On pourrait maintenant choisir de traiter ainsi tous les paragraphes d'un groupe identifié comme un accordéon.

Exercice 2

Modifiez le code précédent pour que la gestion de l'accordéon porte sur chaque paragraphe d'un bloc de type accordéon

```
<div id= « accordeon »>
  <section>... </section>
  <section>... </section>
  <section>... </section>
</div>
```

Exercice 3

Ajouter le fait que lorsqu'un paragraphe s'affiche, le dernier paragraphe affiché s'efface.

On a donc un joli accordéon dynamique, par contre, comment gérer 2 accordéons en même temps ? Comment en gérer 3, 4 ?

Exercice 4

Ajouter un autre accordéon qui pourra être manipulé indépendamment du premier.

Exercice 5

On souhaite maintenant fournir un composant à d'autres développeurs qui permettent d'instancier un ou des accordéons dans leurs pages web comme bon leur semble.

On va donc créer une classe Accordéon qui affichera un de ses éléments à la demande. Elle serait manipulée de la manière suivante :

```
var accordéon1 = new Accordéon(document.getElementById('accordéon1'));
var accordéon2 = new Accordéon(document.getElementById('accordéon2'));
accordéon1.afficherSection(1);
// => affiche le contenu de la première section du premier accordéon.
```

Pour vous essayer à la création d'une classe, vous pouvez vous entraîner sur [ce petit exercice](#) très simple.

```
class Accordéon {
  constructor(element) {
    // implémenter l'initialisation du composant dans element
    // (cacher les articles, etc...)
  }
  // définir la méthode afficherSection()
}
```

Code à implémenter dans la nouvelle classe

```
var headers = document.getElementsByTagName('header');
var articles = document.getElementsByTagName('article');
```

```
// cette fonction affiche le i-ème article
function afficherArticle(i) {
  articles[i].classList.display= "block";
}
```

```
// cette fonction cache tous les articles
function cacherArticles() {
  for (var i = 0; i < articles.length; i++) {
    articles[i].classList.display="none";
  }
}
```

```
// cette fonction retourne une fonction qui sert à n'afficher que l'i-ème article
function pourAfficherArticle(i) {
  return function(){
    cacherArticles();
    afficherArticle(i);
  };
}

// initialisation des headers
for (var i = 0; i < headers.length; i++) {
  headers[i].onclick = pourAfficherArticle(i);
}
```

N'oubliez pas de transpiler votre code pour le rendre accessible à tous les navigateurs.

Exercice 6

Despoiler universel

Créez une classe DesPoiler qui hérite de la classe Accordéon et qui permet d'afficher tous les éléments de l'accordéon lorsque l'on clique sur un des titres.