

Développement front

MMI 3 – TP#4 S5

Danielo **JEAN-LOUIS**

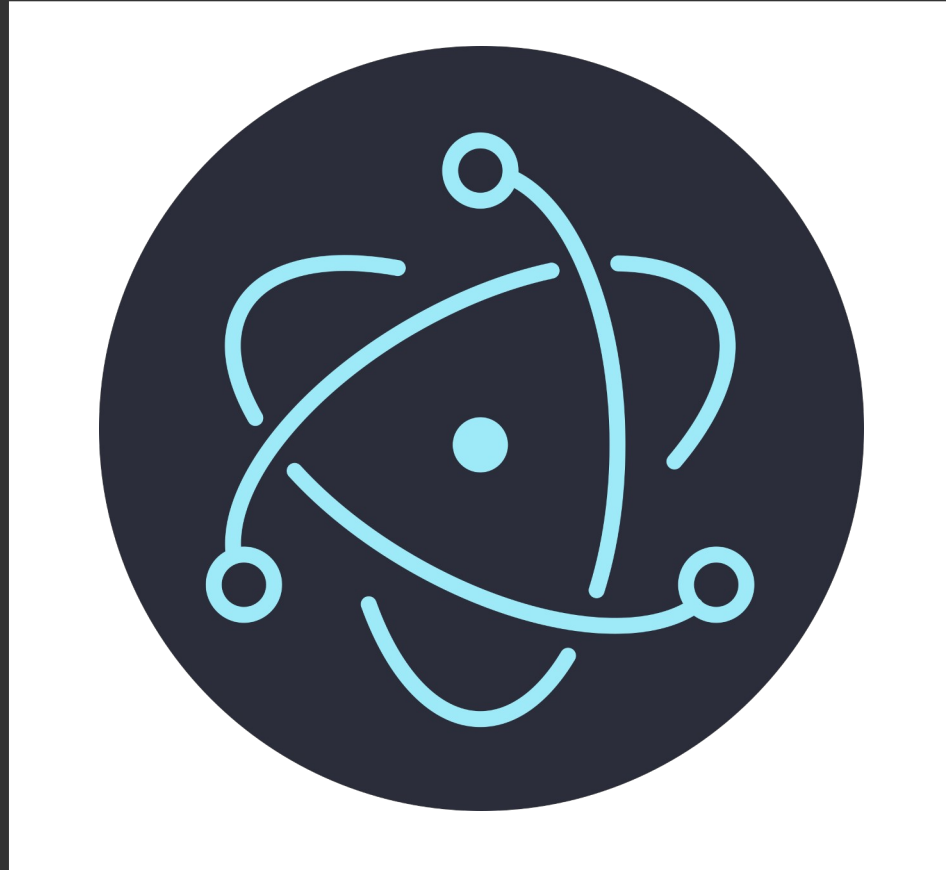
Quels sont les points communs entre ?

- Discord
- Visual Studio Code
- Slack
- Postman

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/template>

Electron



Source(s) :

- <https://www.electronjs.org/apps>
- <https://www.electronjs.org/>

Electron

- Framework front-end permettant de développer des apps pour MacOS, Windows et Linux
- Outil Open Source et gratuit
- Très populaire

Source(s) :

- <https://www.electronjs.org/apps>
- <https://www.electronjs.org/>

Electron

- Utilise les langages HTML, CSS et Javascript
- Un code = Plusieurs exécutables
- Utilise les API natives des systèmes d'exploitation
 - L'application ressemble à une app native

Source(s) :

- <https://www.electronjs.org/apps>
- <https://www.electronjs.org/>

Electron

- Site web embarqué
 - Possibilité d'utiliser la console du navigateur
 - Apps très lourdes avec perfs moyennes
- Basé sur le moteur Chromium

Source(s) :

- <https://www.electronjs.org/apps>
- <https://www.electronjs.org/>

Balise <template>

```
<!DOCTYPE html>
<html lang="en"> event
  <head> ... </head>
  <body>
    <template id="tpl">
      #document-fragment
    </template>
    <script> ... </script>
    <p class="message">Bonjour !</p>
  </body>
</html>
```

Le template est grisé dans la console du navigateur, il n'est donc pas affiché dans la page

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/template>

Balise <template>

```
● ● ●  
  
<template data-tpl-id="tpl">  
  <p class="message">Bonjour !</p>  
</template>  
  
<script>  
  const tpl = document.querySelector("[data-tpl-id=tpl]")  
  // Ajout du contenu du template dans la page  
  // on clone le template pour pouvoir le réutiliser  
  document.body.append(tpl.content.cloneNode(true));  
</script>
```

Ce code permet d'afficher un template contenant une balise <p> avec du texte.

content.cloneNode() permet de cloner le contenu du template

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/template>
- <https://developer.mozilla.org/fr/docs/Web/API/Node/cloneNode>

Méthode .append()

- Méthode javascript
- Permet d'ajouter une balise ou du texte à l'intérieur d'une autre balise
- Utilisable sur n'importe quelle balise HTML

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/API/Element/append>

Méthode `.cloneNode()`

- Méthode javascript
- Copie le nœud sur lequel la méthode a été appelée
- Utilisable sur n'importe quelle balise
- Contrairement à la balise `<template>`, il n'y a pas besoin d'utiliser la propriété "content"

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/API/Node/cloneNode>

Balise <template>

- Limite les erreurs lors d'ajouts de nouvelles balises dans la page
- Réutilisable
- Solution de substitution à la place d'un framework javascript
 - Pour un petit projet

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/template>

Pratiquons ! - Template (Partie 1)

Pré-requis :

- Avoir la ressource ressources/template

A télécharger ici :

<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/developpement-front-s5/travaux-pratiques/numero-3/ressources>

Bonnes pratiques

- Nommez bien vos templates
 - Utilisation d'id ou de data-attribute

Récupération d'éléments

- `.querySelector()`
 - Retourne le premier élément trouvé
- `.querySelectorAll()`
 - Retourne **tous** les éléments trouvés
 - Retourne un itérateur et non un tableau
- Utilise la syntaxe des sélecteurs CSS

Récupération d'éléments

- Utilisable sur n'importe quelle balise HTML
 - Limite la portée de la méthode
- Évitez les méthodes : `getElementById()` ou `getElementsByTagName()`, elles sont moins polyvalentes et modernes

Récupération d'éléments



```
// [...]
```

```
const item = tpl.content.cloneNode(true)
```

```
let paragraph = item.querySelector("p");
```

```
paragraph.textContent = "mon texte";
```

```
document.body.append(item);
```

Ici, nous modifions le texte de la balise `<p>` contenue dans un template

Pratiquons ! - Template (Partie 2)

Pré-requis :

- Avoir la ressource ressources/template

A télécharger ici :

<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/developpement-front-s5/travaux-pratiques/numero-3/ressources>

Méthode `.cloneNode()` - Suite

- Permet de faire un clone “profond” ou “simple”
 - Désigne si la copie comprend les sous-enfants

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/API/Node/cloneNode>

Méthode .cloneNode() - Suite

```
● ● ●  
  
<template data-tpl-id="tpl">  
  <article class="mt-3">  
    <p class="bg-red-200"></p>  
  </article>  
</template>  
  
<script>  
  const tpl = document.querySelector("[data-tpl-id=tpl]");  
  const tplCopy = tpl.content.cloneNode(true)  
</script>
```

La variable `tplCopy` contient une clone profond du template.

Ainsi, la balise `<p>` est dans la variable.

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/API/Node/cloneNode>

Méthode .cloneNode() - Suite

La variable tplCopy est un clone non-profond du template

Ainsi, la balise <p> **n'est pas** dans la variable et donc inaccessible via querySelector().

```
<template data-tpl-id="tpl">
  <article class="mt-3">
    <p class="bg-red-200"></p>
  </article>
</template>

<script>
  const tpl = document.querySelector("[data-tpl-id=tpl]");
  const tplCopy = tpl.content.cloneNode(false)
</script>
```

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/API/Node/cloneNode>

Pratiquons ! - Template (Partie 3)

Pré-requis :

- Avoir la ressource ressources/template

A télécharger ici :

<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/developpement-front-s5/travaux-pratiques/numero-3/ressources>

Questions ?

