

# Développement front

MMI 3 – TP#3 S5

Danielo **JEAN-LOUIS**

# Balise <template>

- Nouveauté HTML5 gérée par tous les navigateurs
- Permet de définir des gabarits de code HTML
  - Affichés et remplis ensuite en javascript
- **Contenu non affiché/chargé par le navigateur**
  - Aucune incidence sur le temps de chargement

Source(s) :

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

# Balise <template>



```
<template>
  <li class="article bg-sky-500 text-3xl">
    <div>
      
    </div>
    <p>Mon template</p>
  </li>
</template>
```

Définition d'un template "squelette" à destination d'une balise <ul>/<ol>

Source(s) :

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

# Balise <template>

- Peut avoir du contenu prédéfini :
  - Lien, CSS, script, texte...
- Doit respecter les règles du code HTML vues jusqu'à présent
- Pas de limite d'imbrications ou de template sur une page

Source(s) :

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

# Balise <template>

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

Source(s) :

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

# Balise <template>

```
<!DOCTYPE html>
<html lang="en">
  <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>

- Impossibilité d'attacher des évènements sur les éléments (ils n'existent pas). Solutions :
  - Attendre leur affichage
  - Utiliser la délégation d'évènements

## Source(s) :

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



# Balise <template>



```
<template data-template-id="hello-message">
  <p class="message">
    Bonjour !
  </p>
</template>
```

Création du template. On lui met un data-  
attribut pour le récupérer plus facilement



JS

```
const tplRaw = document.querySelector("[data-template-id='hello-message']");
const tpl = tplRaw.content.cloneNode(true);
// [Modification du template]
document.body.append(tpl);
```



Récupération du template et ajout dans la  
page

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
  - A condition qu'elle puisse accepter du contenu

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
- document.importNode est une alternative
- **Ne copie pas les évènements**

Source(s) :

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

# Méthode .cloneNode()



JS

```
const element = document.querySelector("selector");  
const elementCopy = element.cloneNode(true);
```

On clone une balise pour la réutiliser plus tard

Source(s) :

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

# Balise <template>



JS

```
const newDiv = document.createElement("div");  
const newImage = document.createElement("img");  
newImage.setAttribute("src", "[...]");  
newImage.setAttribute("alt", "");  
  
newDiv.appendChild(newImage);
```

Les deux codes font la même chose, mais le javascript pur est plus verbeux



```
<template>  
  <div>  
    <img src="" alt="">  
  </div>  
</template>
```

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://github.com/DanYellow/cours/raw/refs/heads/main/s5-developpement-front/travaux-pratiques/numero-3/s5-developpement-front\\_travaux-pratiques\\_numero-3.ressources.zip](https://github.com/DanYellow/cours/raw/refs/heads/main/s5-developpement-front/travaux-pratiques/numero-3/s5-developpement-front_travaux-pratiques_numero-3.ressources.zip)

# Bonnes pratiques

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

# Récupération d'éléments

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

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/API/Document/querySelector>
- <https://developer.mozilla.org/fr/docs/Web/API/Document/querySelectorAll>



# 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

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/API/Document/querySelector>
- <https://developer.mozilla.org/fr/docs/Web/API/Document/querySelectorAll>

# Récupération d'éléments



JS

```
const item = tpl.content.cloneNode(true);  
const paragraph = item.querySelector("p");  
paragraph.textContent = "Mon nouveau texte";  
  
document.body.append(item);
```

Ici, nous récupérons et modifions le texte de la balise <p> contenue dans un template  
Note : la modification doit se faire **avant** la méthode append()

# Pratiquons ! - Template (Partie 2)

Pré-requis :

- Avoir la ressource ressources/template

A télécharger ici :

[https://github.com/DanYellow/cours/raw/refs/heads/main/s5-developpement-front/travaux-pratiques/numero-3/s5-developpement-front\\_travaux-pratiques\\_numero-3.ressources.zip](https://github.com/DanYellow/cours/raw/refs/heads/main/s5-developpement-front/travaux-pratiques/numero-3/s5-developpement-front_travaux-pratiques_numero-3.ressources.zip)

## Méthode `.cloneNode()` - Suite

- Accepte un booléen en paramètre
  - `true` : clone profond, tous les enfants sont clonés
  - `false` : clone simple, ne clone que la première balise du template
- **Ne clone (toujours) pas les évènements**

Source(s) :

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

## Méthode `.cloneNode()` - Suite

- Évitez de mettre des id dans vos templates car ils seront également dupliqués
  - Et on ne peut (toujours) pas avoir des ids identiques

Source(s) :

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

# Méthode .cloneNode() - Suite



```
<template data-template-id="tpl">
  <div class="bg-sky-500 p-4">
    <p class="message">
      Bonjour !
    </p>
  </div>
</template>
```

La variable tpl contient un clone profond du template.

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



```
const tplRaw = document.querySelector("[data-template-id='tpl']");
const tpl = tplRaw.content.cloneNode(true);
```

Source(s) :

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


# Méthode .cloneNode() - Suite

La variable tpl est un clone non-profond du template

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



```
<template data-template-id="tpl">
  <div class="bg-sky-500 p-4">
    <p class="message">
      Bonjour !
    </p>
  </div>
</template>
```



```
const tplRaw = document.querySelector("[data-template-id='tpl']");
const tpl = tplRaw.content.cloneNode(false);
```

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://github.com/DanYellow/cours/raw/refs/heads/main/s5-developpement-front/travaux-pratiques/numero-3/s5-developpement-front\\_travaux-pratiques\\_numero-3.ressources.zip](https://github.com/DanYellow/cours/raw/refs/heads/main/s5-developpement-front/travaux-pratiques/numero-3/s5-developpement-front_travaux-pratiques_numero-3.ressources.zip)



**Questions ?**

