Développement front

MMI 3 – TP#3 S5

IUT CERGY-PONTOISE

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 data-template-id="destination">
  <li class="mb-3">
    <figure>
      <img>
    </figure>
    <p></p>
  </li>
</template>
```

Définition d’un template « squelette » à destination d’une balise

Source(s) :

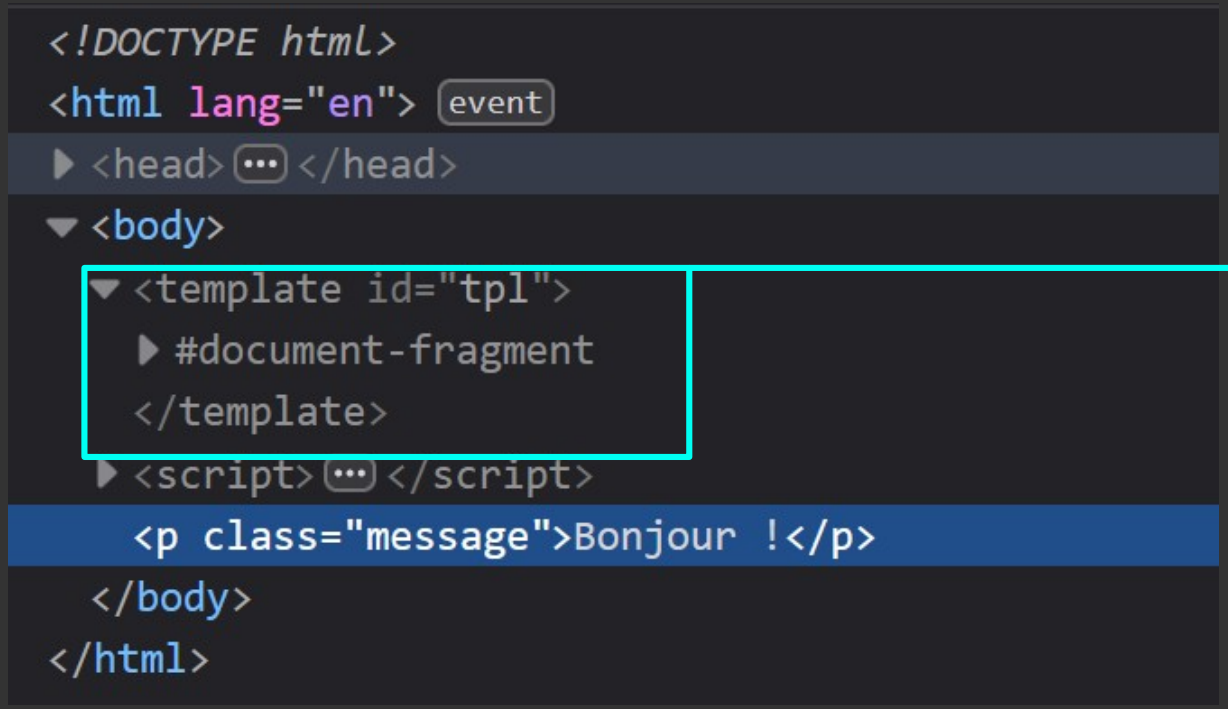
- https://developer.mozilla.org/fr/docs/Web/HTML/Element/template

Balise <template>

- Peut avoir du contenu prédéfini :
  - Lien, CSS, script, texte...
- Doit respecter les règles du code HTML vues précédemment
- 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>



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>

```html
<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 ?