

DOM : Modifier le document

Un élément possède des enfant, parmi ces enfant il existe un type un peu particulier : `TextNode`.

Certains éléments (par exemple une balise) possède du texte en tant qu'enfant.

Il est possible en utilisant `element.innerText` de manipuler de contenu textuel d'une balise

innerText vs innerHTML

Pour changer ou récupérer le contenu textuelle il faut utiliser `element.innerText` :

```
// On récupère le premier paragraph
const p = document.querySelector('p')

console.log(p.innerText) // Affiche le text à l'intérieur du paragraph

// Change le contenu textuel
p.innerText = 'Comment ça va ?'
```

innerText ne peut pas ajouter, enlever ou éditer ... De la balise HTML ! Attention, je souhaite ajouter une balise `` à l'intérieur de mon paragraph je ne peux pas utiliser

`innerText`

Pour changer le contenu **html** d'un élément, nous devons utiliser `element.innerHTML` :

```
// On récupère le premier paragraph
const p = document.querySelector('p')

console.log(p.innerHTML) // Affiche le html à l'intérieur du paragraph

// Change le contenu html
p.innerHTML = 'Comment <strong>ça va</strong> ?'
```

Supprimer un élément

Pour supprimer un élément de notre page HTML il suffit d'utiliser la fonction `element.remove()` :

```
// On récupère le premier paragraph
const p = document.querySelector('p')

p.remove() // Supprime le paragraph du HTML
```

Créer de nouveau élément

Pour créer une nouvelle balise html en utilisant le DOM, on utilise la fonction

`document.createElement` :

```
// Créer une balise img. Le premier paramètre de createElement
// est le nom de la balise html souhaité
const img = document.createElement('img')
```

Lorsque l'on utilise `document.createElement`, rien ne s'affiche à l'écran, il faudra placer le nouvel élément dans la page html

Pour placer un élément dans une page html, plusieurs solutions s'offrent à nous :

- `element.append` : Ajoute un élément à la fin des enfants d'un autre élément
- `element.prepend` : Ajouter un élément au tout début des enfants d'un autre élément
- `element.insertAdjacentElement` : Permet d'insérer un élément au début, à la fin des enfants mais aussi à côté ou avant

```
// On crée une nouvelle image
const img = document.createElement('img')

// On ajoute un attribut src
img.setAttribute('src', 'http://monimage.com')

// Je souhaiterais afficher mon image à l'intérieur de la
// div avec la classe "test1"
const div = document.querySelector('.test1')

// J'ajoute l'image à la suite de ses enfants
div.append(img)

// J'ajoute l'image au début de ses enfants
div.prepend(img)

// Permet d'ajouter l'image dans plein d'endroits à la fois :
div.insertAdjacentElement('beforebegin', img) // Juste avant
div.insertAdjacentElement('afterbegin', img) // Au début des enfants
div.insertAdjacentElement('beforeend', img) // À la fin des enfants
div.insertAdjacentElement(
  'afterend',
  img,
) // Juste après
``
```