

# Programmation Web – Avancé

JavaScript & Node.js

Génération dynamique d'HTML



Attribution –  
Partage dans les  
Mêmes Conditions  
4.0 International  
(CC BY-SA 4.0)

*Presentation template  
by [SlidesCarnival](#)*



# Génération dynamique d'HTML

J'aimerais écrire de l'HTML de manière dynamique côté client...



## **Génération dynamique d'HTML côté client : les possibilités**

---

- Option A : génération à partir de String représentant l'HTML
- Option B : génération à partir d'éléments du DOM créés



## Option A : génération à partir de String représentant l'HTML

- Utilisation de la propriété **innerHTML** d'un élément HTML DOM

```
let htmlText = `

###### 


```



## Option A : génération à partir de String représentant l'HTML

🕒 Quand ajouter un écouteur d'événement ?

```
// setting the content of the div#page element to add the prepared list
page.innerHTML += htmlText;
// setting event listeners on the <ul> can only be done after that the
// ul elements have been rendered (after page.innerHTML)
const uls = document.querySelectorAll("ul");
uls.forEach((ul) => {
  ul.addEventListener("click", () => {
    alert("you click on me !");
  });
});
```



## Option A : génération à partir de String représentant l'HTML

- ⦿ Danger N°1 associé à la position de l'ajout d'écouteurs d'événements : **oubli du render**

```
// Example of badly positionned listener : adding an event listener on a  
list that has not yet been rendered !  
  
...  
const itemList = document.querySelector("ul");  
itemList.addEventListener("click", () =>  
  alert("badly positionned listener will never be called")  
);  
// update to innerHTML reset the attached event listeners !  
page.innerHTML = htmlText;
```



## Option A : génération à partir de String représentant l'HTML

- ⦿ Danger N°2 associé à la position de l'ajout d'écouteurs d'événements : **re-render**

```
page.innerHTML = htmlText;
// example of badly positionned listener : adding an event listener on a
// list that will be reset by the next rerender of page element !
const itemList = document.querySelector("ul");
itemList.addEventListener("click", () =>
  alert("badly positionned listener will never be called")
);
...
// update to innerHTML reset the attached event listeners !
page.innerHTML += htmlText;
```



## Option B : génération à partir d'éléments du DOM créés

- Création d'un élément HTML :  
**document.createElement()**
- Attacher l'élément au sein du « DOM tree » :  
**appendChild()**

```
// Get reference to the parent element
let page = document.getElementById("page");
// Create the child element
let list = document.createElement("ul");
// Attach the child element to its parent
page.appendChild(list);
```





## Option B : génération à partir d'éléments du DOM créés

● Quand ajouter un écouteur d'événement ?

```
let list = document.createElement("ul");
for (let i = 0; i < LIBRARIES.length; i++) {
  let item = document.createElement("li");
  item.innerText = LIBRARIES[i];
  item.addEventListener("click", onClickHandler);
  list.appendChild(item);
}
page.appendChild(list);
```



## Génération dynamique d'HTML côté client & gestion d'événements

- **DEMO : Génération dynamique d'HTML**  
Différents types de boucles pour générer des listes.