

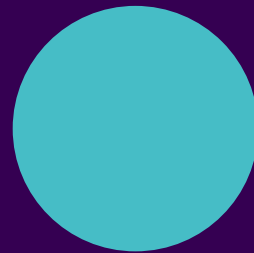
DC 4

Front end :
JavaScript pour le web



1.

Objectifs

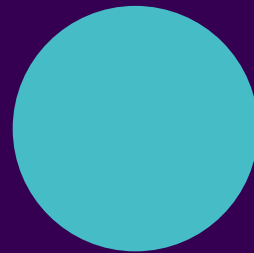


Développer en html/js

- Savoir développer un site avec html/css et javascript
- Savoir modifier dynamiquement une page web
- Lien de rendu du tp fait en cours nom du repo dc4-webjs-nom-nom
- Repos à récupérer

2.

Introduction



DOM (Document Object Model)

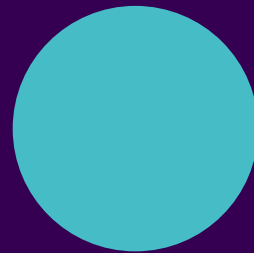
- Interface de programmation qui est une représentation du HTML d'une page web
- Permet d'accéder aux éléments de cette page web et de les modifier avec le langage JavaScript
 - La modification du contenu d'un élément précis
 - La modification du style d'un élément
 - La création ou la suppression d'éléments
 - L'interaction avec les utilisateurs, afin de repérer des clics sur un élément ou encore de récupérer leur nom dans un formulaire

Lier un script JavaScript à du HTML

```
<footer>
  <div class="limitedWidthBlockContainer footerMain">
    <div class="limitedWidthBlock">
      <div>
        
      </div>
      <div>
        <p>10 quai de la charente <br>75019 Paris 19</p>
      </div>
      <div>
        <p>Téléphone : 01 23 45 67 89</p>
      </div>
      <div>
        <p>Email : support@name.com</p>
      </div>
    </div>
  </div>
  <div class="limitedWidthBlockContainer footerSecondary">
    <div class="limitedWidthBlock">
      <p>© Copyright 2021 – 2042 | Openclassrooms by Openclassrooms | All Rights Reserved | Powered by <3</p>
    </div>
  </div>
</footer>
<script src="../../js/script.js"></script>
</body>
</html>
```

3.

Accès au DOM



L'objet document

- Objet - accessible de partout dans votre JavaScript
- Point de départ d'interaction avec le DOM
- Contient les fonctions dont vous aurez besoin pour retrouver les éléments que vous cherchez.

Accéder à un élément

- `document.getElementById()`
 - Prend en paramètre l'id de l'élément que vous recherchez et vous retourne cet élément s'il a été trouvé.
- `document.getElementsByClassName()`
 - Prends une classe CSS en paramètre
- `document.getElementsByTagName()`
 - Prends le nom d'une balise en paramètre (a, button etc)

```
<p id="my-anchor">My content</p>
```

```
const myAnchor = document.getElementById('my-anchor');
```

innerHTML

- Texte représentant un contenu HTML

```
let elt = document.getElementById('main');  
elt.innerHTML = "<ul><li>Élément 1</li><li>Élément 2</li></ul>";
```

```
<div id="main">  
  <ul>  
    <li>Élément 1</li>  
    <li>Élément 2</li>  
  </ul>  
</div>
```

appendChild

- Cette fonction permet d'ajouter un élément à la liste des enfants du parent depuis lequel la fonction est appelée.

```
const newElt = document.createElement("div");  
let elt = document.getElementById("main");  
  
elt.appendChild(newElt);
```

Supprimez et remplacez des éléments

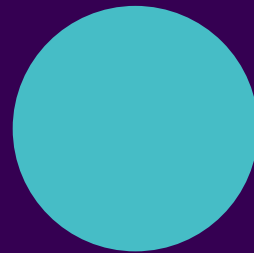
- `removeChild()` - supprime un élément
- `replaceChild()` - remplace un élément

```
const newElt = document.createElement("div");  
let elt = document.getElementById("main");  
elt.appendChild(newElt);
```

```
elt.removeChild(newElt);    // Supprime l'élément newElt de l'élément elt  
elt.replaceChild(document.createElement("article"), newElt);    // Remplace l'élément newElt par un nouvel élément de type article
```

4.

Récupération de
données



Événement

- Un événement est une réaction à une action émise par l'utilisateur, comme le clic sur un bouton ou la saisie d'un texte dans un formulaire.
- Un événement en JavaScript est représenté par un nom (click , mousemove ...) et une fonction que l'on nomme une callback.

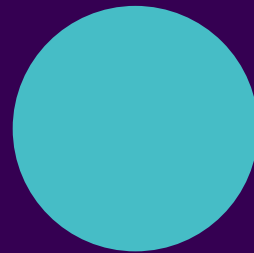
Ecouter un événement

- `addEventListener(<event>, <callback>)` `element.addEventListener('click', onClick);`

```
const elt = document.getElementById('mon-lien'); // On récupère l'élément sur lequel on veut détecter le clic
elt.addEventListener('click', function() { // On écoute l'événement click
|   elt.innerHTML = "C'est cliqué !"; // On change le contenu de notre élément pour afficher "C'est cliqué !"
});
```

5.

Stockage locale



localStorage et sessionStorage

- Micro bdd stocké dans le navigateur (5Mo/site)
- forme de chaînes de caractères (des paires clé/valeur)
- localStorage
 - alternative à l'utilisation des sessions en PHP, il est plus rapide à utiliser puisqu'il ne nécessite pas de requête HTTP
- sessionStorage
 - maintient le stockage seulement lorsque que la fenêtre du navigateur est active

Manipulation

- Stockage

```
localStorage.setItem("prenom", "dany");
```

- Lecture

```
localStorage.getItem("prenom");
```

- Suppression

```
localStorage.removeItem("prenom");
```

Exemple

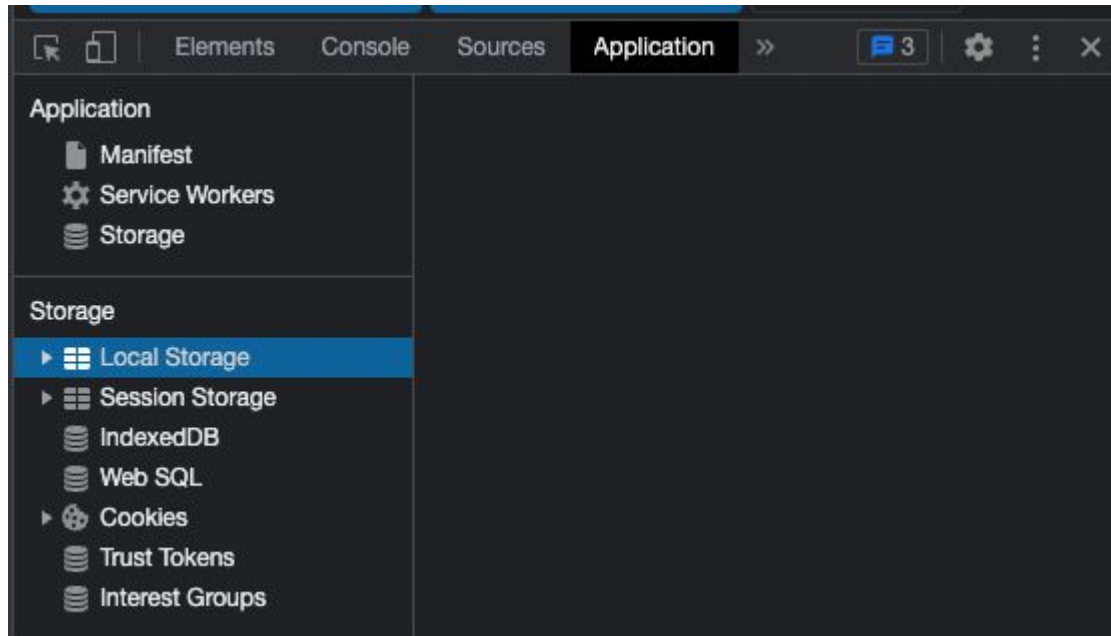
- Écriture

```
let objJson = {  
  prenom : "dany",  
  age : 30,  
  taille : 170  
}  
let objLinea = JSON.stringify(objJson);  
localStorage.setItem("obj",objLinea);
```

- Lecture

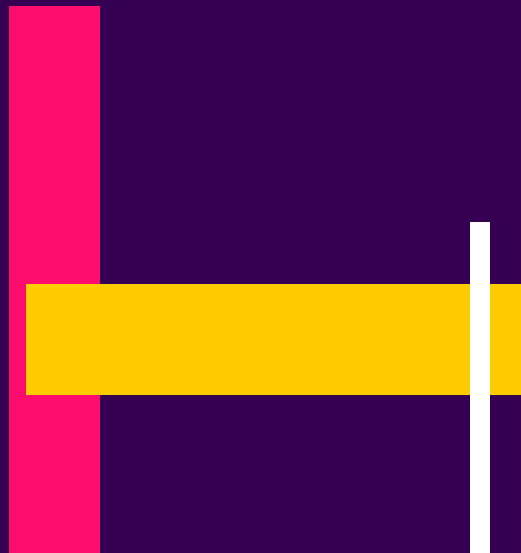
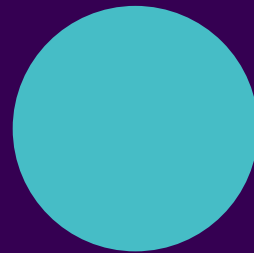
```
let objLinea = localStorage.getItem("obj");  
let objJson = JSON.parse(objLinea);  
alert(objJson.age) // renvoie 30
```

Exemple



6.

Paramètre d'URL

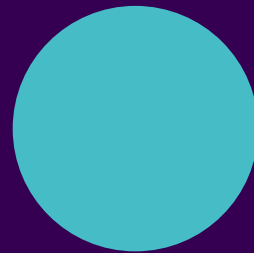


Récupération d'un paramètre dans une URL

```
var str = "https://waytolearnx.com/t.html?name=alex-baptise&age=25&address=paris";  
  
var url = new URL(str);  
  
var search_params = new URLSearchParams(url.search);  
  
if(search_params.has('name')) {  
    var name = search_params.get('name');  
    console.log(name)  
}
```

6.

Requêtes



Get

```
fetch('https://api.example.com/data')  
  .then(response => response.json())  
  .then(data => {  
    // Traitement des données de réponse  
    console.log(data);  
  })  
  .catch(error => {  
    // Gestion des erreurs  
    console.error('Une erreur s\'est produite:', error);  
  });
```


Post

```
const url = 'https://api.example.com/data';
const data = {
  username: 'john.doe',
  password: 'secret123'
};

fetch(url, {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(data)
})
  .then(response => response.json())
  .then(responseData => {
    // Traitement de la réponse
    console.log(responseData);
  })
  .catch(error => {
    // Gestion des erreurs
    console.error('Une erreur s\'est produite:', error);
  });
```

Ressources

**Pages
dynamiques**

OC

Cours web js

OC

Cours debug

OC

URLSearchParams

Doc

LocalStorage

Doc

Memo

Dev Web