

Javascript

Modules

Généralités

L'écriture en modules permet une meilleure organisation du code en le scindant en différents fichiers, qui échangent des fonctionnalités via les instructions `import` et `export`.

- Nécessite un serveur de développement (ex : [Live Server](#))
- Exécution en [mode strict](#)
- Les différents modules ne partagent pas le même contexte

Généralités

Avant l'introduction des modules, tous les scripts partageaient le même espace global

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8"/>
    <title>Ma page</title>
  </head>
  <body>
    <script src="script1.js"></script>
    <script src="script2.js"></script>
    <script>
      alert(maVariable); //tata
    </script>
  </body>
</html>
```

script1.js

```
var maVariable = "toto";
```

script2.js

```
var maVariable = "tata";
```

Généralités

Pour pallier ce problème, on utilisait des IIFE (Immediately Invoked Function Expression).

```
(function() {  
  "use strict";  
  
  function affiche() {  
    window.alert("un clic s'est produit");  
  }  
  
  let div = document.getElementById("maDiv");  
  
  div.addEventListener("click", affiche);  
  
  window.variableApartager = "quelque chose à partager";  
  
})();
```

Le code est encapsulé dans une fonction anonyme aussitôt exécutée, ce qui protège les variables de l'espace global.

Exemple d'utilisation

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8"/>
    <title>Ma page</title>
    <script type="module" src="script1.js"></script>
  </head>
  <body>
    <!-- Contenu de la page -->
  </body>
</html>
```

script1.js

```
import { test, nom } from "./script2.js"
```

script2.js

```
export const nom = "une valeur à partager";

export function test() { /* une fonctionnalité à partager */ }
```

Export par défaut

script1.js

```
export default function test() { /* une fonctionnalité à partager */ }
```

script2.js

```
import nomAuChoix from "./script1.js"
```

Renommage

script1.js

```
export function test() { /* une fonctionnalité à partager */ }
```

script2.js

```
import { test as monChoixDeNom } from "./script1.js"
```

Objet module

script1.js

```
export const nom = "une valeur à partager";  
  
export function test() { /* une fonctionnalité à partager */ }
```

script2.js

```
import * as Module from "./script1.js"  
  
Module.nom // "une valeur à partager"  
Module.test()
```


Aggrégation de modules

utils/fct1.js

```
export default function fct() { /* ... */ }
```

utils/fct2.js

```
export default function fct() { /* ... */ }
```

utils/index.js

```
export { default as fct1 } from "./utils/fct1.js"  
export { default as fct2 } from "./utils/fct2.js"
```

script.js

```
import { fct1, fct2 } from "./utils/index.js"
```

Chargement dynamique

Permet de charger la ressource uniquement si besoin

script1.js

```
export function fct() { /* ... */ }
```

script2.js

```
const button = document.getElementById("monBouton");

button.addEventListener("click", async() => {
  const Module = await import("./script1.js");
  Module.fct();
})
```