

# Développement front avancé

MMI 3 – TP#3 S6

Danielo **JEAN-LOUIS**

# Import de scripts

- Utilisation de la balise `<script>`
- Utilisation de modules
  - Via npm et un bundler (S5)
  - L'attribut `type="module"`
- Et...

# importmap

- Nouveauté gérée par tous les navigateurs depuis 2023
- Attribut de balise `<script>`
- Permet de lister les modules à charger et où les télécharger sous forme de dictionnaire

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/script/type/importmap>

# importmap



```
<script type="importmap">
{
  "imports": {
    "lodash": "https://cdn.skypack.dev/lodash",
    "moment": "https://cdn.skypack.dev/momentjs"
  }
}
</script>
```

Ici, nous définissons deux modules : lodash et moment  
Note : il n'est pas possible de mettre le json dans un fichier externe

# importmap



JS

```
1 import lodash from 'lodash';
```

On importe notre module comme l'accoutumée  
(On n'oublie pas de rajouter "type=module" pour la balise <script>)

# Importmap - Avantages

- Simplification de la gestion des modules
  - Raccourcissement des noms
  - Possibilité de gérer facilement plusieurs versions du même module
- Centralisation des versions de modules
- Pas besoin de bundlers ou de npm/yarn

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/script/type/importmap>

# Pratiquons ! - importmaps (Partie 1/2)

Pré-requis :

- Avoir la ressource `resources/importmap`

A télécharger ici :



# Importmap - Scopes

- Possibilité de charger les modules en fonction de l'emplacement du module
- Définition d'une clé "scopes" dans l'importmap

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/script/type/importmap>

# Importmap - Scopes



```
1 <script type="importmap">
2 {
3   "imports": {
4     "lodash": "https://cdn.jsdelivr.net/npm/lodash@4.17.21/+esm"
5   },
6   "scopes": {
7     "/test/": {
8       "utils": "../test/utils.js"
9     }
10  }
11 }
12 </script>
```

Le module “utils” ne sera disponible que si le module est dans un dossier “test”

# Importmap - Scopes

- Si un module matche plusieurs modules identiques, le navigateur utilisera celui avec le chemin le plus précis

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/script/type/importmap>

# Importmap - Scopes

```
1 <script type="importmap">
2 {
3   "imports": {
4     "lodash": "https://cdn.jsdelivr.net/npm/lodash@4.17.21/+esm"
5   },
6   "scopes": {
7     "/test/": {
8       "lodash": "https://cdn.jsdelivr.net/npm/lodash@4.6.0/+esm"
9     }
10  }
11 }
12 </script>
```

Si un module est contenu dans le dossier “test”, il chargera la version 4.6.0 de lodash, les autres la version 4.17.21

# Pratiquons ! - importmaps (Partie 3)

Pré-requis :

- Avoir la ressource `resources/importmap`

A télécharger ici :

# Importmap - Limites

- Pas adapté pour les gros projets
  - A préférer pour le prototypage
- Impossibilité de :
  - Faire du tree shaking ou du hot reloading
  - Charger plusieurs importmap par page
  - Externaliser l'importmap

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/script/type/importmap>

**Questions ?**

