Environnement de dev

Installation de l’environnement de dev

Mise en place de l'environnement de dev

Installation de l'éditeur de code (IDE)

IDE: Integrated development environment
Visual Studio Code
https://code.visualstudio.com/download

Version 1.17.2 ou supérieure

Une fois installé, vérifiez que l'éditeur se lance bien.

Mise en place de l'environnement de dev

Installation de la dernière version de Node (LTS)

LTS: Long-Term Support
https://nodejs.org/en/download/

Commande pour voir quelle version est installée
node -v

La version de Node devrait s'afficher. Si elle ne s'affiche pas, cela veut dire que Node est mal installé.
La version devrait être 16.17.0

Si la commande node -v ne s'exécute pas bien, vérifiez bien que le répertoire d'installation de Node (le répertoire où se trouve node.exe) est bien inclus dans le PATH

La commande suivante donne le PATH sous Windows
path

Mise en place de l'environnement de dev

Installation du navigateur Chrome

https://www.google.com/intl/fr/chrome/

Vérifiez que la dernière version est installée sur votre poste.

Vous pouvez également utiliser d'autres navigateurs.

FireFox
https://www.mozilla.org/fr/firefox/new/

Edge
https://www.microsoft.com/fr-fr/edge

Vérification de l’environnement

Vérification de l’environnement

Lancez Visual Studio Code.

Ouvrez un terminal d'exécution à partir du menu terminal où à partir du raccourci (Ctrl + Shift + ù)

Vérifiez que la commande suivante renvoie la version de Node
node -v

Vérifiez que la commande suivante renvoie la version de npm (Node Package Manager)
npm -v

Utilisation de Node

Créez un dossier. Ce dossier servira à contenir votre code.

Ouvrez votre IDE (Visual Studio Code).

Dans votre IDE, ouvrez le répertoire que vous avez créé précédemment.
Vous pouvez utiliser le raccourci Ctrl K + Ctrl O
Ou utiliser le menu File > Open Folder…

Ouvrez le terminal d'exécution (Ctrl + Shift + ù)

Exécutez la commande suivante pour générer le fichier de dépendances (package.json)
npm init -y
Un fichier package.json devrait être créé

Utilisation de Node

Le fichier package.json devrait contenir les attributs, name, version, main et scripts

L'attribut "main" contient le chemin du fichier principal à exécuter
L'attribut "scripts" contient les différents scripts pouvant être exécutés à l'aide de la commande 'npm start'.

```
{
  "name": "monapplication",
  "version": "1.0.0",
  …
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node ."
  },
  …
}
```

Utilisation de Node

Vous pouvez ajouter la commande “start” dans “scripts” afin de pouvoir exécuter le programme à l’aide de la commande
npm start

Dans “scripts”, la propriété “start” fournit la commande à exécuter lorsqu’on utilise la commande npm start

```
"start": "node index.js"
```

Signifie que Node exécutera le fichier ‘index.js’ qui se trouve au même niveau que ‘package.json’.

```
"start": "node ."
```

Signifie que Node exécutera le fichier déclaré par l’attribut “main”.

Utilisation de Node

Créez un fichier JavaScript ‘index.js’.
Tous les fichiers JavaScript se terminent avec l’extension ‘.js’.

Ecrivez la ligne suivante dans le fichier créé précédemment

```
console.log("Bonjour");
```

Sauvegardez le fichier ‘index.js’.

Exécutez le fichier dans le terminal d’exécution à l’aide la commande
node index.js
Ou
npm start

Le texte “Bonjour” devrait s’afficher.

Utilisation de Nodemon

Pour ne pas avoir à taper la commande d'exécution à chaque changement, nous pouvons installer Nodemon.

Pour installer Nodemon globalement utilisez la commande
npm i -g nodemon

Nodemon devrait apparaître dans la liste des modules installés globalement.
npm list -g

Pour lancer Nodemon il suffit d'ajouter la commande nodemon dans le fichier 'package.json'.

```
"scripts": {

  "start": "nodemon ."
},
```

nodemon . si le fichier est spécifié dans la propriété "main", sinon nodemon <nom du fichier>

Utilisation de Nodemon

Pour lancer le programme de monitoring, lancer la commande
npm start

Maintenant à chaque modification et sauvegarde de notre fichier principal, le programme s'exécute à nouveau avec les derniers changements.

Pour arrêter Nodemon, cliquez dans le terminal et utilisez la commande Ctrl + C
Un prompt devrait s'afficher pour confirmer l'arrêt de Nodemon

Terminate batch job (Y/N)?

Tapez 'Y' et appuyez sur entrée pour terminer le programme.

Utilisation de TypeScript

Dans Visual Studio code, lancez le terminal d'exécution à l'aide de Ctrl + Shift + ù

Dans le terminal d'exécution tapez la commande
npm i -g typescript
Pour installer TypeScript globalement.

Pour vérifier que TypeScript est correctement installé, tapez la commande suivante
npm list -g typescript

Vous devriez avoir la version 4.8.3 ou une version supérieure.

Utilisation de TypeScript

Générez le fichier de configuration de TypeScript

Pour générer le fichier ‘tsconfig.json’, tapez la commande

npx tsc --init

Pour nos exercices, nous utiliserons la commande suivante

npx tsc --init --rootDir src --outDir output --resolveJsonModule --strictNullChecks --noImplicitAny --sourceMap

La commande précédente pré-remplit le fichier ‘package.json’ avec la configuration souhaitée.

Les sources doivent être placés dans le dossier scr/

Les fichiers transpilés seront placés dans le dossier output/

Utilisation de TypeScript

Pour transpiler les fichiers, vous pouvez utiliser la commande
npx tsc
Cette commande génère les fichiers JavaScript à partir des fichiers TypeScript.

Pour exécuter le programme, tapez la commande (node <fichier JavaScript>)
node ./output/index.js

Vous pouvez également ajouter les scripts suivants dans votre 'package.json'

```
"scripts": {

    "build": "tsc",
    "start": "npm run build && node ./output/index.js"
  },
```

Tapez 'npm run build' pour transpiler vos fichiers en fichiers JavaScript.
Tapez 'npm start' pour transpiler puis exécuter votre programme.

Utilisation de TypeScript

Créez un fichier TypeScript dans le dossier src/
Vous pouvez créer un fichier ‘index.ts’ qui contient la ligne suivante

```
console.log("Bonjour");
```

Tous les fichiers TypeScript se terminent avec l’extension ‘.ts’.

Pour exécuter le fichier précédent tapez la commande suivante dans le terminal d’exécution
npm start
Ou
npx tsc puis node ./output/index.js

Utilisation de TypeScript

Pour ne pas avoir à taper la commande d'exécution à chaque changement, nous pouvons utiliser le flag '-w' de la commande tsc.

npx tsc -w

Vous pouvez également enregistrer la commande en tant que script dans package.json

```
"scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "build": "tsc -w",
    "start": "node ./output/index.js"
  },
```

Il est possible de lancer la commande 'build' sur un terminal et de lancer la commande start sur un autre
npm run build
npm start