

Babel

前回でJavaScriptの言語仕様について学びました。

その中で疑問点があるのでそこに触れつつBabelの学習を行います。

実際に開発する時にどうすれば良いのか

前回ECMAScriptについて触れ、JavaScriptが毎年更新されているのが分かりました。

その反面ブラウザが最新のサポートをしている可能性がないことも分かったかと思います。

では開発の時にどうしたら良いのでしょうか？

1番の理想は開発者側がブラウザのサポート状況に縛られず開発できるのが理想的です。

そんな時に使用されるツールの1つが**Babel**です。

Babelとは

次世代のJavaScriptの標準機能を、ブラウザのサポートを待たずに使えるようにするNode.js製のツールです。

次世代の標準機能を使って書かれたコードを、それらの機能をサポートしていないブラウザでも動くコードに変換（トランスパイル）します。

たとえばES11の言語仕様でコーディングしたとします。ですが対応予定のブラウザの一部が対応してないとします。

そんな時にBabelを使用してES5の言語仕様に変換して対応できます。

そうすればES11に対応していないブラウザも対応できるのと、開発者側も言語仕様を気にせずコーディングができます。

Babelの導入

まずは空のフォルダーを作成してその中でpackage.jsonを作成してください。

作成ができれば下記のコマンドを実行してください。（コマンドの実行は作成したフォルダー内で実行）

```
npm install --save-dev @babel/core @babel/cli
```

インストールが完了するとpackage.jsonにバージョンの表示がされるようになります。

```
// 2021年4月現在なので作業時期によってはバージョンが変わりますが気にしないで作業進めてください。
"devDependencies": {
  "@babel/cli": "^7.13.16",
  "@babel/core": "^7.13.16"
}
```

package.json内のscriptsの部分に下記のスクリプトを追加してください。（testが書かれているのでその下に追加してください。）

```
"build": "babel src -d lib"
```

これでnpm run buildと実行するとbabelが実行されるようになります。

試しにコマンドを実行して見てください。エラーが出るかと思います。

スクリプト内のsrcがあると思いますがそのフォルダーがないのでエラーを起こして見ます。

なのでそのフォルダーを作成しその中でJSファイルを作成し、ES6以上の言語仕様で何か書いておいてください。

現状実行しても書いた内容をそのまま出力しているだけです。

設定が必要なので下記のコマンドを実行してください。

```
npm install @babel/preset-env --save-dev
```

これはプリセットを有効にするためのコマンドです。

それができたらbabel.config.jsonを作成し、下記のコードをコピペしてください。

```
{
  "presets": [
    "@babel/preset-env"
  ]
}
```

ここまでできましたらもう1度buildコマンドを実行して見てください。

ES6以降で書いたJSがES5の仕様になっているかと思います。

このように言語仕様に左右されることなく開発ができます。

注意点

開発環境は会社、案件によって違うので必ずBabelのようなトランスパイルツールが導入されているとは限りません。

もしも導入していない場合、対応ブラウザで動作するようにコーディングする必要があるのでECMAScriptの仕様の把握はできるようになっておいてください。

導入されていないからといって勝手に導入するのはやめてください。（影響範囲が広いので）

もし導入する場合はきちんと許可を得てから導入するようにしてください。

また、自由に導入しても良い案件もありますがその時はドキュメントを残して引き継げるようにしておいてください。

課題

トランスパイルとコンパイルの違いを説明してください。