WEB 講義 参考プリント No.2

TypeScript の基礎

基本的にはサバイバル TypeScript(https://typescriptbook.jp/)を参考にすること。

・環境構築

(コマンド)

```
npm init
npm i -D typescript
npx tsc --init
```

npm i に-D オプションをつけると development(開発用)という意味になる。実行時には使用しないパッケージだから。

npx tsc とは tsc コマンドを実行するためのもの。tsc –init コマンドは tsconfig.json を生成する。tsconfig.json には、typescript のコンパイルオプションが書かれている。

tsconfig.json の中の設定のうち次のようにしておけばよい。

(tsconfig.json)

```
- "target": "es2016",
+ "target": "ESNext",
-  // "outDir": "./",
+ "outDir": "./dist",
```

(場所がわからない場合は Ctrl+F で検索してみよう。)

・コンパイル

index.js ではなく index.ts というファイルを作る。このファイルに下の内容を書く。

(index.ts)

```
function add(a:number, b:number) {
    return a + b;
}
console.log(add(1, 2))
```

(コマンド)

```
npx tsc
```

dist というフォルダが作成され、中に index.js ファイルがあるはず。

これを node コマンドで実行すると..

(コマンド)

```
node dist/index.js
```

もちろん実行される。

・コンパイルの意味

TypeScript は JavaScript に型注釈を与えるための言語であり、実行するには JavaScript ファイルに書き換える必要がある。そのために使う道具が tsc コマンドである。この型注釈について学んでいく。

・コンパイルの自動化

package.json を次のように書き換えてみよう。

(package.json)

```
"scripts": {
--- "test": "echo ¥"Error: no test specified¥" && exit 1"
  },
```

```
"scripts": {
+++ "start": "npx tsc && node dist/index.js"
  },
```

そして npm start コマンドを実行すると

```
> typescript@1.0.0 start
> npx tsc && node dist/index.js
```

このように一つのコマンドで実行まででできる。

・座学

サバイバル TypeScript の「読んで学ぶ TypeScript＞ 値・型・変数 ＞ 変数宣言: let と const」から読んでいこう。

・問題

(1)もちろん次のように宣言するだけで変数の型は勝手に指定してくれるが、

(index.ts)

```
const a=1 //a は number 型
```

自分で指定せよ。(例↓)

```
const a:number=1
```

```
"hello world"
[1, 2, 3]
true
12n
```

(2)関数の返り値の型はどのように指定すればよいか。

(3)高校 3 年生、高校 2 年生、高校 1 年生という string だけが入る type を作成し、変数に注釈をつけることでそれいがいの string が入らないことを確認せよ。

(4)以下の変数に対し getDate()関数を実行するとき、a が undefined の場合実行しないようにするにはどうすればよいか？オプショナルチェーンを用いて記述せよ。

```
let a:Date|undefined;
```

(5)

(i)以下の object に対し a,b という変数を取り出せ。ただし分割代入を使うこと。

(ii)分割代入して取り出した a,b に対し、value1,value2 という別名を定義するにはどうすればよいか？

```
const obj = { a: 1, b: 2 };
```

(6)以下の配列に対しスプレッド構文を用いて console 出力せよ。

```
const a=[1,21,2,23,23233,32,32]
```

ただし次のような出力になる。

```
＞　1 21 2 23 23233 32 32
```

(7)D が C と同じ型になるようにせよ。ただし、インターセクション型を用いること

```
type A={
    id:string,
    name:string
}
type B={
    email:string
}

type C={
    id:string,
    name:string,
    email:string
}

type D=???
```

(8) 以下のコードについて、指定された部分を三項演算子を用いることで一行で表せ。

```
const point = 30
type Grade = "A" | "B" | "C" | null
```

```
//ここから
let grade: Grade = null

if (point >= 90) {
    grade = "A"
} else if (point >= 80) {
    grade = "B"
} else {
    grade = "C"
}
//ここまで
```

・総合課題

(1)Express というパッケージを使ってみよう。下のコードを動かしたのち、localhost:3000 というサイトをブラウザで開く。(ググりながらやってください。)

```
import express from "express"

const app = express()

app.get("/", (req, res) => {
  res.send("Hello World")
})

app.listen(3000, () => {
  console.log("Server listening on port 3000")
})
```

(i) デベロッパーツールで network タブを見てみよう。

(ii) app.get の get は何を意味しているか？

(iii) console.log(req.query)などでプロパティの意味を調べてみよう。

・次回

関数,class