

Section3-6 理解度確認テスト②

たかし君のテスト

私は下記のように書きましたが、このコードと違っていても求めたい結果が出力されるのであればそのコードは「正解」と言えます。

```
function takashi1() {
  const kokugo = 80;
  const sansu  = 100;
  const eigo   = 60;

  //平均点を出す
  const average = (kokugo + sansu + eigo) / 3;
  console.log(`今回のたかし君の平均点は ${average} 点です。`);

  //成績の判断
  let grade;
  if (80 <= average) {
    grade = '優';
  } else if ((60 <= average) && (average < 80)) {
    grade = '良';
  } else if ((40 <= average) && (average < 60)) {
    grade = '可';
  } else {
    grade = '不可';
  }
  console.log(`よって成績は「${grade}」でした。`);
}
```

```
if (75 <= average) {  
    console.log('お小遣いアップできました。');  
} else {  
    console.log('お小遣いアップできませんでした。');  
}  
}
```

ここでは、私がレクチャーをした時に、受講者が書いたコードに対しておこなったフィードバックをいくつか紹介します。

インデントを揃えよう

インデントは「字下げ」のことで、適切なインデントを行なうことで、「どこからどこまでが何のかたまりなのか」がわかりやすくなります。

たとえば下記の2つのコードを比較してみましょう。

(A)

```
let grade;  
if (80 <= average) {  
    grade = '優';  
} else if ((60 <= average) && (average < 80)) {  
    grade = '良';  
} else if ((40 <= average) && (average < 60)) {  
    grade = '可';  
} else {  
    grade = '不可';  
}
```

(B)

```
let grade;
if (80 <= average) {
  grade = '優';
} else if ((60 <= average) && (average < 80)) {
  grade = '良';
} else if ((40 <= average) && (average < 60)) {
  grade = '可';
} else {
  grade = '不可';
}
```

(B)のほうが `grade = '優';` が `if (80 <= average)` の中にあることがわかりやすいですね？

スクリプトエディタには自動でインデントを揃えてくれる機能が存在しています。スクリプトエディタで右クリックをして「ドキュメントのフォーマット」を選択すると、自動でインデントを揃えてくれますので、活用してください。

数値の大小比較のみやすさ

下記の2つは同じ意味ですが、どちらが理解しやすいですか？

(a)

```
if ((average >= 60) && (average < 80))
```

(b)

```
if ((60 <= average) && (average < 80))
```

数直線をイメージした時に、左が小さい数、右が大きい数になりますので、コード内でもそのように表現したほうが理解しやすいと思います。(変数・定数を左に書かなければならないというルールはありません)

小 $\xrightarrow{\dots\dots 60 \dots\dots \text{average} \dots\dots 80 \dots\dots}$ 大

そのため、私は**(b)**のほうがわかりやすいのですが、いかがでしょうか？

評価の順番

私のコードでは平均点の高い順にチェックしています。

```
// 点数の高い方から
if ( 80 <= average ) {
    grade = '優';
} else if ((60 <= average) && (average < 80)) {
    grade = '良';
} else if ((40 <= average) && (average < 60)) {
    grade = '可';
} else {
    grade = '不可';
}
```

あるいはこのように平均点の低い順にチェックすることもできます。

```
// 点数の低い方から
if ( average < 40 ){
    grade = '不可';
} else if ((40 <= average) && (average < 60)) {
    grade = '可';
} else if ((60 <= average) && (average < 80)) {
    grade = '良';
} else if (80 <= average) {
    grade = '優';
}
```

ここは好みでいいと思います。別の発想もあるよね、という気付きのために紹介しておきます。

無駄を省いた表記

いまのコードは下記のように「80以上80未満」の両方を書いています。

```
if ( 80 <= average ) {
    grade = '優';
} else if ((60 <= average) && (average < 80)) {
    grade = '良';
} else if ((40 <= average) && (average < 60)) {
    grade = '可';
} else {
    grade = '不可';
}
```

これは

```

if ( 80 <= average ) {
    grade = '優';
} else if ( 60 <= average ) {
    grade = '良';
} else if ( 40 <= average ) {
    grade = '可';
} else {
    grade = '不可';
}

```

と書いても同じであることはわかりますか？

`if (80 <= average)` は「80 以上であるか？」という条件ですが、これが `false` になるということは「80 以上ではない」つまり「80 未満である」ということです。

よって

```

} else if ((60 <= average) && (average < 80)) {

```

の評価がされるときはすでに「80 未満である」ことは満たしているので、

```

else if ( 60 <= average ) {

```

とだけ書いても同じことになります。

出力内容の固定部のところと可変なところ

私の解答例では

```
//分岐によって、文章全体を出力するパターン
if ( average >= 75){
    console.log('お小遣いアップできました。');
} else {
    console.log('お小遣いアップできませんでした。');
}
```

になっていますが、この部分は下記のように書いても同じ結果です。

```
//変更箇所「だけ」を切り出して、出力時に文章を組み立てるパターン
let text;
if ( average < 75 ) {
    text = 'できませんでした。';
} else {
    text = 'できました。';
}
console.log(`お小遣いアップ${text}`);
```

可変の部分を変数に入れておいて、固定文章の中に入れていきます。今回は文章自体が短いのでどちらでも大きな差はないですが、たとえば出力するのが長いメール文章で、変更したいのが一部分だけという場合は可変部分だけを変数にするほうがスマートに書けます。