

DIG プログラミング基礎 プレゼンテーション

Week 5 Lesson 18 中級演習2.

関数 intersection を宣言してください。

学習
ポイント

関数 intersection は、引数(配列型)を2つとり、
アウトプットとして、与えられた配列の両方に
存在する要素だけが入った配列を返す関数

例. intersection([1, 2, 3], [1, 2, 3]) ⇒ [1, 2, 3]
intersection([1, 2, 3], [3, 4, 5]) ⇒ [3]

高階関数を学ぶ前 (for...ofを使う)

```
// ↓↓ 関数宣言 : functionキーワード 関数名(1つ目の引数, 2つ目の引数)
function intersection(array1, array2) {

    // ↓↓ 空の配列を格納した変数(resultArray)を準備
    const resultArray = [];

    // ↓↓ 1つ目の配列内の要素を1つずつ変数number1に格納し、処理を実行
    for (const number1 of array1) {

        // ↓↓ 2つめの配列内の要素を1つずつ変数number2に格納し、処理を実行
        for (const number2 of array2) {

            // ↓↓ もし、number1とnumber2が一致したら、
            if (number1 === number2) {

                // ↓↓ 配列resultArrayの末尾にnumber1の値を要素として追加する
                resultArray.push(number1);
            }
        }
    }

    // ↓↓ 配列resultArrayを返す
    return resultArray;
}
```

インデント

メソッド

array1 : [2, 3], array2 : [1, 2]の場合

resultArray : []

1つ目のfor 1回目

number1 : 2

2つ目のfor 1回目

number2 : 1

2と1を比較 ⇒ 不一致

処理なし

2つ目のfor 2回目

number2 : 2

2と2を比較 ⇒ 一致

resultArrayに2を追加

resultArray : [2]

1つめのfor 2回目

number1 : 3

.

.

高階関数 (Array.filter()) を使うと…

```
// ↓↓ 関数宣言 : functionキーワード 関数名(1つ目の引数, 2つ目の引数)
function intersection(array1, array2) {

    // ↓↓ array1の各要素をnumberに代入し、()内で定義された関数を実行する
    return array1.filter(function (number) {

        // array2の配列内にnumberが含まれるか判定 ⇒ 含まれる場合に新しい配列を生成し、返す
        return array2.includes(number)

    })
}
```

無名関数

array1 : [2, 3], array2 : [1, 2]の場合

number に2 と 3 を代入し、

array2 : [1, 2] に 2 と 3 が含まれているか判定 ⇒ 2が含まれているので、[2]を生成し、返す

文がスッキリ！

配列が新しく生成されるため、予め空の配列を準備する必要がない

更にアロー関数を使うと...

```
function intersection(array1, array2) {  
  
    return array1.filter(number => array2.includes(number))  
  
}
```

更に文がスッキリ！ww

他にも 学習した .map の他、 .forEach や .reduce といった
まだ見ぬ高階関数もあるので、フォローアップ期間で、
勉強できればと思っています。

今後は、ここで学んだこと+αで、
グループ内やチーム内で共有したい
ファイルを閲覧できるwebページを
作ってあげたいと思っています。

その際には、皆様のお力をお借り
することになるかもしれませんが、
その際は、暖かく教えていただけると
幸いです。 WW