

Section3-13 理解度確認テスト④

二次元配列のループ

解答例1: 通常の for 文で書く

```
function seisekiTotal1() {  
    const sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('配列');  
    const data = sheet.getDataRange().getValues();  
    let total = 0;  
  
    // ここに すべての点数を足す処理を書く  
    for(let i=0; i<data.length; i++){  
        total = total + data[i][2];  
    }  
  
    console.log(total); // 460が出力されるはず  
}
```

のように書いてみました。

```
const data = sheet.getDataRange().getValues();
```

が実行されると、data の中身が下記のようにになっています。

```
data = [  
    [たかし, 国語, 80],  
    [たかし, 算数, 100],
```

```
[たかし, 英語, 60],  
[みゆき, 国語, 70],  
[みゆき, 算数, 50],  
[みゆき, 英語, 100]  
]
```

そのあとに data から1つずつ取り出すわけですが、

```
for(let i=0; i<data.length; i++){  
  total = total + data[i][2];  
}
```

のコードで、data[i] が [名前, 教科, 点数] という配列になっています。この3つ目(インデックス2)が「点数」ですので、data[i][2] と書くと 点数 が取り出せます。

また、ちょっとしたテクニックですが、今のコードだと data[i][2] の 2 が何を意味するのか、をコードをみて考えないとなりませんよね？そこで、下記のようにしておくことで「それは点数のインデックスだよ」とわかりやすくなります。

```
// 点数が存在するインデックスは「2」  
const indexScore = 2;  
  
for (let i=0; i<data.length; i++){  
  total = total + data[i][indexScore];  
}
```

解答例2: for...of を使う

同じことを for...of を使って書くこともできます。

```
function seisekiTotal2() {  
    const sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('配列');  
    const data = sheet.getDataRange().getValues();  
    let total = 0;  
  
    // ここに すべての点数を足す処理を書く  
    for(const rowData of data) {  
        const score = rowData[2];  
        total = total + score;  
    }  
  
    console.log(total); // 460が出力されるはず  
}
```

data から一つずつ取り出して rowData に入れます。このときのrowDataは「スプレッドシートの1行分の情報が入った配列」ですので、rowData[2] で C列の値が取得できます。（インデックス 0 がA列）