プログラミング

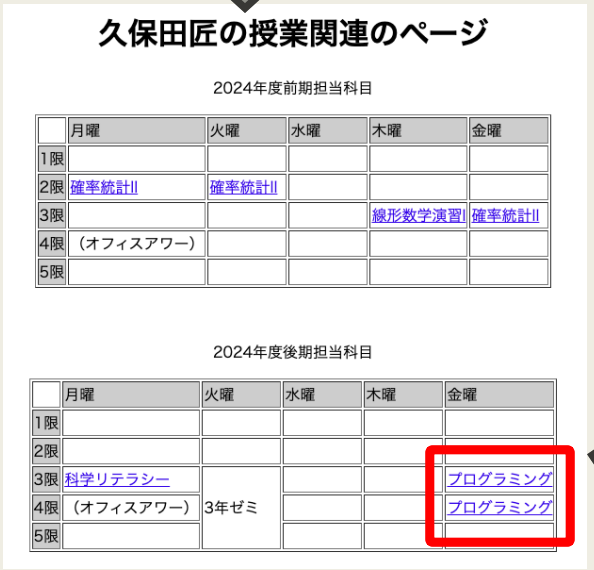
第11回
配列

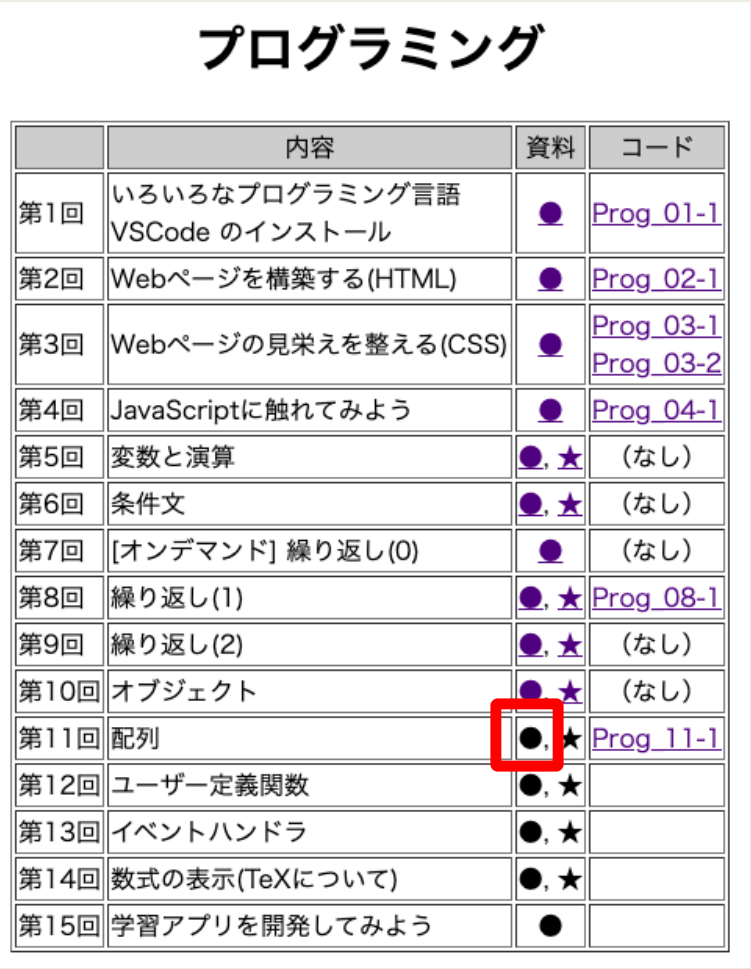
久保田 匠

[準備]授業資料にアクセス

いつもの作業

- 久保田の授業ホームページに資料がアップロードされている。
- まずは「愛教大 数学」と検索してみよう。





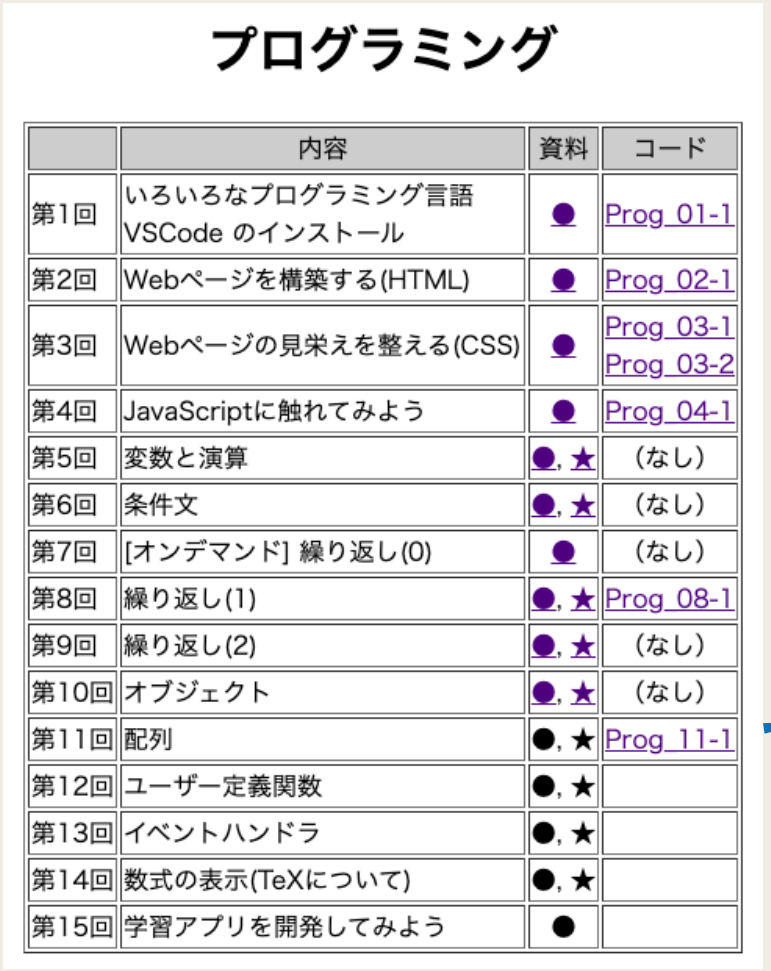
プログラミング

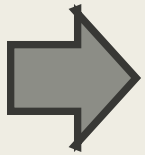
	内容	資料	コード
第1回	いろいろなプログラミング言語 VSCode のインストール	●	Prog_01-1
第2回	Webページを構築する(HTML)	●	Prog_02-1
第3回	Webページの見栄えを整える(CSS)	●	Prog_03-1 Prog_03-2
第4回	JavaScriptに触れてみよう	●	Prog_04-1
第5回	変数と演算	●, ★	(なし)
第6回	条件文	●, ★	(なし)
第7回	[オンデマンド] 繰り返し(0)	●	(なし)
第8回	繰り返し(1)	●, ★	Prog_08-1
第9回	繰り返し(2)	●, ★	(なし)
第10回	オブジェクト	●, ★	(なし)
第11回	配列	●, ★	Prog_11-1
第12回	ユーザー定義関数	●, ★	
第13回	イベントハンドラ	●, ★	
第14回	数式の表示(TeXについて)	●, ★	
第15回	学習アプリを開発してみよう	●	

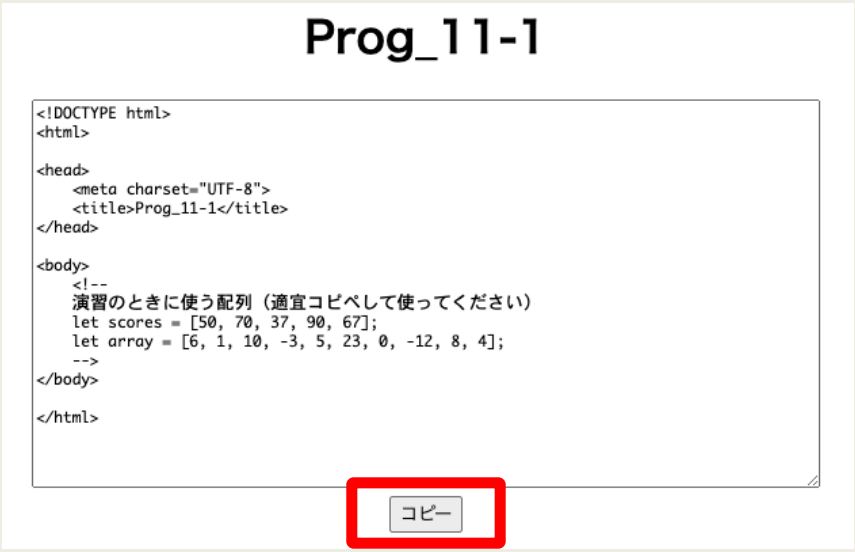
[準備]コードの新規作成①

いつもの作業

- 授業用ホームページからサンプルコードをコピーしよう。





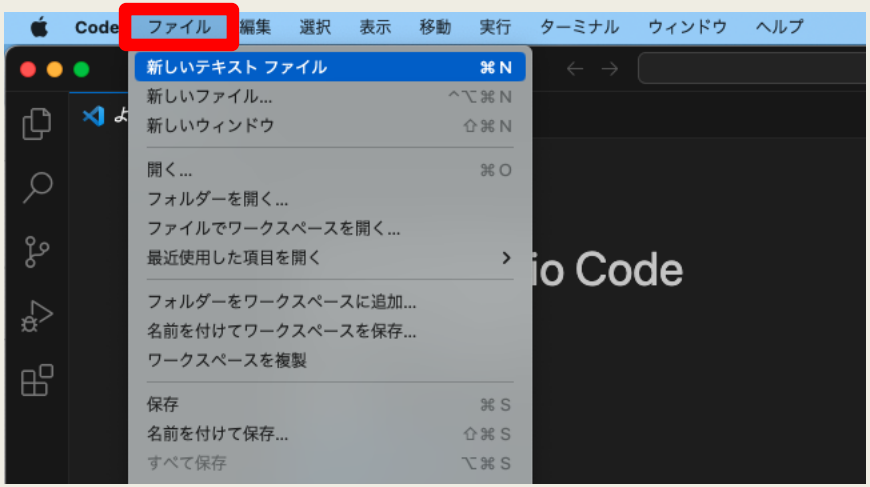


今日は「Prog_11-1」を選択してください。

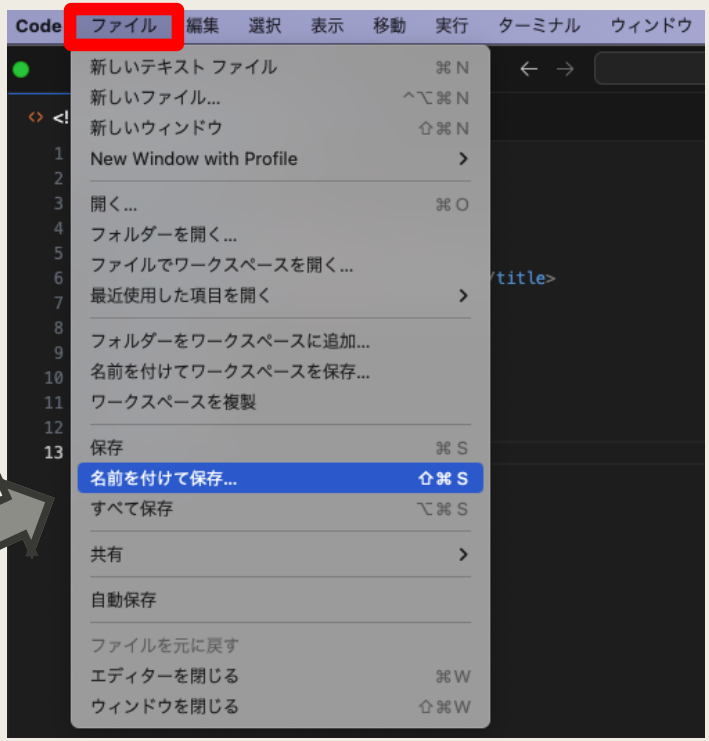
[準備]コードの新規作成②

いつもの作業

- VSCode を起動し「ファイル」から「新しいテキストファイル」を選択。
- そのあと、さきほどコピーした文書をペースト（Ctrl + V）して「名前をつけて保存」。



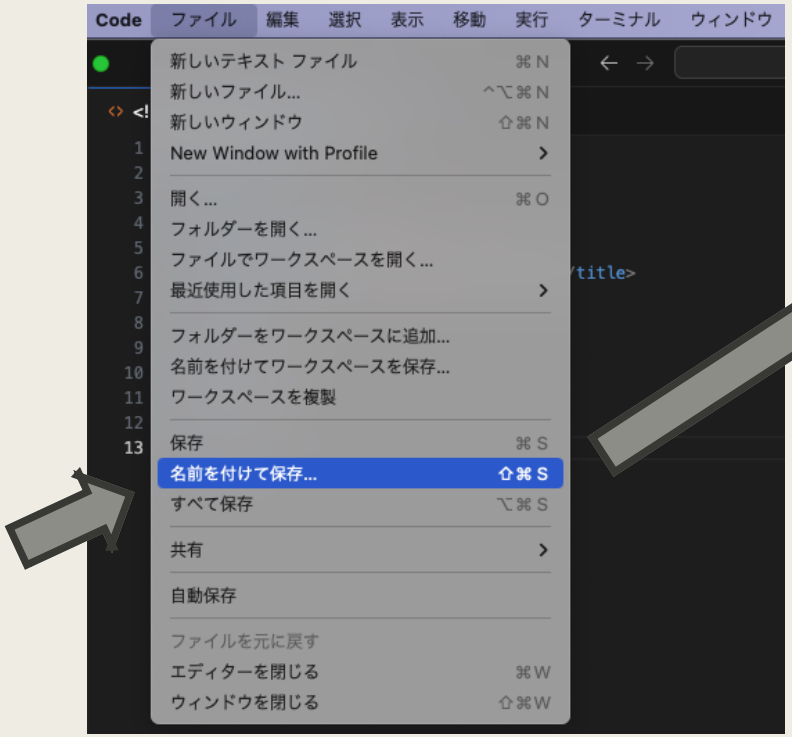
Ctrl + V



[準備]コードの新規作成②

いつもの作業

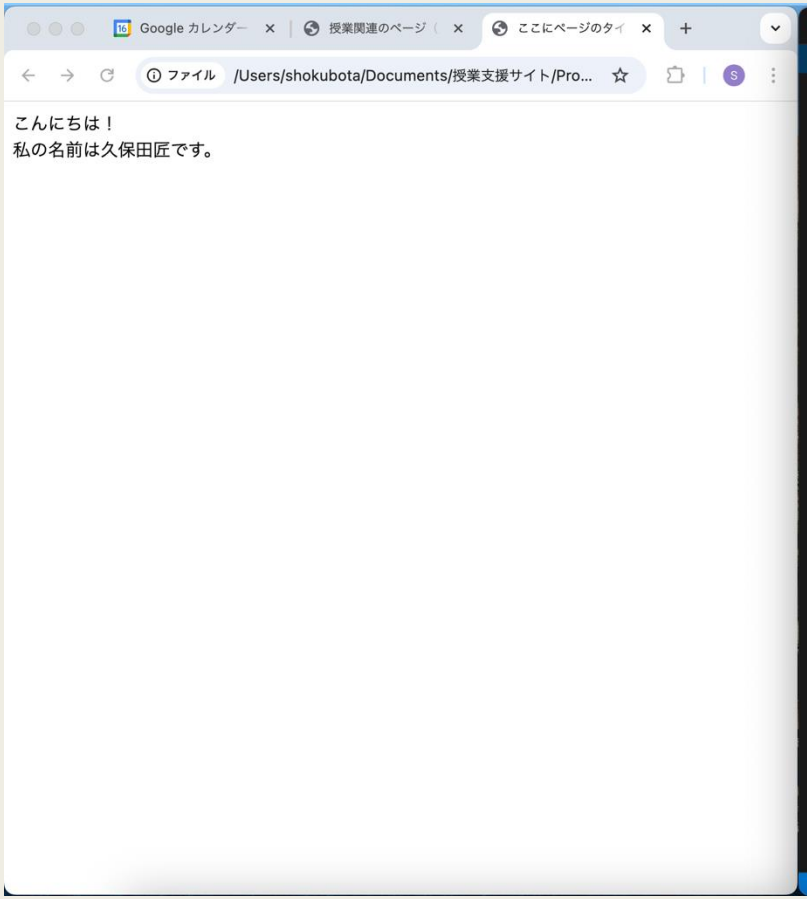
- VSCode を起動し「ファイル」から「新しいテキストファイル」を選択。
- そのあと、さきほどコピーした文書をペースト（Ctrl + V）して「名前をつけて保存」。

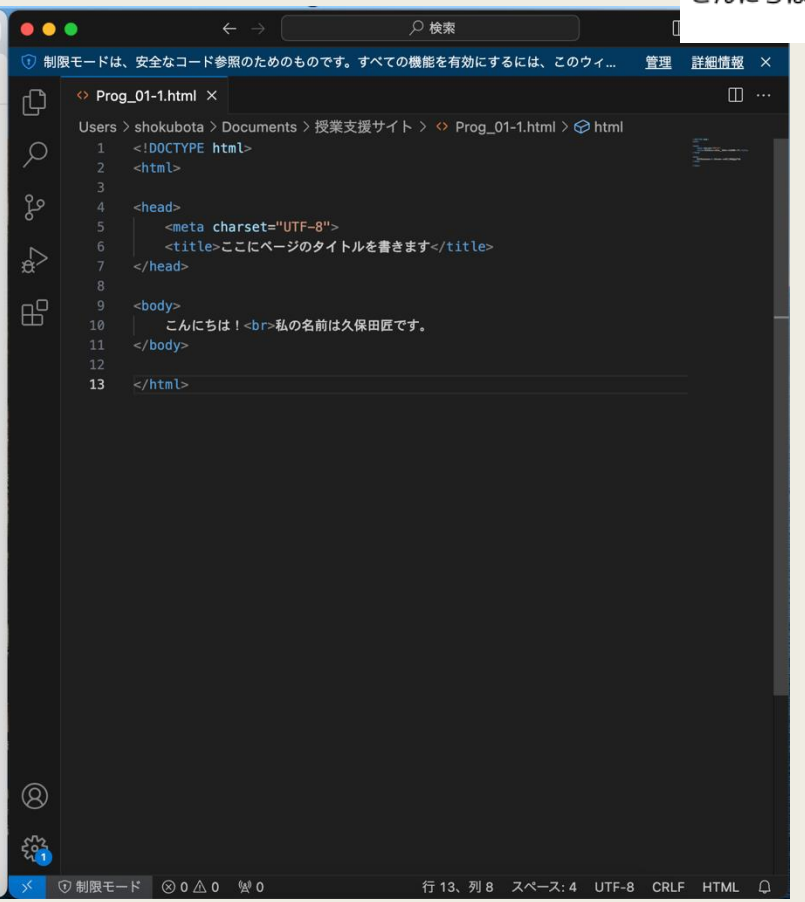


[準備]作業環境を整える

いつもの作業

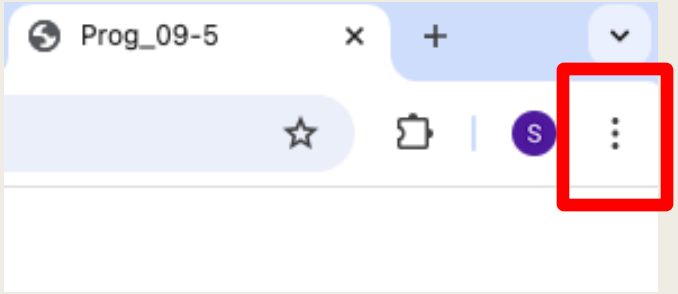
- 保存したhtmlファイルをダブルリックして開いておく。
- PCの画面をふたつに分け、片方はブラウザ、もう片方は VSCode を開いておくと便利。

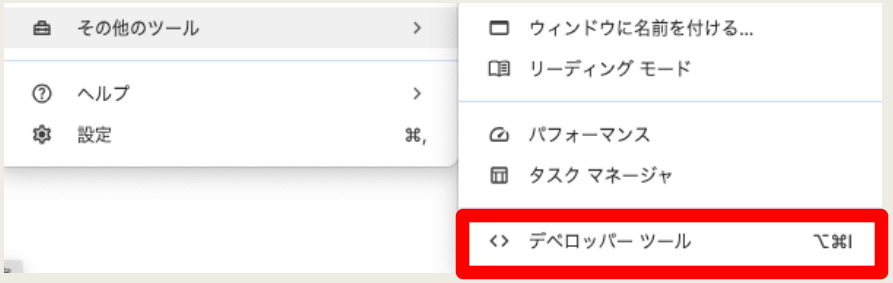


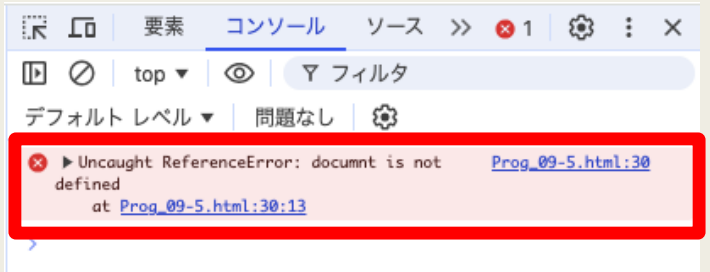


[再掲]デベロッパーツール

- 画面に何も表示されないときや、途中までしか表示されないときはプログラムに間違いがある可能性が高い。
- そのときは「デベロッパーツール」を開き、何行目でエラーが発生しているかを見てみよう。







ショートカットキーもある
Windows → Ctrl + Shift + i
Mac → Option + Command + i

30行目でエラーが発生。
documnt が未定義と言われている
（スペルミスが発生していた）

[復習]オブジェクト

- 深入りせず「使えれば良し」とする。
- 「データ」と「データに関する操作」をセットにしたものを オブジェクト という。
- 「データ」のことを プロパティ といい、「データに関する操作」を メソッド という。
- 例として2次元ベクトルをオブジェクトとして考える。

プロパティ （2次元ベクトルがもつデータ）
x座標
y座標

メソッド （2次元ベクトルに関する操作）
大きさを計算する
他のベクトルとの和や スカラー倍を計算する
他のベクトルとの内積を計算する

[復習]インスタンスの生成（実体化）

- 前回は Math, String, Date オブジェクトを紹介した。
- オブジェクトを使用するときは通常 インスタンスの生成（具体的なデータを作る作業）が必要だが、 Math オブジェクトは例外的に直接利用できる。
  - 2次元ベクトルオブジェクトを例に考えると、インスタンスの生成は「具体的に考える2次元ベクトルを作る」ことに相当する。
- Math オブジェクトのプロパティやメソッドは次の通り。

Mathオブジェクトのプロパティ（数学に関するあらゆる定数）
円周率π
自然対数の底 e
⋮

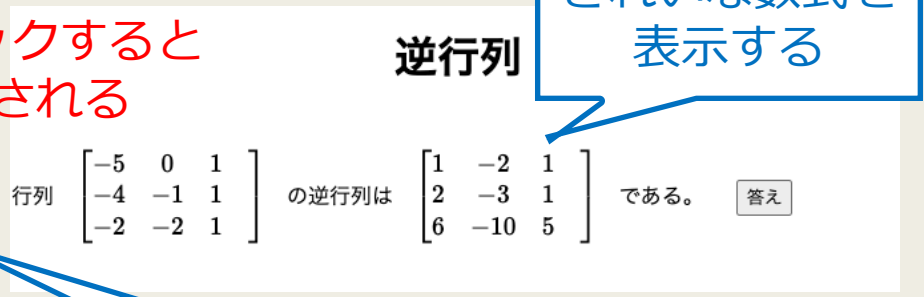
Mathオブジェクトのメソッド（数学に関するあらゆる操作）
絶対値を返す
0以上1未満の乱数を返す
小数点以下を切り捨てる
⋮

Math.random()

Math.floor(a)

[再掲]第10回以降に学ぶこと

ボタンをクリックすると
答えが表示される



[第14回]
きれいな数式を
表示する

[第10回]
乱数を発生させる

[だいたい済]
生成した問題に
対して答えを計算
(透明色で表示)

[第13回]
ボタンを押したときに
特定の処理を行う

[第12回]
処理のかたまりを
定義する

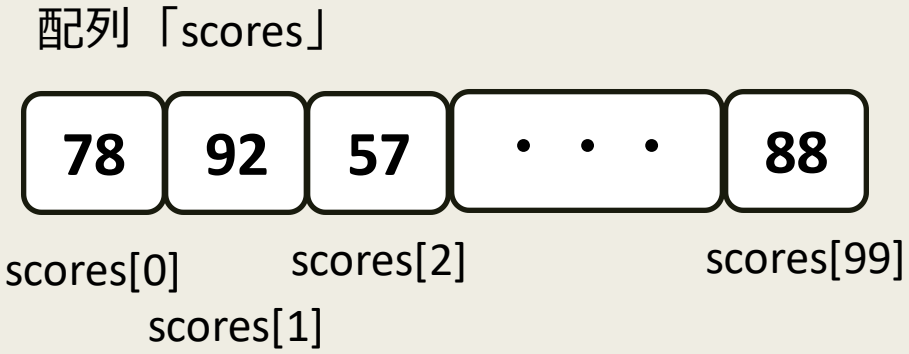
- 第11回では配列を扱う。
- 配列はひとつの変数名で複数のデータをまとめて管理できるようにしたもの。
- 例えば、上の例で配列を使わずにプログラムすると、問題の行列と答えの行列の各成分で合計18個の変数を用意しなければならない。

配列

- 配列とは、添字（そえじ）と呼ばれる番号を使ってひとつの変数名で複数のデータをまとめて管理できるようにしたもの。
- 例えば、100名の学生の点数を管理したいとき、通常の変数を使用すると個別に100個の変数を宣言する必要がある。
- これに対して、配列を使えば scores のようなひとつの変数名だけで100個のデータをまとめて管理できる。

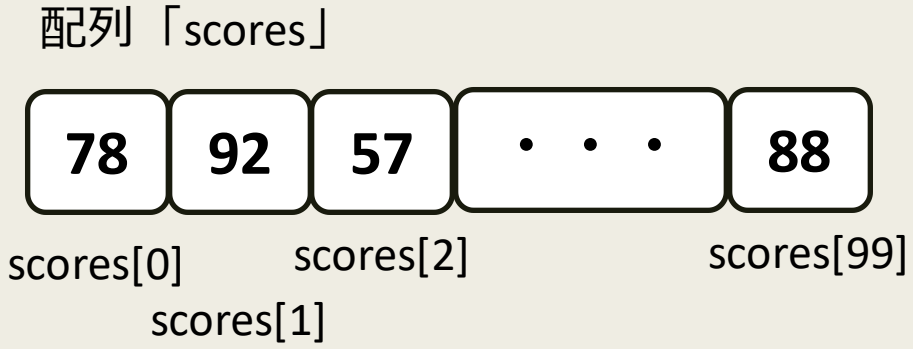
```
let score1 = 78;
let score2 = 92;
let score3 = 57;
  ⋮
let score100 = 88;
```

これまでの方法では
100個の変数を
宣言する必要がある



配列はひとつ宣言すれば
あとはまとめて管理できる

配列名と添字



- 配列に格納された個々の値を 要素 という。
- それぞれの要素には次の形式でアクセスする。

  配列名[添字]　（上の例では score[0] で 78 を取得）
- 上記のように配列名のあとに添字を大括弧（[ ]）で囲って記述する。
- 添字の番号は「0」から始まることに注意。
- 100個の要素からなる配列の最後の要素の添字は 99 である。

点数の合計を求める

- 配列は次の形式で宣言する。

  let 配列名 = [要素, 要素, ...]

- 配列に含まれる要素の個数は 配列名.length で取得できる。
- 次のコードを入力してみよう。

```
1   <!DOCTYPE html>
2   <html>
3
4   <head>
5       <meta charset="UTF-8">
6       <title>Prog_10-1</title>
7   </head>
8
9   <body>
10      <p>
11          <script>
12              let scores = [50, 70, 37, 90, 67];
13              let total = 0;
14              for(let i=0; i < scores.length; i++){
15                  total += scores[i];
16              }
17              document.write("合計は " + total + " 点です。");
18          </script>
19      </p>
20  </body>
21
22  </html>
```

12行目はサンプルコードからコピペできる

合計は 314 点です。

※配列を使わない場合、例えば100個の変数の和を計算するのは非常に面倒

[演習]点数の平均を求める

- 先のプログラムの修正し、配列の要素の平均を求めるプログラムを完成させよ。

```
1   <!DOCTYPE html>
2   <html>
3
4   <head>
5       <meta charset="UTF-8">
6       <title>Prog_10-2</title>
7   </head>
8
9   <body>
10      <p>
11          <script>
12              let scores = [50, 70, 37, 90, 67];
13
14
15
16
17
18
19          </script>
20      </p>
21  </body>
22
23  </html>
```

考えてみよう

平均は 62.8 点です。

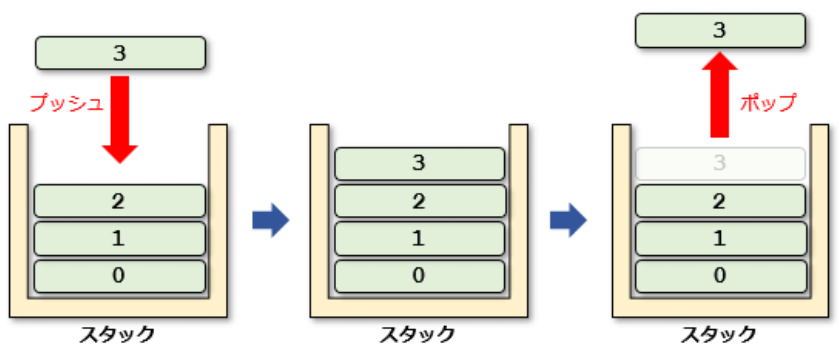
Array オブジェクト

- JavaScript で配列を作った場合、それは必ず Array オブジェクトとなる。
  - Array とは配列のこと。
- よって、Array オブジェクトのプロパティやメソッドが利用できる。
- さっき使った 配列名.length は Array オブジェクトのプロパティのひとつ。
- Array オブジェクトの主なメソッドは教科書の p193 を参照。
- Array オブジェクトのメソッドのうち、この授業では pop と push を紹介する。

空配列（からはいれつ）

- 配列を宣言するとき、要素や要素の個数がその時点では決まっておらず、とりあえず配列を用意だけしておきたい状況がある。
- そのようなときは要素をひとつももたない配列を宣言しておくのが便利である。
- 要素をひとつももたない配列を 空配列 という。
- push メソッドは配列の最後に新しい要素を追加する。
- pop メソッドは配列の最後の要素を削除し、その削除した要素を返す。

let array = [];
のように宣言する




```
let array = [0,1,2];
array.push(3);
console.log(array); // [0,1,2,3]
let x = array.pop();
console.log(array); // [0,1,2]
console.log(x); // 3
```

https://atmarkit.itmedia.co.jp/ait/articles/1908/06/news015.html

[演習] 5より大きい数を選ぶ

- 次のプログラムの空欄をうめ、与えられた配列の要素のうち、5より大きいものだけを残した配列を求めよう。

```
1   <!DOCTYPE html>
2   <html>
3
4   <head>
5       <meta charset="UTF-8">
6       <title>Prog_10-3</title>
7   </head>
8
9   <body>
10      <p>
11          <script>
12              let array = [6, 1, 10, -3, 5, 23, 0, -12, 8, 4];
13              let greaterThanFive = [];
14              for(l
15                  i
16
17                  }
18              }
19              document.write("5より大きい要素を残した配列は [" + greaterThanFive + "] です。");
20          </script>
21      </p>
22  </body>
23
24  </html>
```

12行目はサンプルコードからコピペできる

考えてみよう

5より大きい要素を残した配列は [6,10,23,8] です。

配列のコピー

- 次のコードを入力してみよう。
- どんな出力になるか予想しながら入力しよう。

```
11      <script>
12          let a = 3;
13          let b = a;
14          a = 5;
15          document.write("a の値は"+ a + "で、b の値は" + b + "です。<br>");
16          let A = [1,2,3];
17          let B = A;
18          A[0] = 10;
19          document.write("配列Aは [" + A + "] で、配列Bは [" + B + "] です。");
20      </script>
```

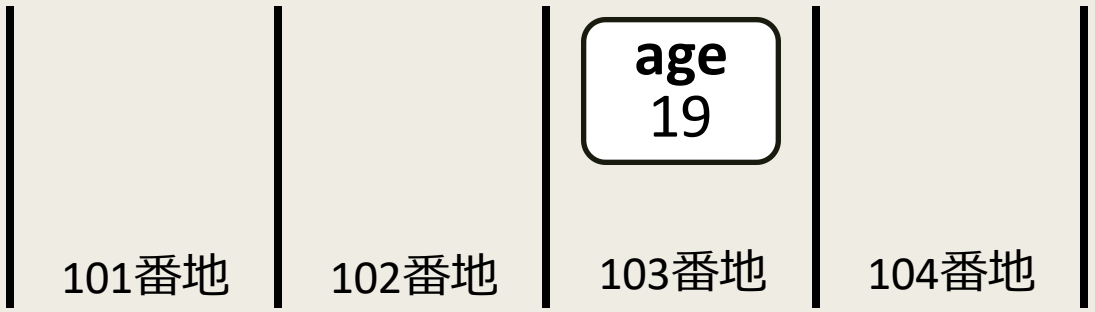
a の値は5で、b の値は3です。
配列Aは [10,2,3] で、配列Bは [10,2,3] です。

- 配列 B の出力結果は予想通りだろうか？
- なぜこのような出力になるのだろうか？

[復習]変数（第5回「変数と演算」より）

- プログラミング言語において最も重要な要素のひとつに、値を一時的に格納する変数がある。
- 変数は、値を入れる箱のようなもの。
- それぞれの箱には変数名と呼ばれる名前がある。
- アドレスと呼ばれる、箱を置いてある場所を表す数字の列もある（後の学習のために頭の片隅に入れておくとよい）。

例：年齢を管理する変数「age」を用意して 19 を代入




```
let age;
age = 19;
```

|| 同じ意味

```
let age = 19;
```

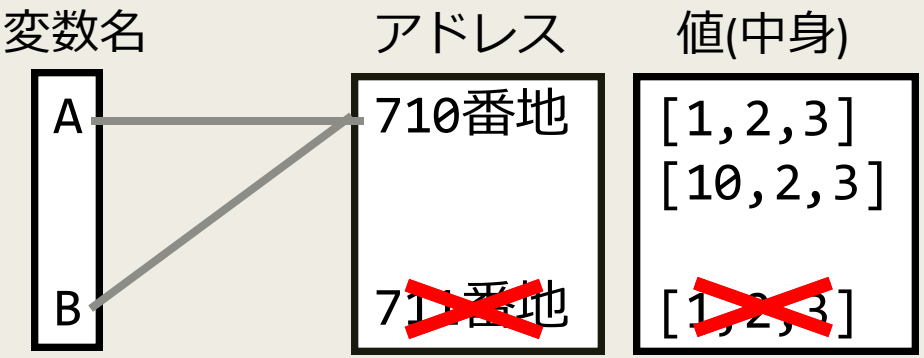
実際のアドレスは 0x7ffee10b2a4c のようにもっと複雑

値渡しと参照渡し

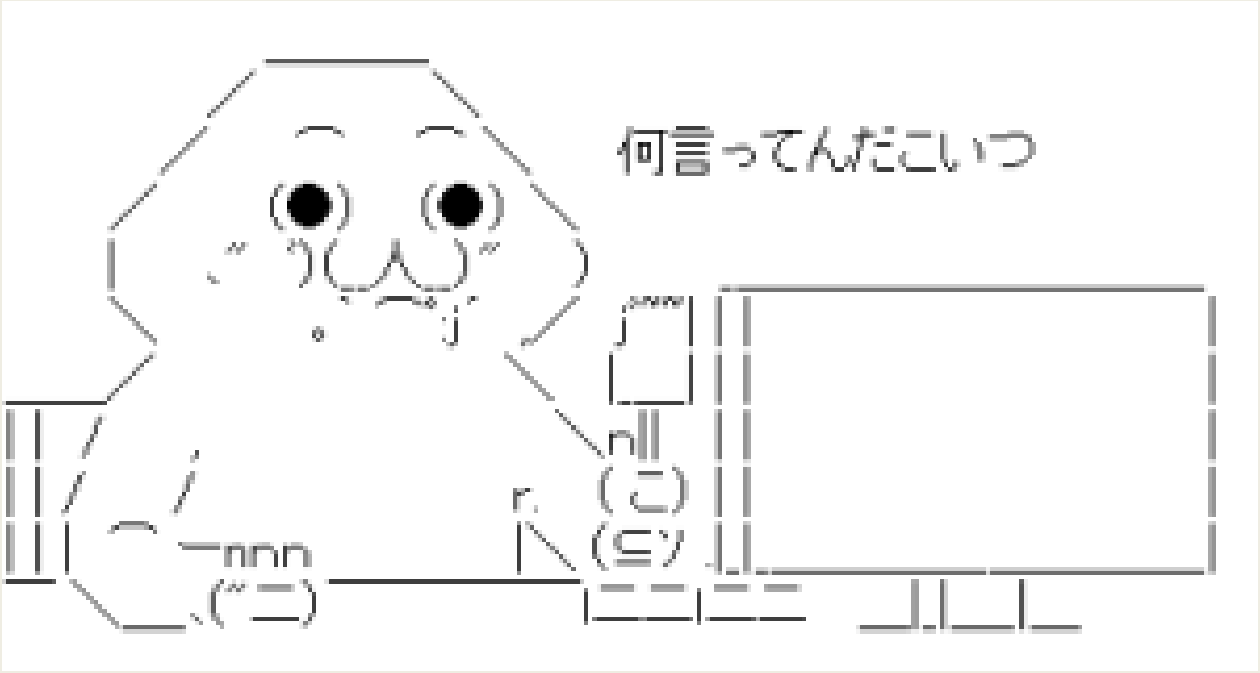
```
b = a;
```

- JavaScript に限らず多くの言語において、変数に変数を代入するときは次のように処理する。
- その変数が数値、文字列、真偽値の場合は変数の中にあるデータそのものをコピーする（人間の直感通り処理する）。
- これを 値渡し という。
- [発展]数値、文字列、真偽値のようにオブジェクトでないデータを プリミティブ型 という。
- しかし、変数に変数を代入するとき、その変数がオブジェクト（配列など）の場合は、その変数のアドレスをコピーする。
- これを 参照渡し という。

```
let A = [1,2,3];
let B = A;
let A[0] = 10;
```



よく分からなかった人向けの説明



- 配列をコピーするときは注意しましょう。

[演習]配列をコピーする

- それでは配列 A の中身を配列 B にコピーしたいときはどうすればよいだろうか？
- for文を使って配列Aのひとつひとつの値をコピー（push）すればよい。
- 出力が次になるように、以下のプログラムの空欄をうめよう。

配列Aは [10,2,3] で、配列Bは [1,2,3] です。

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <title>Prog_10-5</title>
</head>

<body>
    <p>
        <script>
            let A = [1,2,3];
            let B = [];

            考えてみよう

            A[0] = 10;
            document.write("配列Aは [" + A + "] で、配列Bは [" + B + "] です。");
        </script>
    </p>
</body>

</html>
```

[演習]教科書を熟読しよう

- 今日の内容は教科書の p176～p193 がベースになっている
- 残った時間で自分でも該当箇所を熟読してみよう。
- 配列の宣言の仕方が授業のやり方と少し違う（教科書ではオブジェクトであることを強調した方法をとっている）ので、その違いも踏まえながら読んでみよう。
- 授業で解説していないコードは自分でも入力してみてどのような出力結果になるか確かめてみよう。