

# プログラミング

## 第9回 繰り返し(2)

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。



## 久保田匠の授業関連のページ

2024年度前期担当科目

	月曜	火曜	水曜	木曜	金曜
1限					
2限	<a href="#">確率統計I</a>	<a href="#">確率統計I</a>			
3限				<a href="#">線形数学演習I</a>	<a href="#">確率統計II</a>
4限	(オフィスアワー)				
5限					

2024年度後期担当科目

	月曜	火曜	水曜	木曜	金曜
1限					
2限					
3限	<a href="#">科学リテラシー</a>				<a href="#">プログラミング</a>
4限	(オフィスアワー)	3年ゼミ			<a href="#">プログラミング</a>
5限					

## 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	●	<a href="#">Prog_01-1</a>
第2回	Webページを構築する(HTML)	●	<a href="#">Prog_02-1</a>
第3回	Webページの見栄えを整える(CSS)	●	<a href="#">Prog_03-1</a> <a href="#">Prog_03-2</a>
第4回	JavaScriptに触れてみよう	●	<a href="#">Prog_04-1</a>
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)	●, ★	<a href="#">Prog_08-1</a>
第9回	繰り返し(2)	●, ★	
第10回	オブジェクト	●, ★	
第11回	配列	●, ★	
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	●	

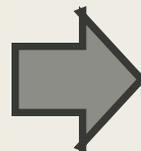
# [準備]コードの新規作成①

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。

## 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	●	<a href="#">Prog_01-1</a>
第2回	Webページを構築する(HTML)	●	<a href="#">Prog_02-1</a>
第3回	Webページの見栄えを整える(CSS)	●	<a href="#">Prog_03-1</a> <a href="#">Prog_03-2</a>
第4回	JavaScriptに触れてみよう	●	<a href="#">Prog_04-1</a>
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)	●, ★	<a href="#">Prog_08-1</a>
第9回	繰り返し(2)	●, ★	
第10回	オブジェクト	●, ★	
第11回	配列	●, ★	
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	●	



## Prog\_08-1

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Prog_08-1</title>
  <!-- しばらくここは使いません。 -->
</head>

<body>
  <!--
  このようにコメント機能を有効活用すると
  授業で入力したコードを一時的に無効にできます。
  <script>
    let year = prompt("今は西暦何年?");
    document.write("今は令和" + (year - 2018) + "年です。");
  </script>
  -->
</body>
```

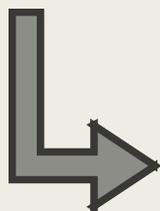
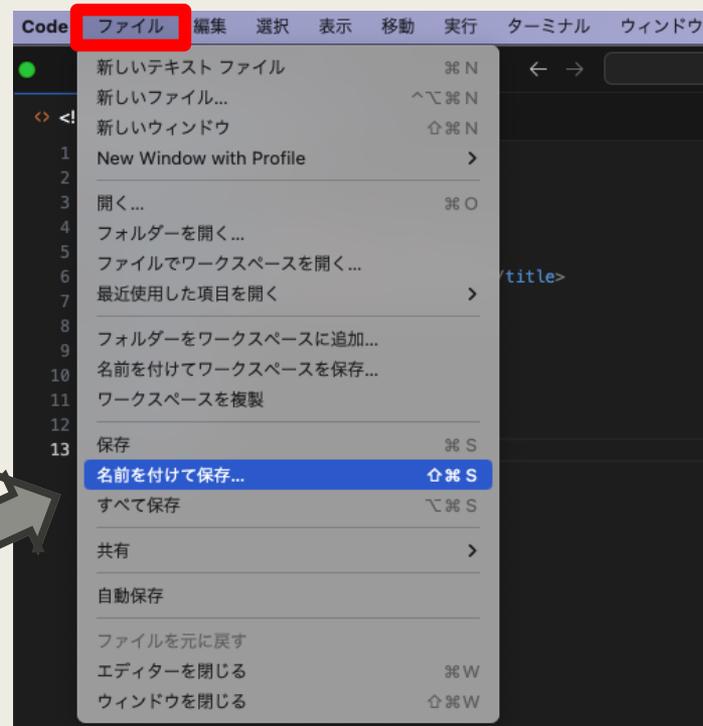
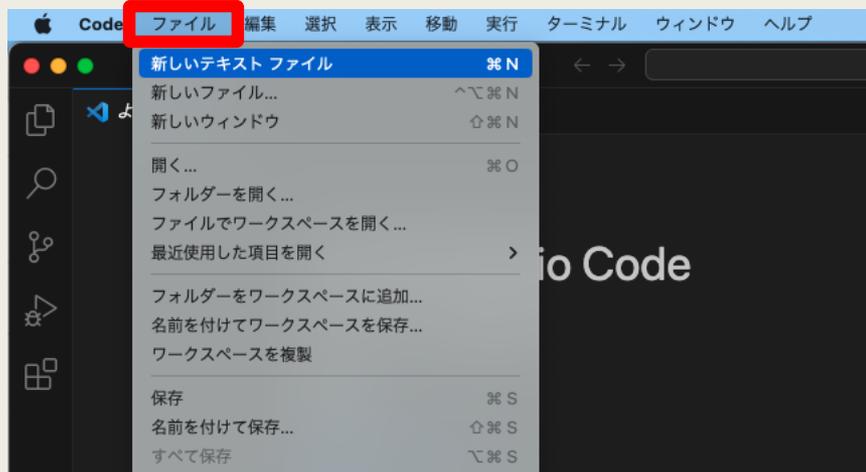
コピー

今日も「Prog\_08-1」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

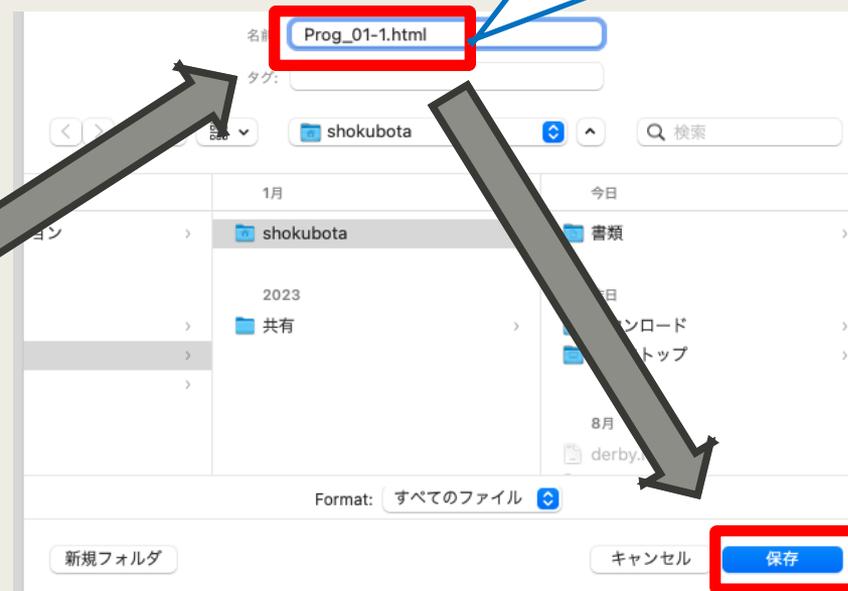
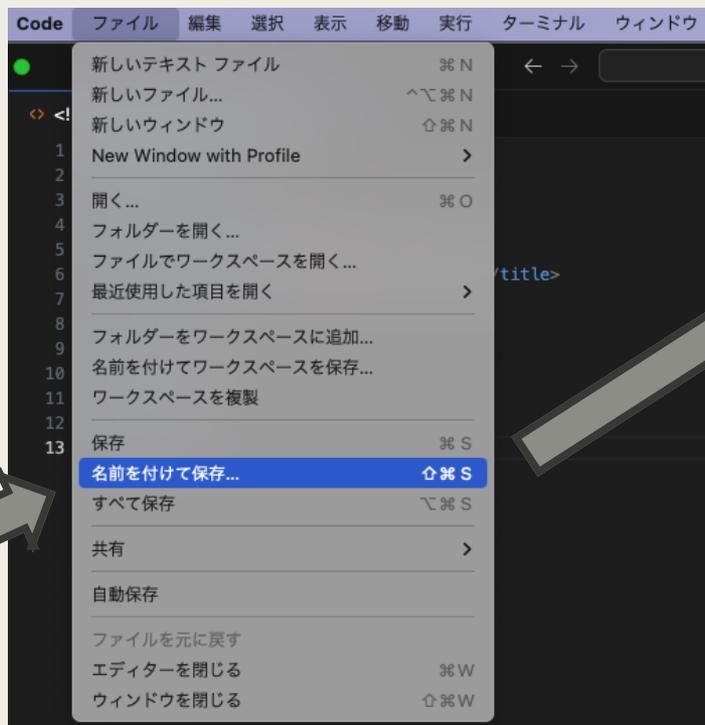
```
ようこそ <!DOCTYPE html> Untitled-1
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>ここにページのタイトルを書きます</title>
7 </head>
8
9 <body>
10   こんにちは！
11 </body>
12
13 </html>
```

# [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」。

今日は  
「Prog\_09-1.html」  
とつける。



# [準備]作業環境を整える

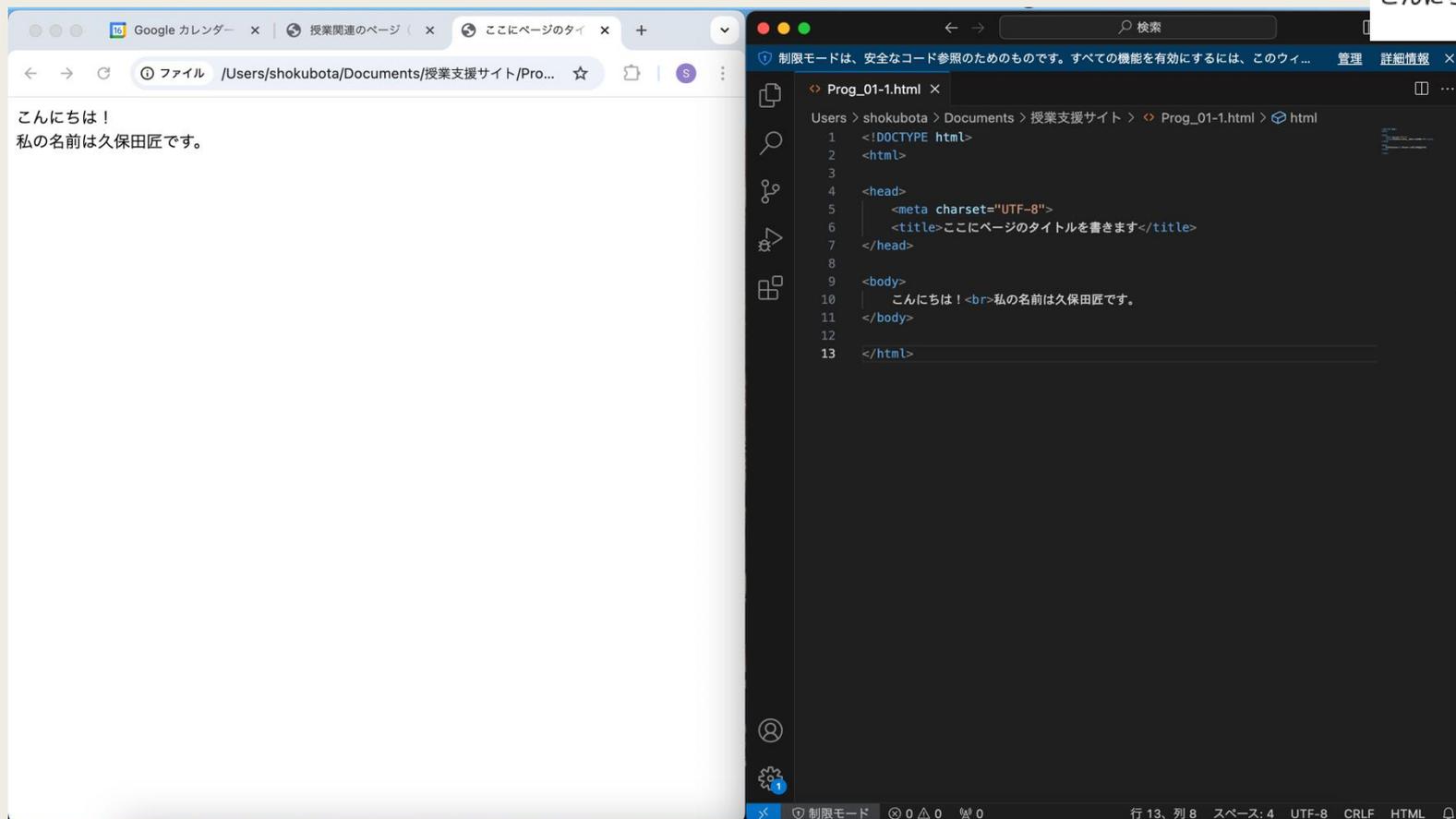
いつもの作業

- 保存したhtmlファイルをダブルク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。

ダブル  
クリック

← → ↻ ⓘ ファイル

こんにちは！



# [復習]指定した回数だけ処理を繰り返す

- if文のような条件判断に並び、プログラムに欠かせない概念に**ループ（繰り返し）**がある。
- ループを記述する方法は主に **for文** と **while文** の2種類がある。
- for文は繰り返し回数が分かっている場合に使い、while文は繰り返し回数が定まっていない場合（特定の条件を満たすまで繰り返すなど）に用いる。
- for文の基本的な書式は以下の通り。**今日中に慣れてくてください。**頑張ろう。

## 基本的な書式

```
for(初期化式; 条件式; 制御変数の更新){  
    ステートメント  
    ステートメント  
    ステートメント  
}
```

## 具体例（“やきにく”と10回言うプログラム）

```
for(let i = 1; i <= 10; i++){  
    document.write(“やきにく”);  
}
```

i++ は変数 i の値を 1 増やす  
という意味。i = i+1 と同じ。

# [復習] 「やきにく」と10回言うプログラム

- 前回、次のコード（「やきにく」と10回言うプログラム）を入力してもらった。

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Prog_07-3</title>
7 </head>
8
9 <body>
10  <p>
11    <script>
12      for(let i=1; i<=10; i++){
13        document.write("やきにく");
14      }
15    </script>
16  </p>
17 </body>
18
19 </html>
```

- $i$  は 1 から 10 まで動き、処理ごとに  $i$  は 1 ずつ増える。
- $i = 1$  のときの処理
  - 「やきにく」と表示する。
- $i = 2$  のときの処理
  - 「やきにく」と表示する。
- 同じ処理を  $i = 10$  になるまで処理する。
- $i = 10$  のときに処理が終わったら 15 行目に進む。

やきにくやきにくやきにくやきにくやきにくやきにくやきにくやきにくやきにく

# [復習] 1 から 100 までの和を計算する

- 前回、次のコード（1 から 100 までの和を計算するプログラム）も入力してもらった

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Prog_07-4</title>
7 </head>
8
9 <body>
10  <p>
11    <script>
12      let sum = 0;
13      for(let i=1; i<=100; i++){
14        sum = sum + i;
15      }
16      document.write("1から100までの和は" + sum + "です。");
17    </script>
18  </p>
19 </body>
20
21 </html>
```

1から100までの和は5050です。

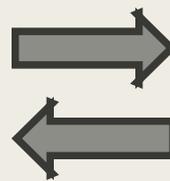
14行目は `sum += i;`  
と書いてもよい

- `i` は 1 から 100 まで動き、  
処理ごとに `i` は1ずつ増える。
- `i = 1` のとき
  - 変数 `sum` に 1 を加える  
(この時点で `sum = 1`)。
- `i = 2` のとき
  - 変数 `sum` に 2 を加える  
(この時点で `sum = 3`)。
- `i = 3` のとき
  - 変数 `sum` に 3 を加える  
(この時点で `sum = 6`)。

# [復習] 条件が成立している間、処理を繰り返す

- 次は while 文。
- for 文では for の直後のカッコに「初期化式」「条件式」「制御変数の更新」の3つを記述した。
- while 文では while の直後のカッコには「条件式」のみを記述する。
- その条件式が満たされる間(while)、処理を繰り返す。
- for 文 は while 文 に書き換えることができる（実は逆も真）。

```
for(let i = 1; i <= 10; i++){  
    document.write("やきにく");  
}
```



互いに  
書き換え可能

```
let i=1;  
while(i <= 10){  
    document.write("やきにく");  
    i++;  
}
```

*i <= 10 が真である限り  
「やきにく」と言い続ける*

# [復習]繰り返しを中断する (break文)

- 繰り返しの処理を記述する際、何らかの条件が成立した時点でループを中断したいといったケースがある。
  - 方程式の解をひとつでも見つけられれば良い状況など。
- そのような場合には **break文** を使う。
- プログラムは `break;` に遭遇した時点でループから抜ける。

```
while true {  
    処理  
  
    if(条件式){  
        break;  
    }  
  
    処理  
}
```

無限ループ前提で条件式に  
`true` を置くことがある。

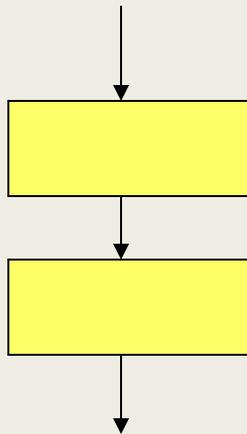
この条件式が満たされれば  
`while` のループから脱却する。

# 構造化定理

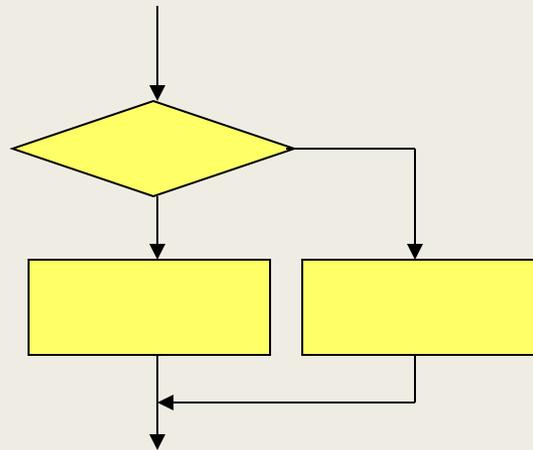
初心者が学ぶプログラム  
はだいたい当てはまる

- 「単純な逐次的フローで、ひとつの入力からひとつの出力を得るような処理を持つプログラムは以下の3つの基本構造の組み合わせで記述できる」という定理。
  - ① 順次：上から下に順番に処理すること。
  - ② 選択：条件によって処理を変えること(if文)。
  - ③ 反復：同じ処理を繰り返すこと(for文やwhile文)。

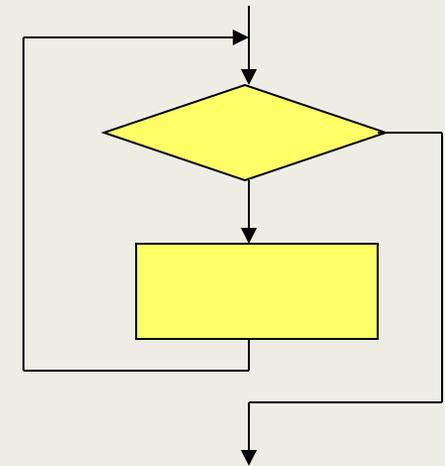
順次



選択

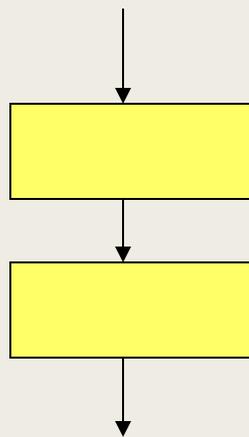


反復

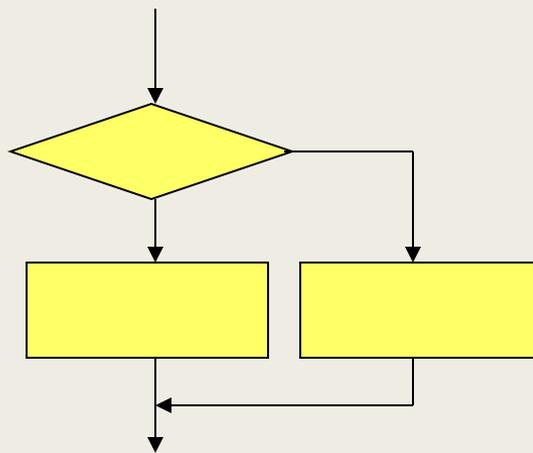


# 構造化定理

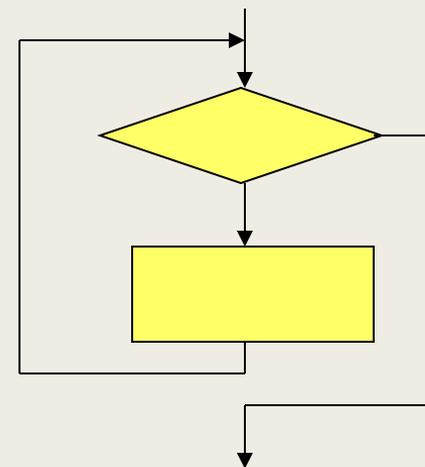
順次



選択



反復



- プログラミング言語によって得意なこと・苦手なことは違うが **if文 や for文 を学んだ皆さんは原理的には既に非常に多くのプログラムが書ける状態**になっている。
- 「原理的には」というのは、「『歩く』と『泳ぐ』が出来れば地球上のどの場所にも行ける」のような意味。
- 実際には、上記の基本構造の他に、データの管理や操作、ユーザーの行動に応じて動作する仕組みを学ぶ必要がある。

# 第10回以降に学ぶこと

逆行列

ボタンをクリックすると  
答えが表示される

逆行列

[第14回]  
きれいな数式を  
表示する

行列  $\begin{bmatrix} -5 & 0 & 1 \\ -4 & -1 & 1 \\ -2 & -2 & 1 \end{bmatrix}$  の逆行列は

である。

答え

行列  $\begin{bmatrix} -5 & 0 & 1 \\ -4 & -1 & 1 \\ -2 & -2 & 1 \end{bmatrix}$  の逆行列は  $\begin{bmatrix} 1 & -2 & 1 \\ 2 & -3 & 1 \\ 6 & -10 & 5 \end{bmatrix}$  である。

[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。

# [演習] FizzBuzz

- 「Fizz Buzz」は英語圏で長距離ドライブや飲み会のおきに行われる言葉遊び。
- プレイヤーは「1」から順に数字を発言していく。
- ただし、3の倍数のおきは「Fizz」、5の倍数のおきは「Buzz」、両方のおきは「FizzBuzz」と発言する。
  - 1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11, Fizz, 13, 14, FizzBuzz, 16, 17, ...
- 1から100までの数について、3の倍数なら「Fizz」、5の倍数なら「Buzz」、両方のおきは「FizzBuzz」と出力するプログラムをかけ。次の出力を参考にせよ。

```
1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11, Fizz, 13, 14, FizzBuzz, 16, 17, Fizz, 19, Buzz, Fizz, 22, 23, Fizz, Buzz, 26, Fizz, 28, 29, FizzBuzz, 31, 32, Fizz, 34, Buzz, Fizz, 37, 38, Fizz, Buzz, 41, Fizz, 43, 44, FizzBuzz, 46, 47, Fizz, 49, Buzz, Fizz, 52, 53, Fizz, Buzz, 56, Fizz, 58, 59, FizzBuzz, 61, 62, Fizz, 64, Buzz, Fizz, 67, 68, Fizz, Buzz, 71, Fizz, 73, 74, FizzBuzz, 76, 77, Fizz, 79, Buzz, Fizz, 82, 83, Fizz, Buzz, 86, Fizz, 88, 89, FizzBuzz, 91, 92, Fizz, 94, Buzz, Fizz, 97, 98, Fizz, Buzz,
```

# [演習] かけ算九九の表 (二重for文)

- 二重 for 文 にも挑戦してみよう。
- 次の出力結果を参考に、かけ算九九の表を作ってみよう。
- document.write 命令の引数に表に関するタグ (<tr>, <td>) を入れることで表そのものも for 文を使って構築できる。



HTMLにおける表のかき方は  
第2回の授業で解説しているので  
適宜復習しよう。

かけ算九九の表 (中身のみ)

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

# [演習] 枠付きかけ算九九の表 (二重for文 + if文)

- 二重 for 文 と if文 のプログラムにも挑戦してみよう。
- 少し難しいと思うが、二重 for 文 と if文 が混在したコードがかけるとプログラムの幅がぐんと広がる。
- 次の出力結果を参考に、枠付きのかけ算九九の表を作ってみよう。
- 枠なしのかけ算九九の表のプログラムも参考にしてみよう。



HTMLにおける表のかき方は  
第2回の授業で解説しているので  
適宜復習しよう。

かけ算九九の表 (枠付き)

×	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

# [演習] 二項係数を求める

$${}^n C_k = \frac{{}^n P_k}{k!}$$

- ${}^n C_k$  のことを **二項係数** という。
- 二項定理の係数に現れる数のため。
- 次の出力例を参考にして、二項係数を計算するプログラムをかけ。ユーザーから入力される  $n$  と  $k$  は自然数であることを想定してよい。

このページの内容

${}^n C_k$  の  $n$  の値を指定してください。

キャンセル OK

このページの内容

${}^n C_k$  の  $n$  の値を指定してください。

キャンセル OK

このページの内容

${}^n C_k$  の  $k$  の値を指定してください。

キャンセル OK

このページの内容

${}^n C_k$  の  $k$  の値を指定してください。

キャンセル OK

$5C2$  の値は 10 です。

$10C5$  の値は 252 です。

# [演習] 2桁の整数の桁の和を求める

- 次の出力例を参考にして、ユーザーから入力される2桁の整数の桁の和を表示するプログラムをかけ。
- この問題は for 文も while文 も if文 も使わない。

このページの内容

2桁の整数 n の値を指定してください。桁の和を計算します。

キャンセル OK

桁の和は 7 です。

このページの内容

2桁の整数 n の値を指定してください。桁の和を計算します。

キャンセル OK

桁の和は 13 です。

# [演習] 正の整数の桁の和を求める

- 先の問題を一般化する。
- 次の出力例を参考にして、ユーザーから入力される正の整数の桁の和を表示するプログラムをかけ。

このページの内容

正の整数  $n$  の値を指定してください。桁の和を計算します。

キャンセル OK

桁の和は 15 です。

このページの内容

正の整数  $n$  の値を指定してください。桁の和を計算します。

キャンセル OK

桁の和は 39 です。