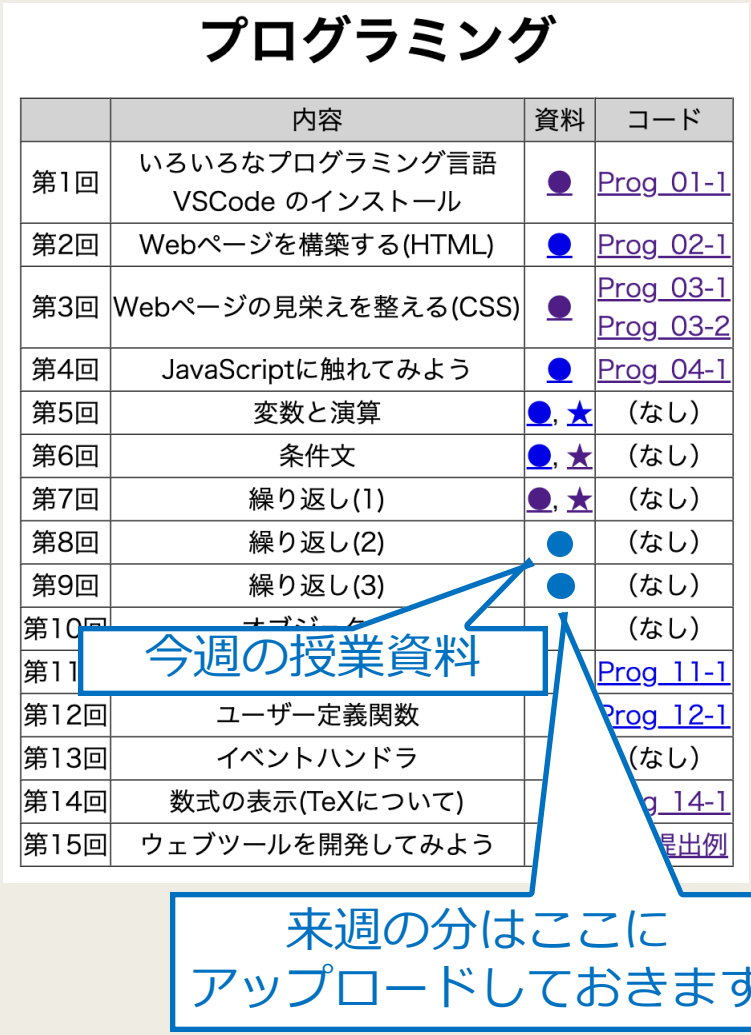
プログラミング

第8回
繰り返し(2)

久保田 匠

[連絡]来週はオンデマンド

- 来週（12月19日）は久保田出張のためオンデマンド講義。
- いつもの授業サイトに資料をアップロードしておくので取り組んでください。
- 主に、for, while, if の組み合わせの問題演習です。
- 来週までにはこれらに慣れておいてください。
- 課題の提出は特にありません。



[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	繰り返し(1)	● ★	（なし）
第8回	繰り返し(2)	●	（なし）
第9回	繰り返し(3)		（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		Prog_12-1
第13回	イベントハンドラ		（なし）
第14回	数式の表示(TeXについて)		Prog_14-1
第15回	ウェブツールを開発してみよう		課題提出例

[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	●	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	繰り返し(1)	●, ★	(なし)
第8回	繰り返し(2)	●	(なし)
第9回	繰り返し(3)		(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		Prog_12-1
第13回	イベントハンドラ		(なし)
第14回	数式の表示(TeXについて)		Prog_14-1
第15回	ウェブツールを開発してみよう		課題提出例

Prog_04-1

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <title>Prog_04-1</title>
    <!-- 今日はここは使いません。 -->
</head>

<body>
    <!-- ここに今日の授業内容を入力します。 -->
</body>

</html>
```

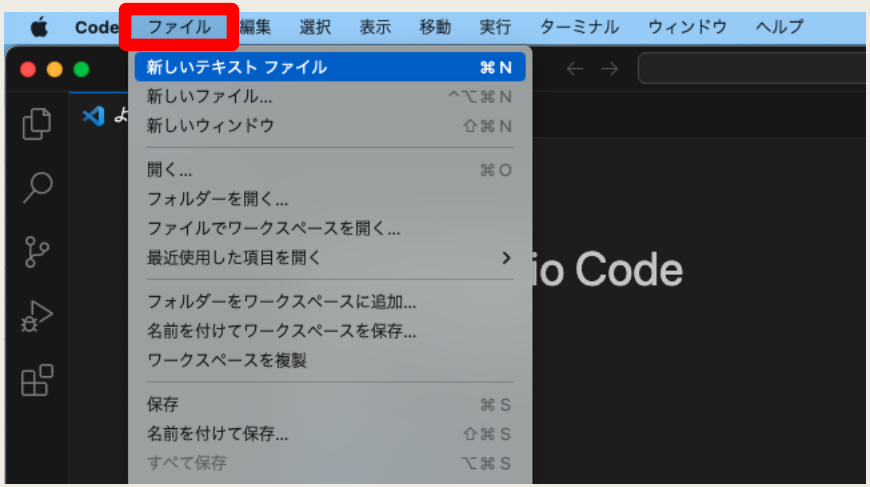
コピー

今日も「Prog_04-1」を選択してください。

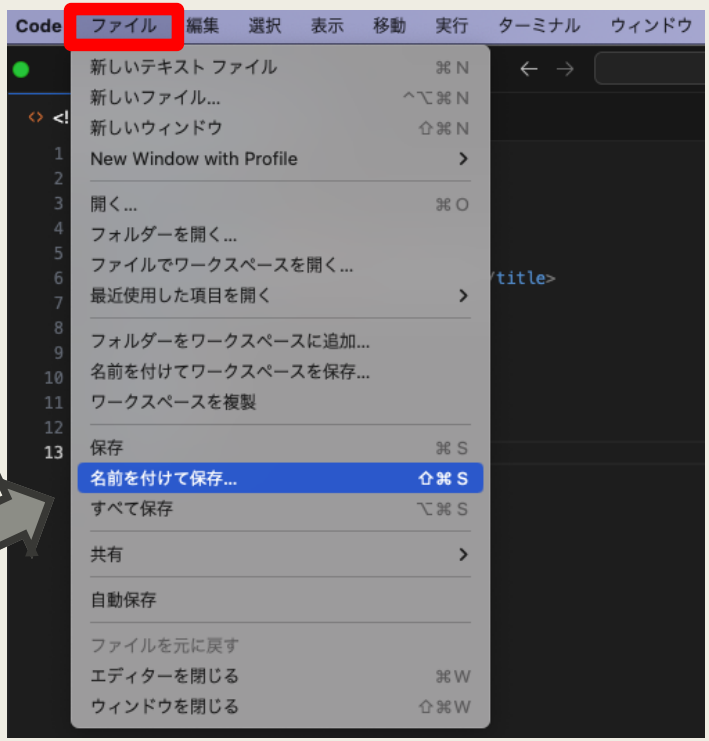
[準備]コードの新規作成②

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」。



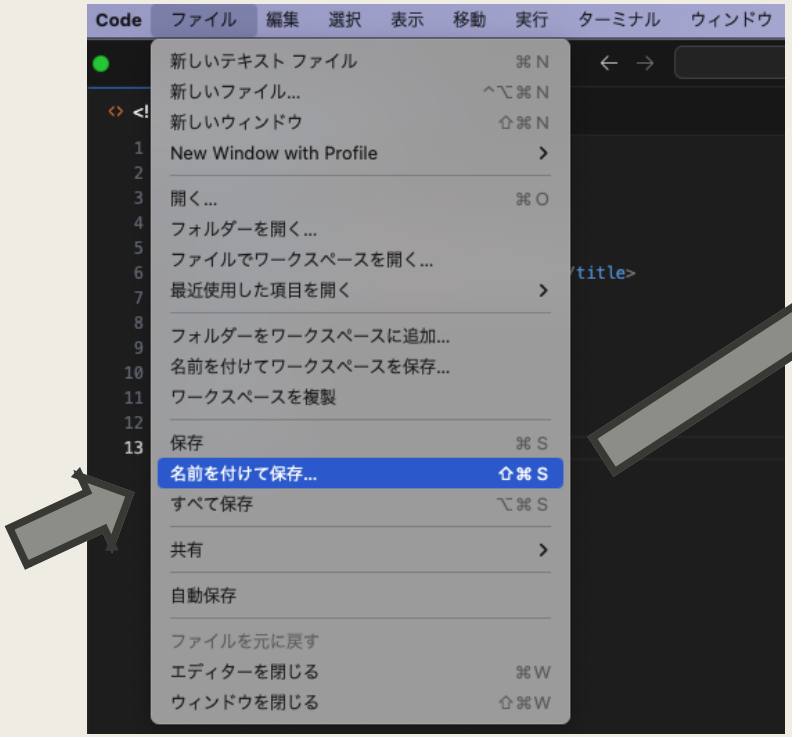
Ctrl + V



[準備]コードの新規作成②

いつもの作業

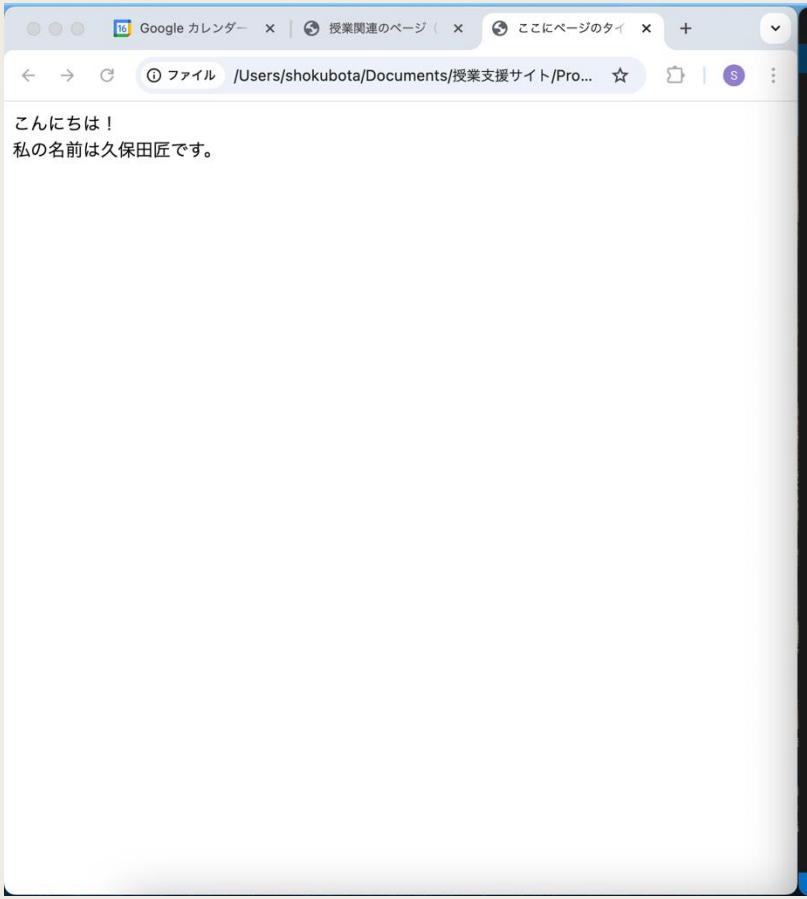
- VSCode を起動し「ファイル」から「新しいテキストファイル」を選択。
- そのあと、さきほどコピーした文書をペースト（Ctrl + V）して「名前をつけて保存」。

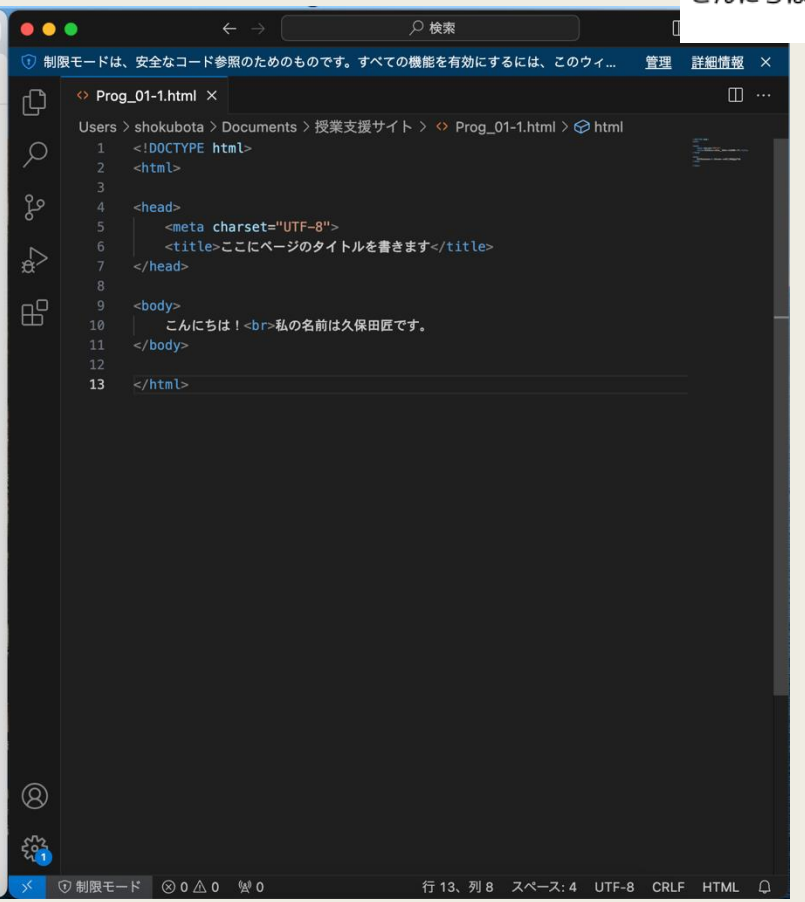


[準備]作業環境を整える

いつもの作業

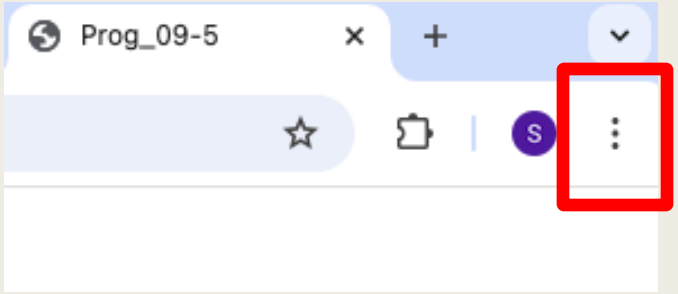
- 保存したhtmlファイルをダブルリックして開いておく。
- PCの画面をふたつに分け、片方はブラウザ、もう片方は VSCode を開いておくと便利。

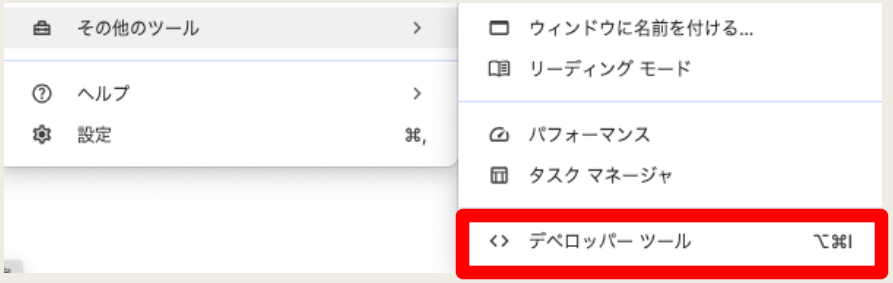


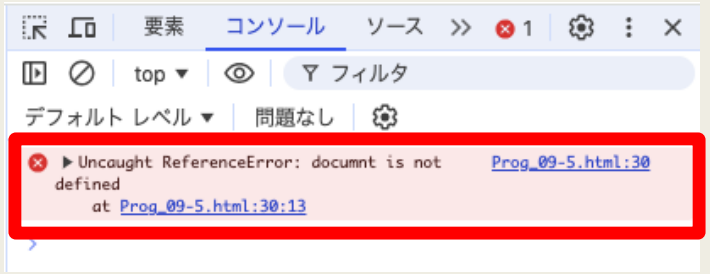


[再掲]デベロッパーツール

- 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]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。**なるべく今日中に慣れてください。**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 や i += 1 と同じ。

[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）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_07-4</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。

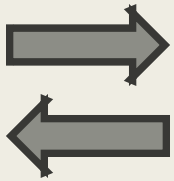
14行目は sum += i;
と書いてもよい

- 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
- i = 4 のとき…
- i が 100 まで繰り返す。

[復習]条件が成立している間、処理を繰り返す

- 前回は while文 も扱った。
- 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。
- 以下は、break 文を使うコードの典型例。

```
while(true){

    処理

    if(条件式){
        break;
    }

    処理

}
```

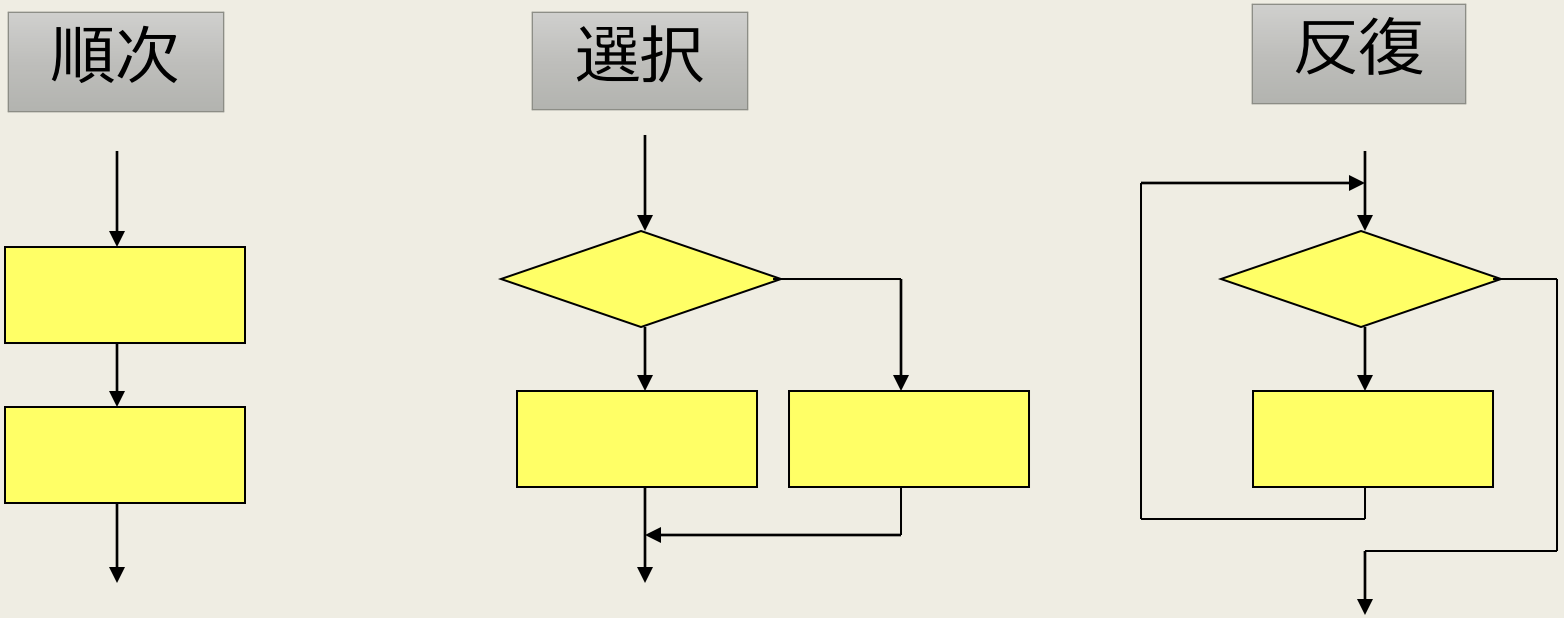
無限ループ前提で条件式に
true を置くことがある。

この条件式が満たされれば
while のループから脱却する。

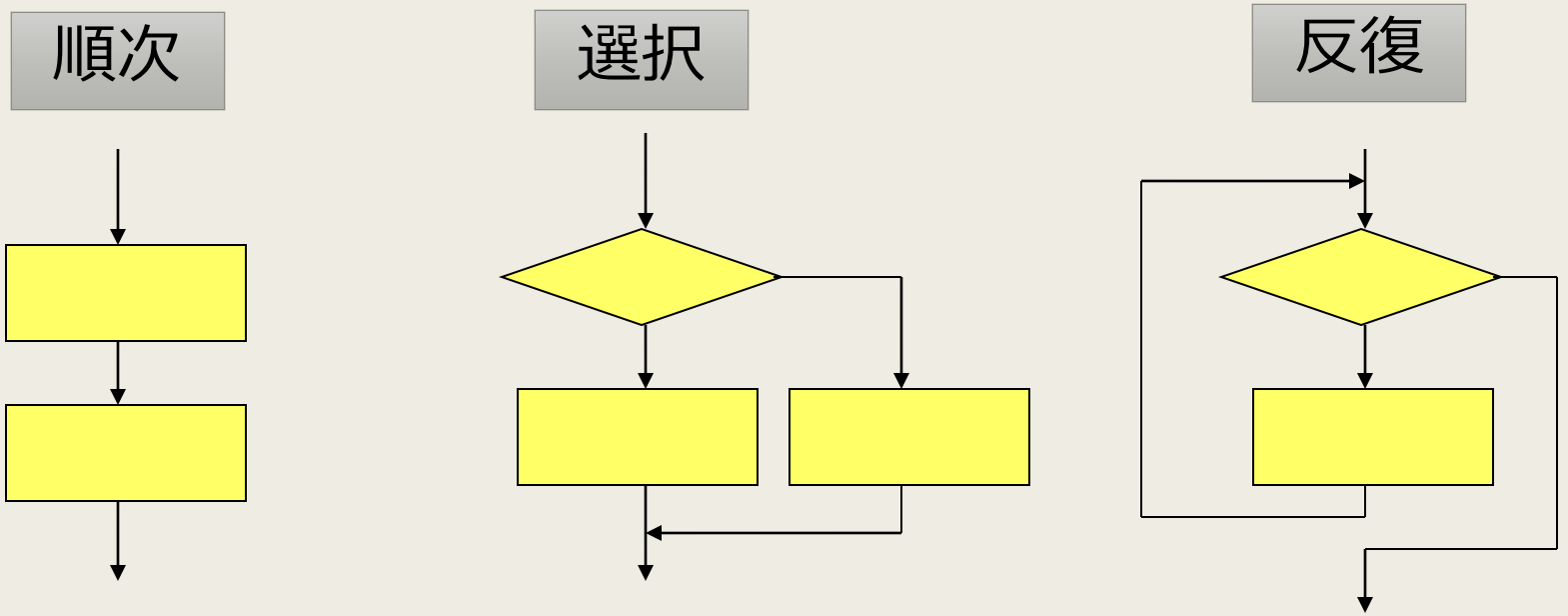
構造化定理

数学の具体的な計算問題はだいたいあてはまる

- 「入力を受け取り出力を返す処理は以下の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)。



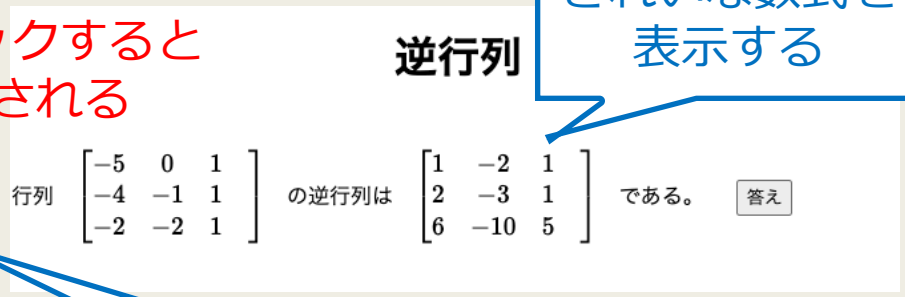
構造化定理



- if文 や for文 を学んだ皆さんは、実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。
- 上記の基本構造の他には、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

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

[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と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の倍数のときは「Fizz」、5の倍数のときは「Buzz」、両方のときは「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」、両方のときは「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.wrtie 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

解答例

```
 9    <body>
10        <p>
11            かけ算九九の表（中身のみ）
12        </p>
13        <table border="1">
14            <script>
15                for(let i = 1; i <= 9; i++){
16                    document.write("<tr>");
17                        for(
18
19                        }
20                    document.write("</tr>");
21                }
22            </script>
23        </table>
24    </body>
```

考えてみよう

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

- 変数 i は行の番号、変数 j は列の番号を表す。
- プログラムの16行目と20行目で、各 i に対して<tr>タグを入力。
- 18行目で i*j の計算結果を <td>タグで挟んでいる。

[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）

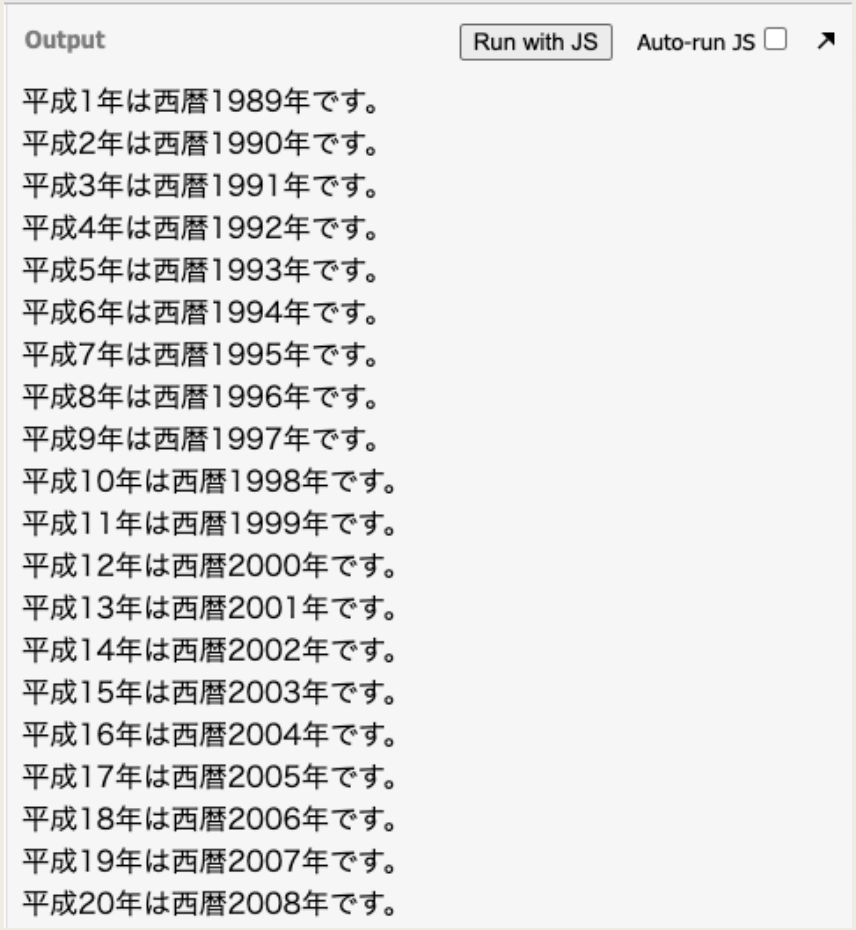
×	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

既に終わった人 or 既に諦めた人

- 次以降のスライドにまだまだ問題を用意しています。
- 今日の演習問題を既に終えている人は、順に取り組んでください。
- 今日の演習問題が難しくて諦めた人は、解けそうな問題から取り組んでみましょう。
- これらは来週各自で取り組んでもらう内容にもなります。

[演習]平成を西暦に変換する

- 平成元年は西暦 1989 年であり、平成最後の年は31年で西暦 2019 年である。
- 出力が次になるように、平成を西暦に変換して表示するプログラムを作成しよう。



[演習]奇数の和

今年度新たに用意した問題→ NEW

- 1 から 99 までの奇数の総和を、for 文を使って求めよう。
- 余裕のある人は if文を使わないコードと、if文を使って求めるコードを両方書いてみよう。

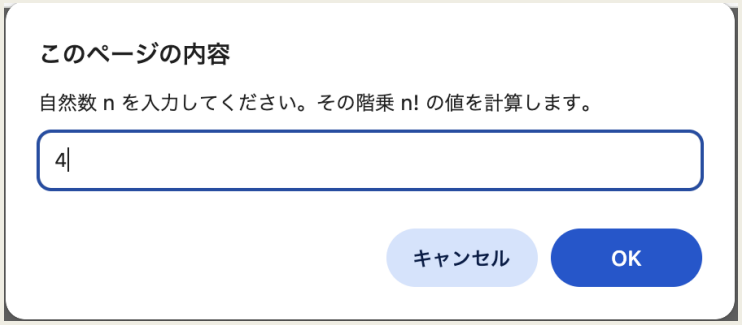
```
1から99までの奇数和は 2500 です。
```

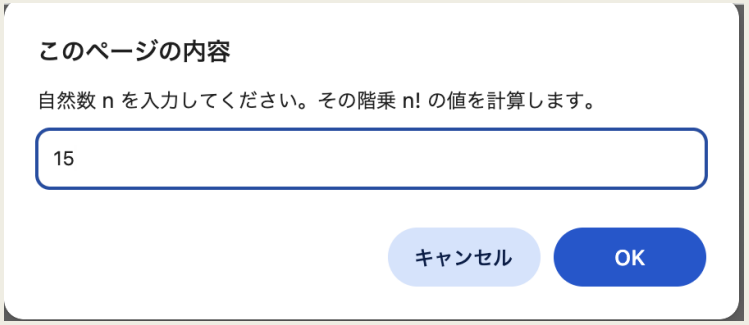
ここをいじってみよう

```
for(let i = 1; i <= 99; i++){
    ステートメント
    ステートメント
      ⋮
}
```

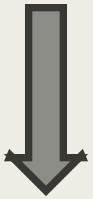
[演習]階乗の計算

- ユーザーから入力された自然数 n に対して、nの階乗 n! 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 は自然数であることを想定してよい。





4の階乗は 24 です。



15の階乗は 1307674368000 です。

[演習]漸化式の最初の項を列挙する

- 次の漸化式で定義される数列 $\{a_n\}$ の最初の10項を列挙せよ。

$$a_1 = 1, \quad a_{n+1} = 2a_n + n$$

第1項は 1 です。
第2項は 3 です。
第3項は 8 です。
第4項は 19 です。
第5項は 42 です。
第6項は 89 です。
第7項は 184 です。
第8項は 375 です。
第9項は 758 です。
第10項は 1525 です。

[演習] 二項係数を求める

$$ {}_n\mathrm{C}_k = \frac{{}_n\mathrm{P}_k}{k!} $$

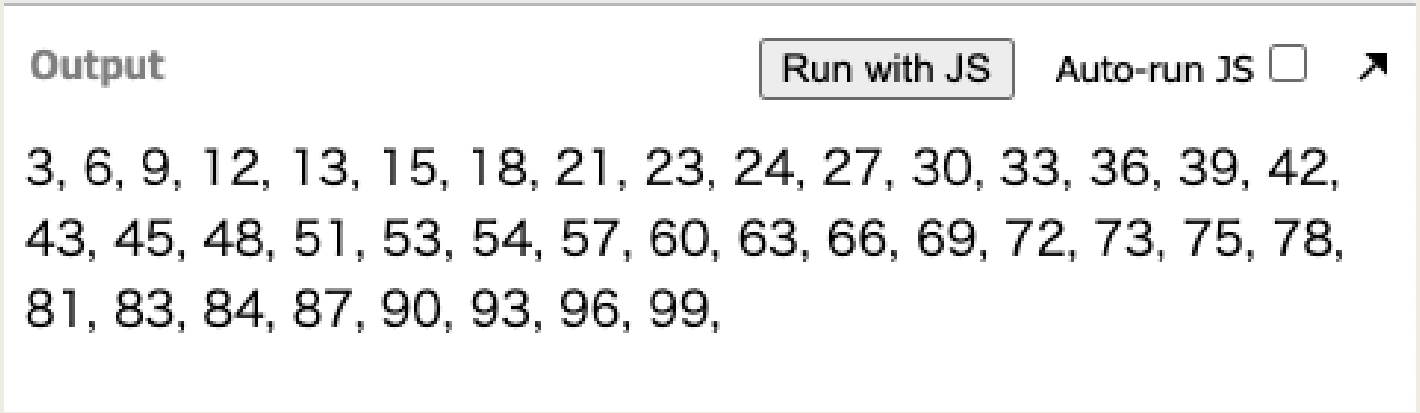
- ${}_n\mathrm{C}_k$ のことを 二項係数 という。
- 二項定理の係数に現れる数のため。
- 次の出力例を参考にして、二項係数を計算するプログラムをかけ。ユーザーから入力される n と k は自然数であることを想定してよい。

5C2 の値は 10 です。

10C5 の値は 252 です。

[演習]特定の数値を表示する

- 出力が次になるように、1以上100以下の整数のうち「3の倍数または下一桁が3の整数」を列挙するプログラムを作成しよう。

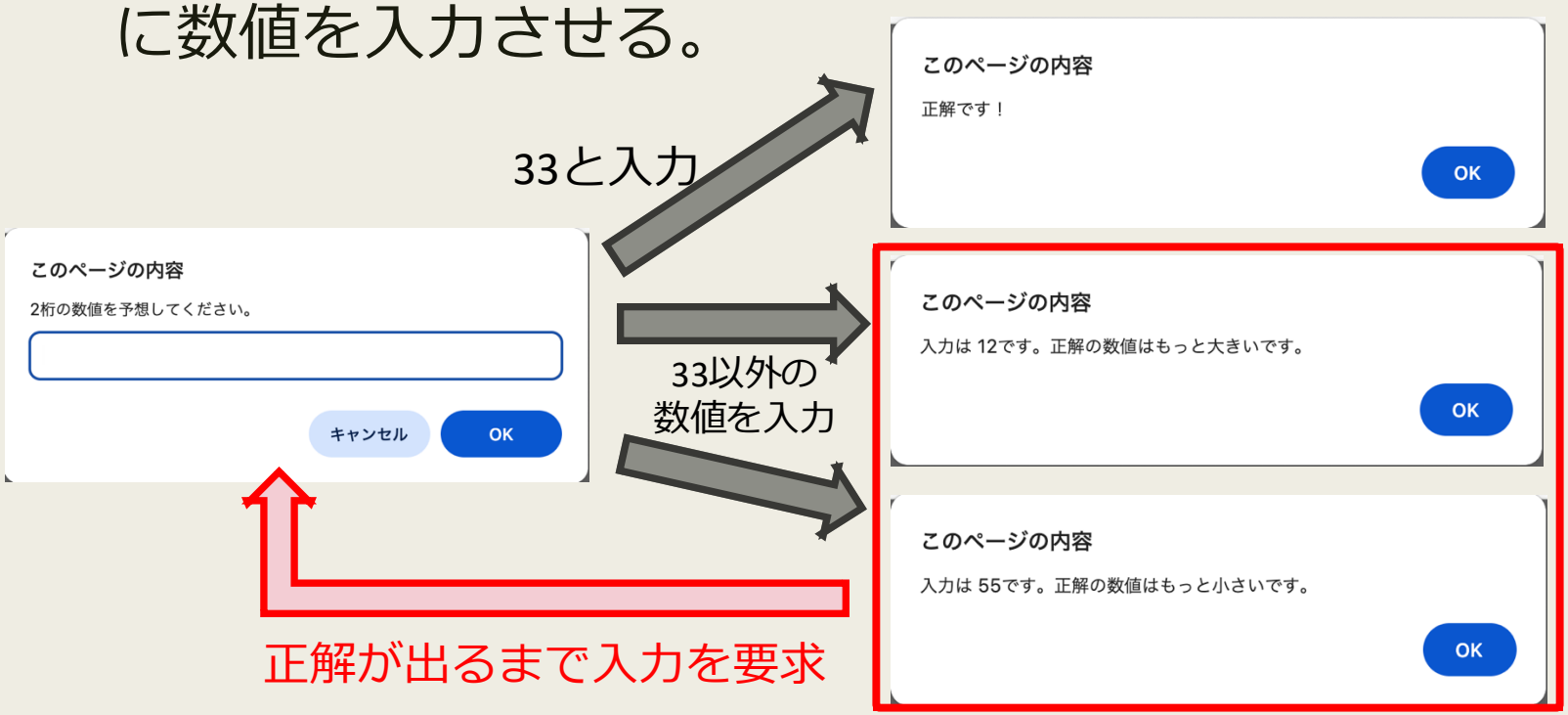


[演習]数当てゲーム

```
let correctNumber = 33;
//let correctNumber = Math.floor(Math.random()*89 + 10);
// ↑ 正解をランダムにしたい場合
```

NEW

- 2桁の数を当てるゲーム。
- まずは簡単のために答えの数値を「33」に設定。
- 下図のようにユーザーに2桁の数値を入力してもらい…
  - 入力が正しい数値なら「正解です！」と表示。
  - 入力が正しくない場合は、答えの数値が入力よりも大きいか小さいかを知らせ、正解が入力されるまでユーザーに数値を入力させる。

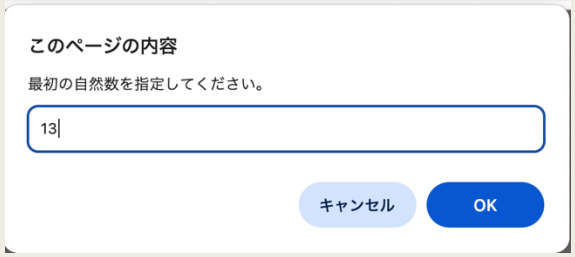


[演習] コラッツの問題

- 自然数をひとつ選び、次の操作を繰り返す。
  - その数が偶数ならば *2* で割る。
  - その数が奇数ならば *3* をかけて *1* を足す。
- 例えば、選んだ自然数が 3 なら、次を経て最終的に 1 に到達する。

$$3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1 \qquad (\bigstar)$$

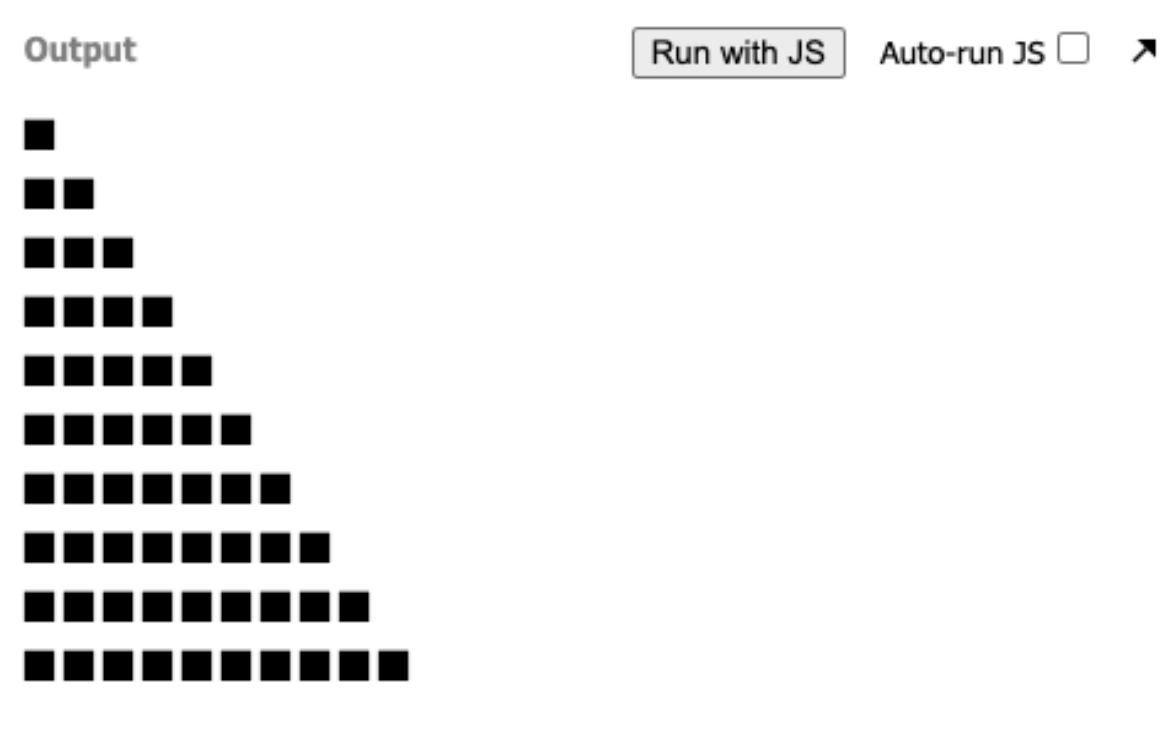
- 最初の数がどんな自然数であっても、この操作を繰り返すと必ず 1 に到達するだろうか？
- この問題を コラッツの問題 といい、人類はいまだに解くことができていない。
- ユーザーから入力された自然数 n に対して、 (★) のような列を表示するプログラムをかけ。



13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1

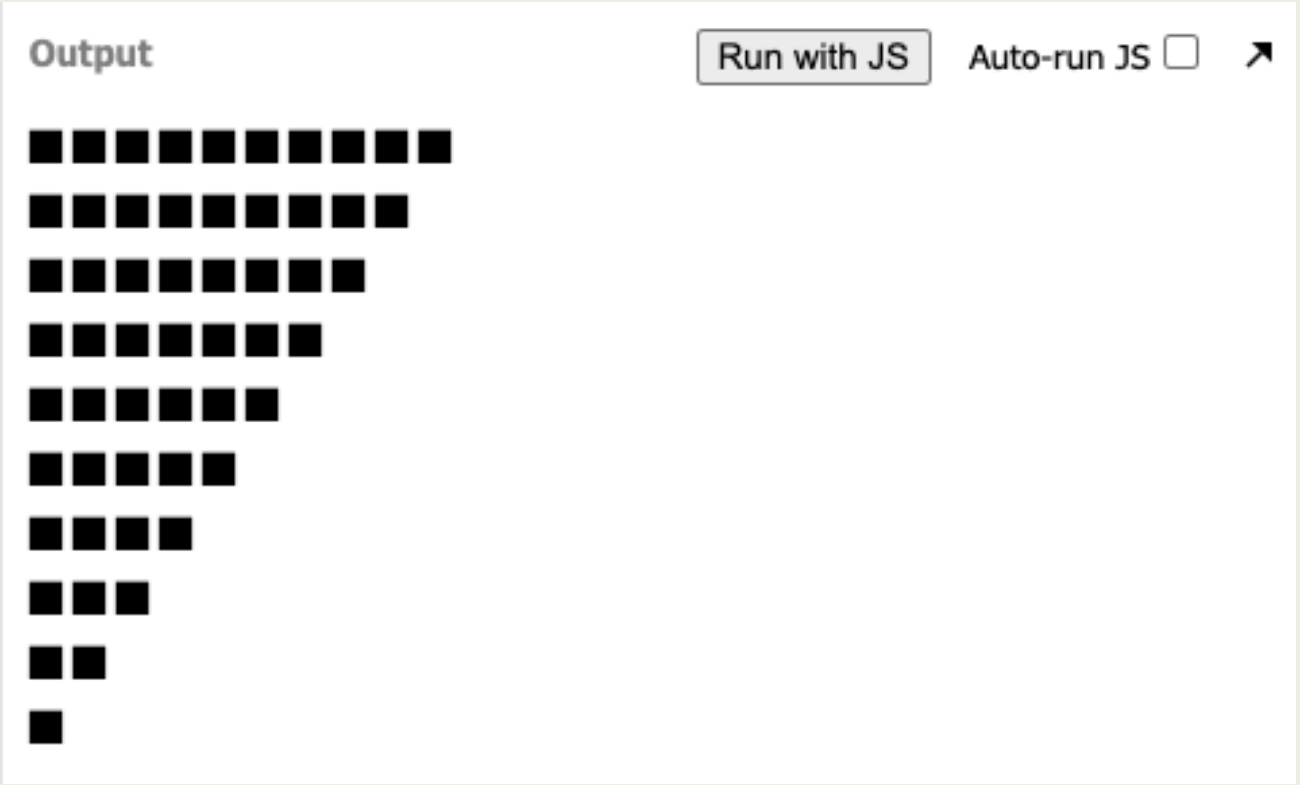
[演習] 模様を描画する①

- 出力が次になるように、i 段目に i 個の■を表示するプログラムを作成しよう。階段は10段ある。



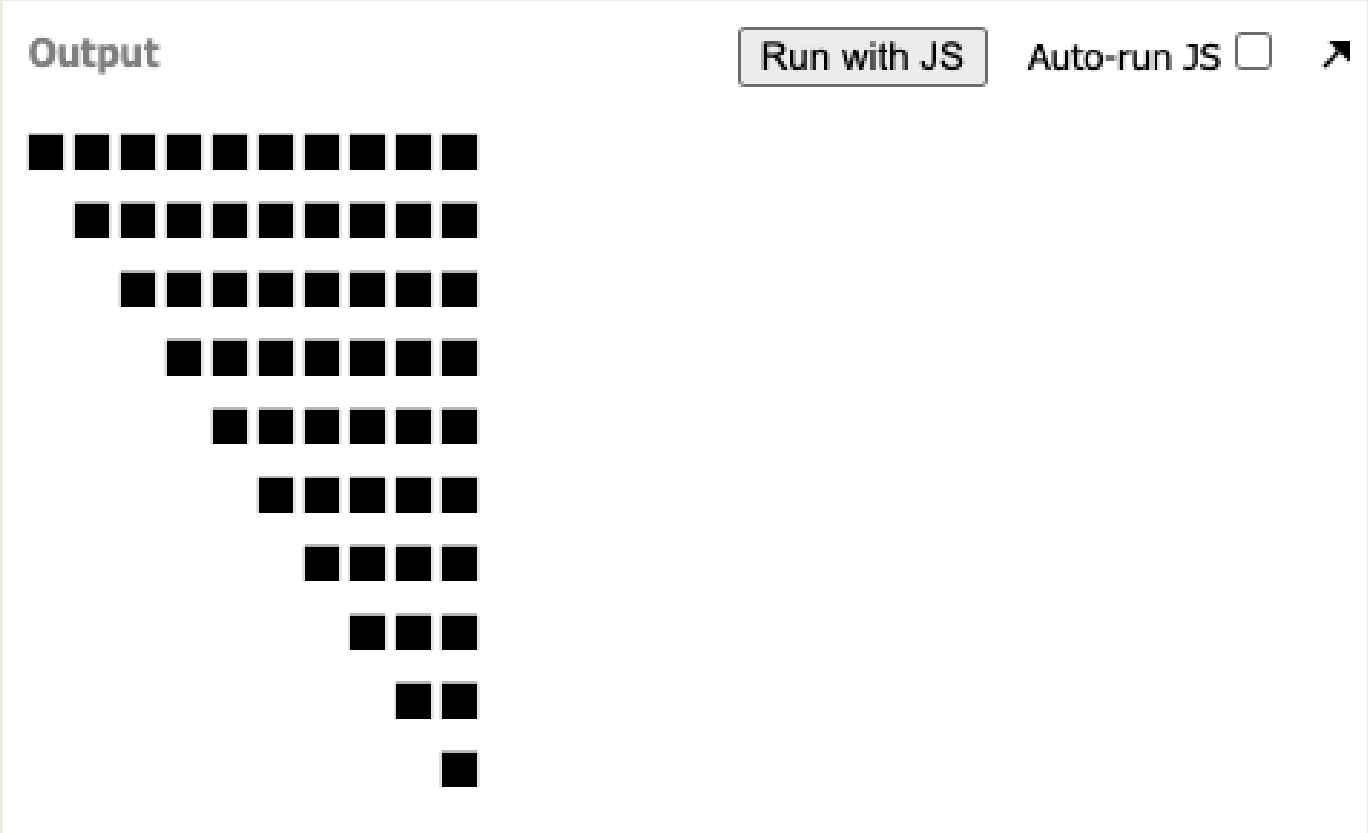
[演習] 模様を描画する②

- 「模様を描画する①」の類題。出力が次になるようにプログラムを作成しよう。



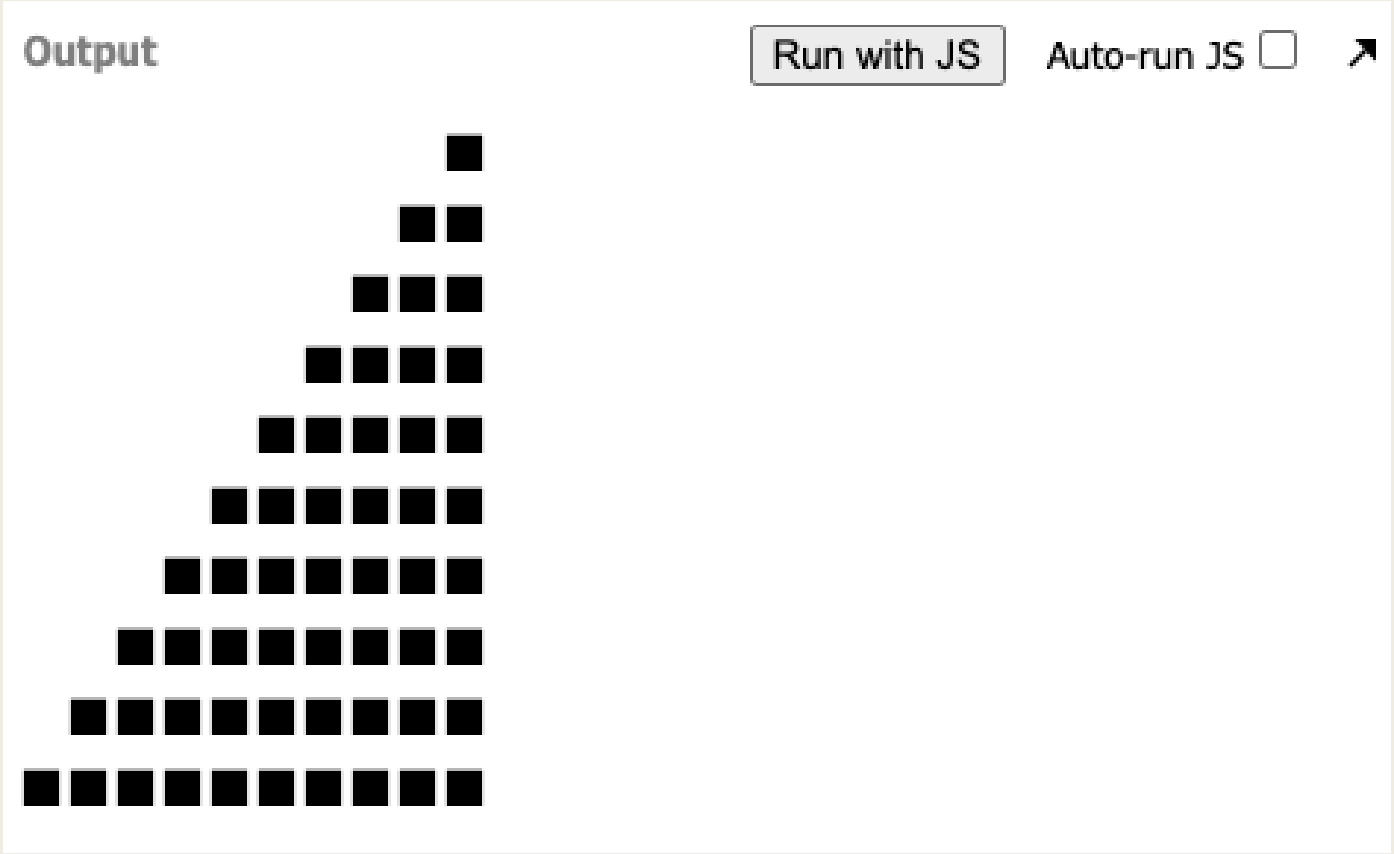
[演習] 模様を描画する③

- 出力が次になるようにプログラムを作成しよう。
- if文 も使ってみよう。



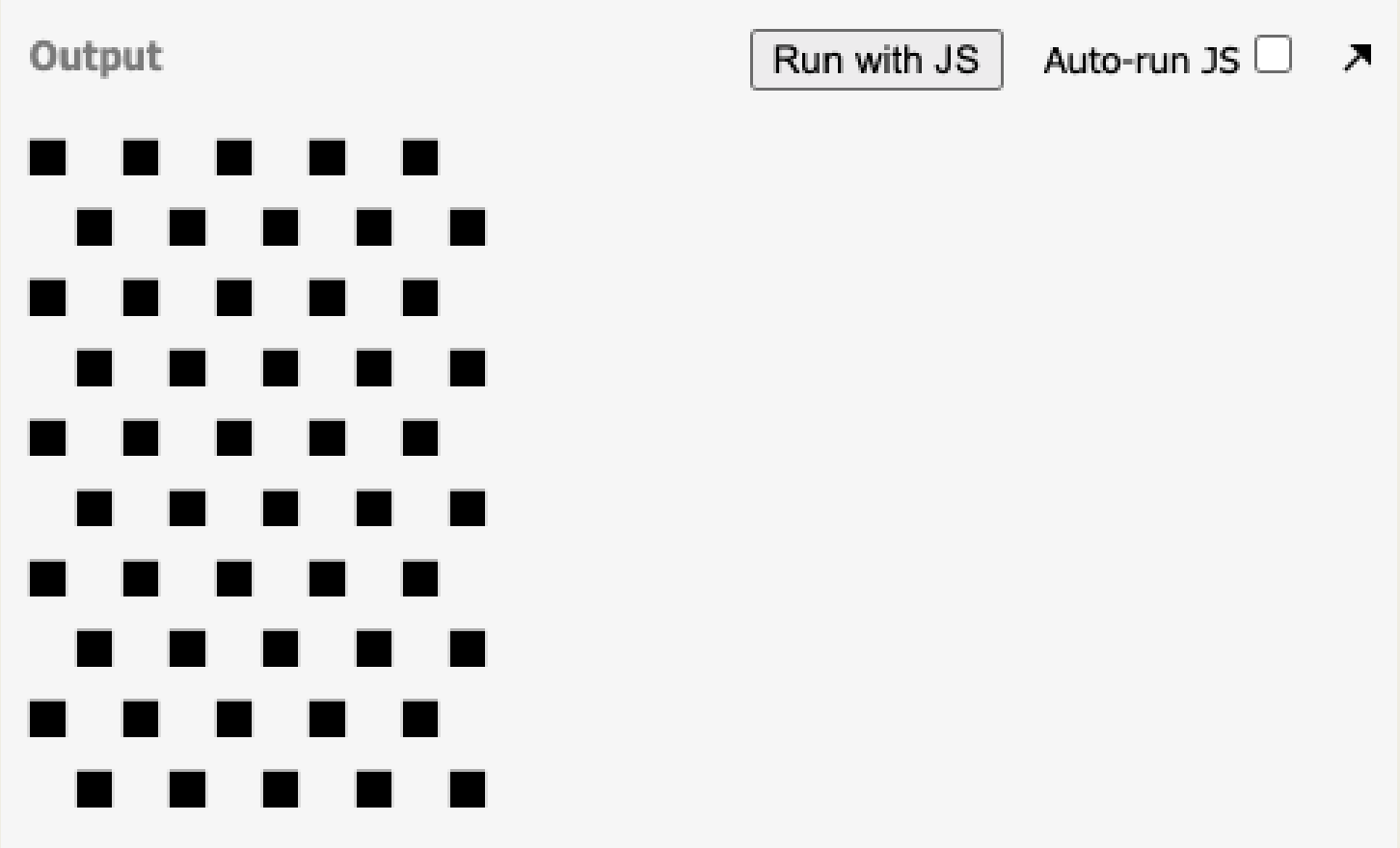
[演習] 模様を描画する④

- 出力が次になるようにプログラムを作成しよう。



[演習] 模様を描画する⑤

- 出力が次になるようにプログラムを作成しよう。



[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。

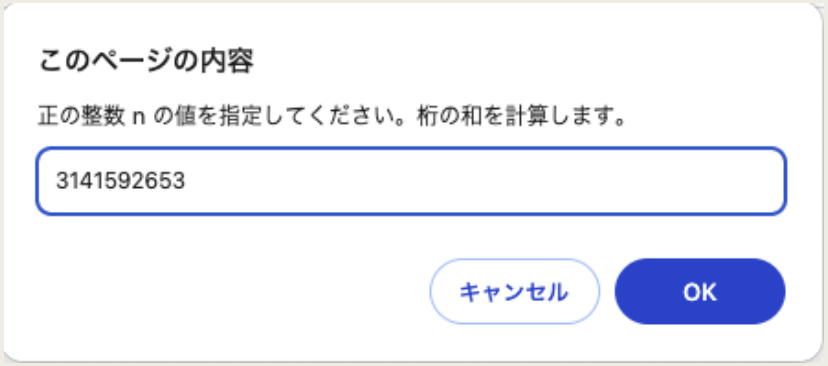
桁の和は 7 です。

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。

桁の和は 15 です。



桁の和は 39 です。