

# プログラミング

第10回  
オブジェクト

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II</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	●	

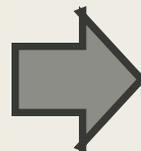
# [準備]コードの新規作成①

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>
```

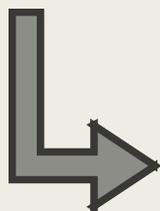
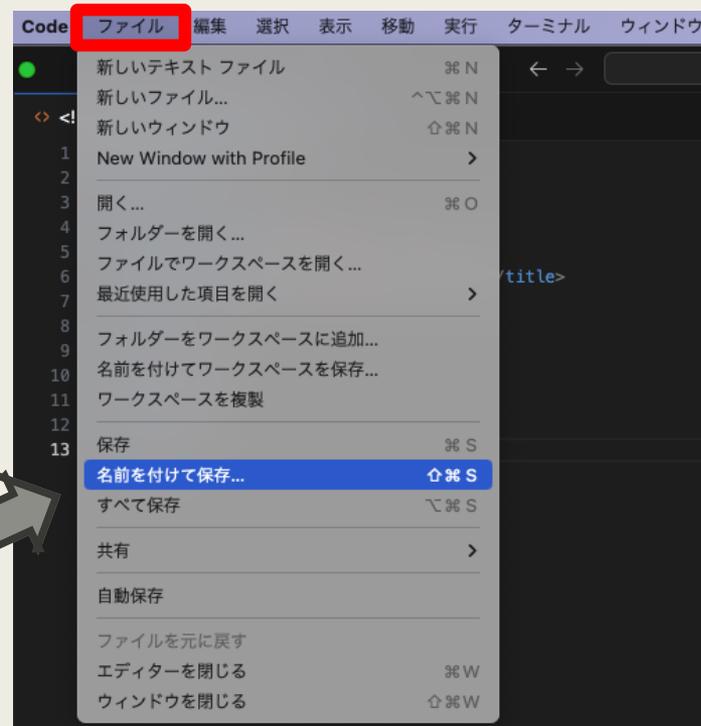
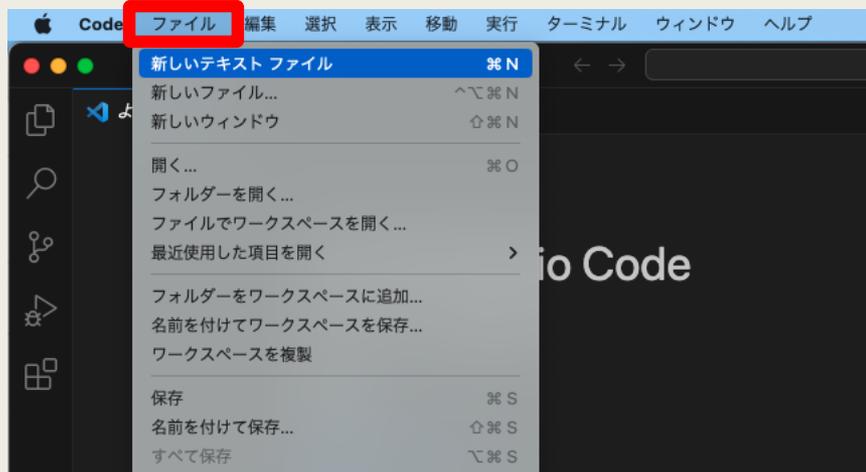
コピー

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

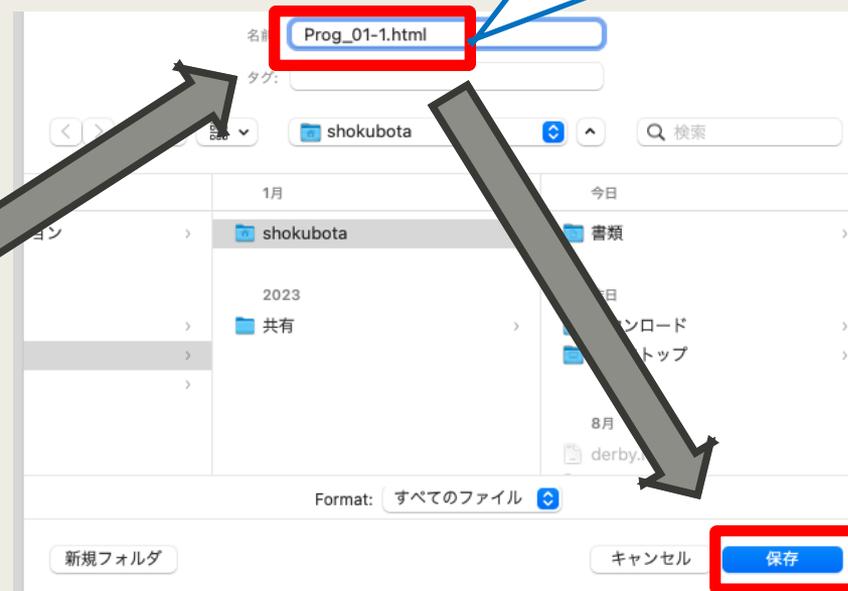
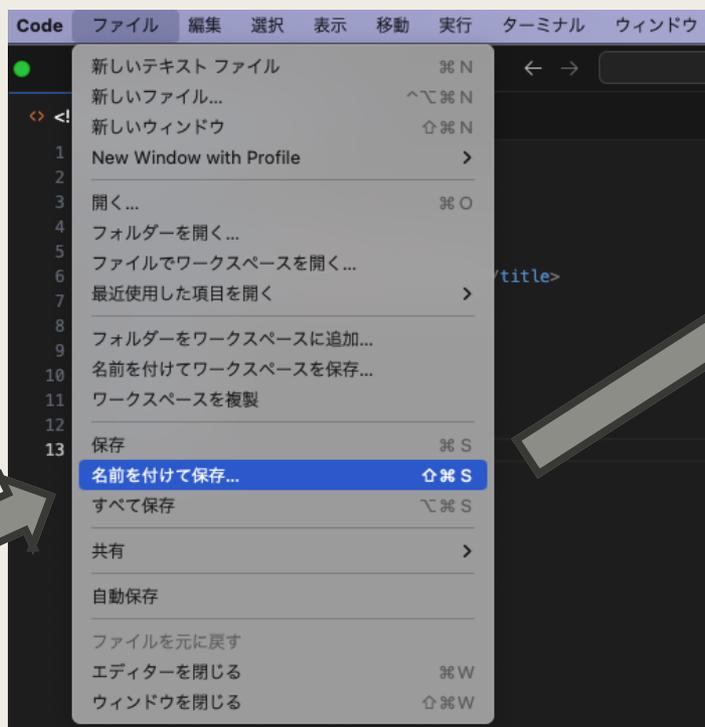


# [準備]コードの新規作成②

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\_10-1.html」  
とつける。



# [準備]作業環境を整える

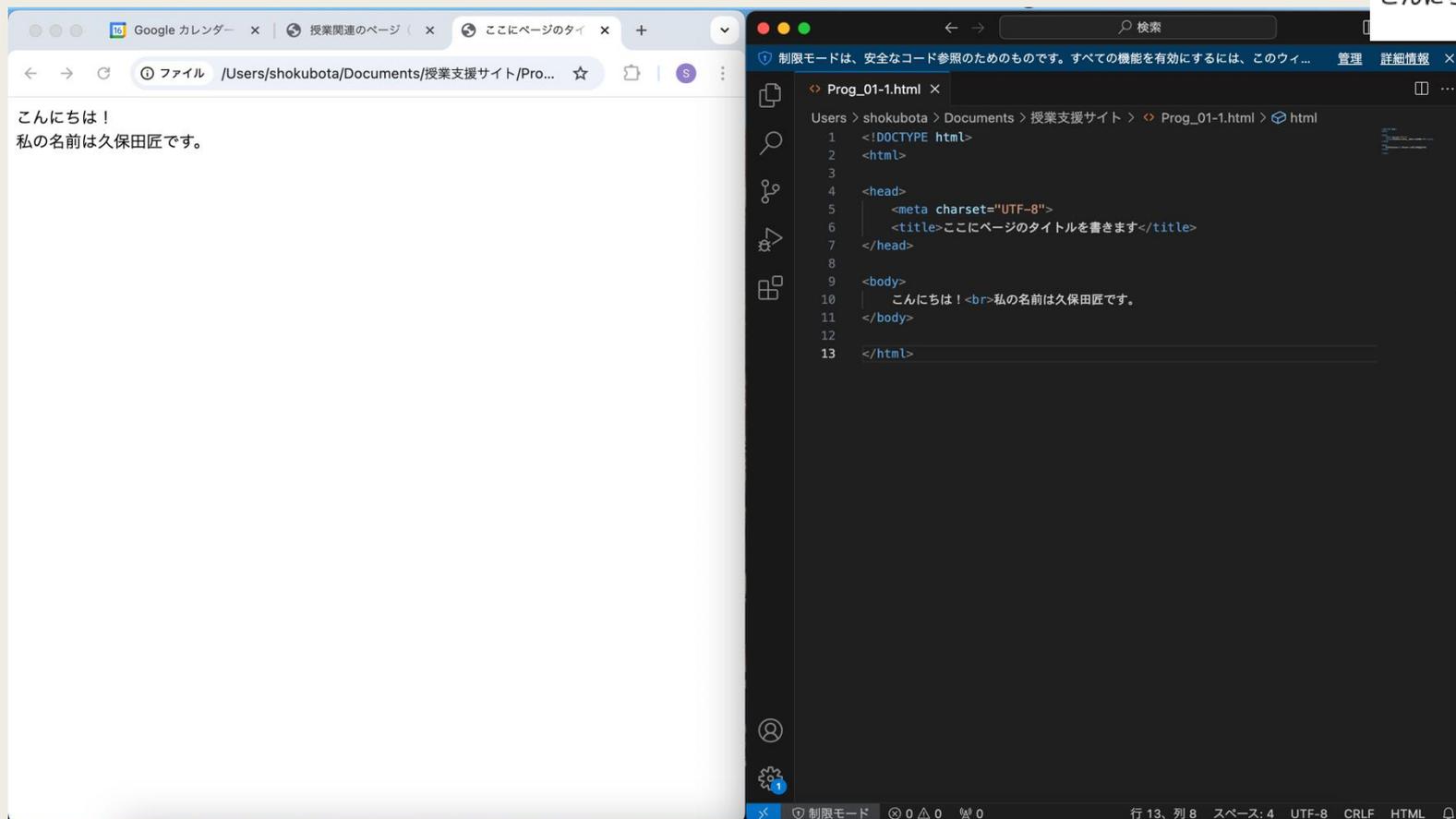
いつもの作業

- 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！



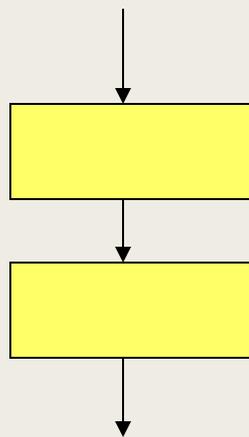
# [復習]構造化定理

初心者が学ぶプログラム  
はだいたい当てはまる

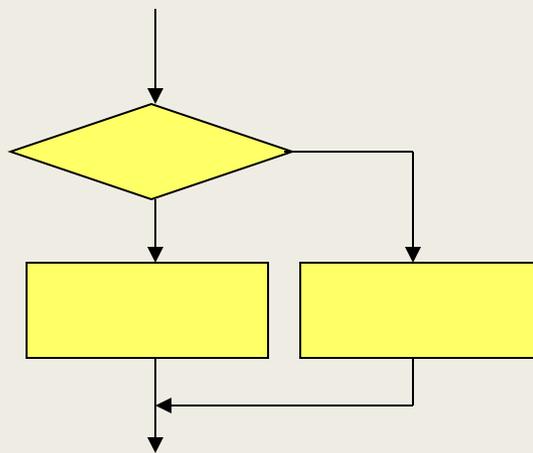
- 「単純な逐次的フローで、ひとつの入力からひとつの出力を得るような処理を持つプログラムは以下の3つの基本構造の組み合わせで記述できる」という定理。

- 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)。

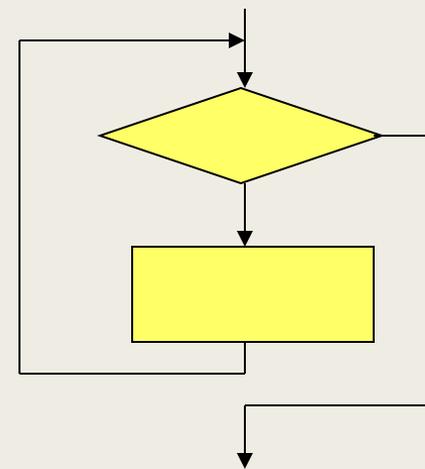
順次



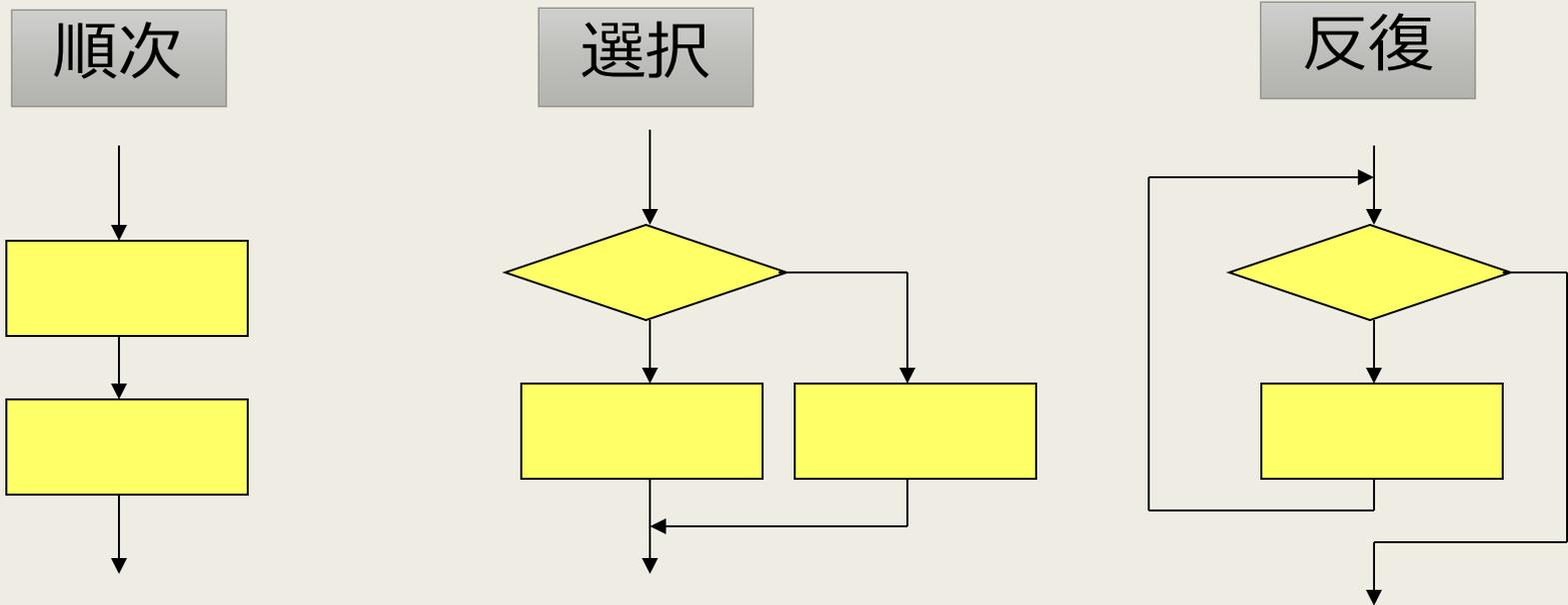
選択



反復



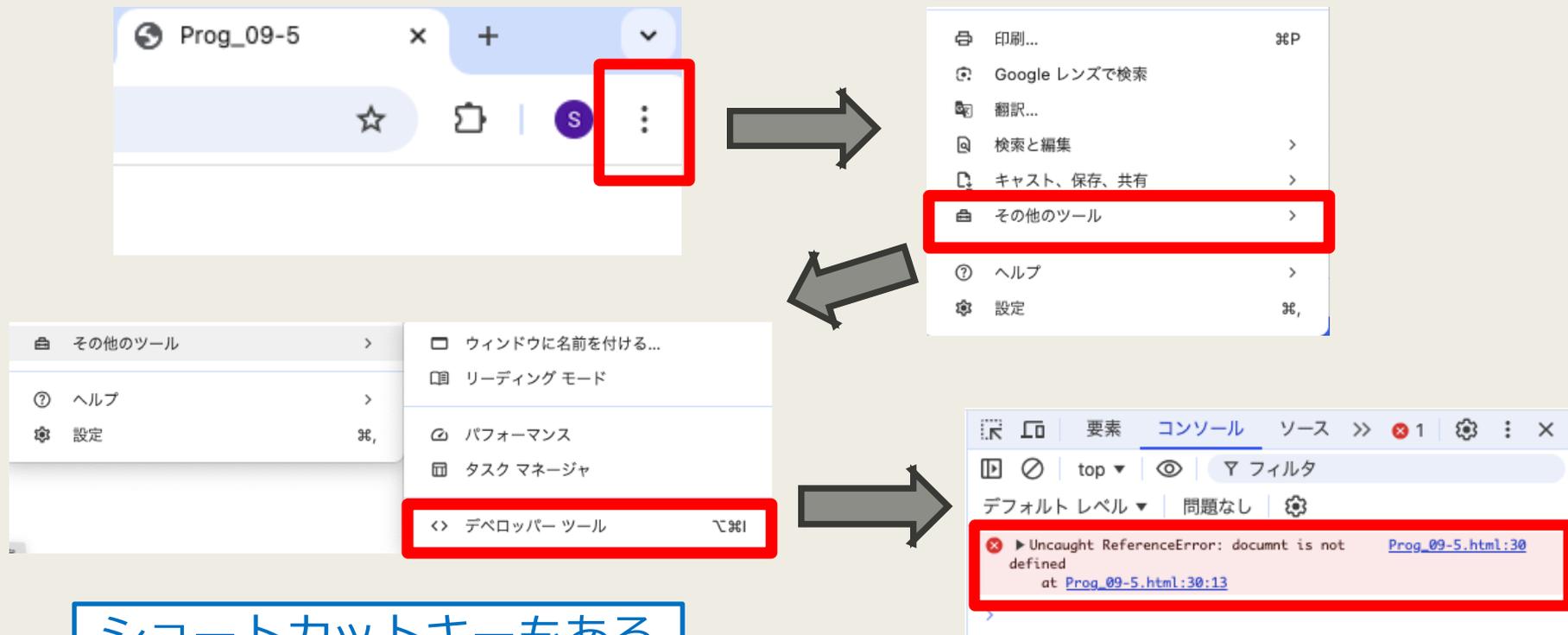
# [復習]構造化定理



- 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。
- 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。

# [復習]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)

# [再掲]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

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

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。

# オブジェクト

- 今日の内容は小難しいかも。
- 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

# インスタンスの生成（実体化）

- 今日は主に Math, String, Date オブジェクトを紹介する。
- 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。
- まずは Math オブジェクトから説明する。

## 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 $\pi$

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 オブジェクト

- Math オブジェクトのプロパティは  
Math.プロパティ名

※これまで扱ってきた  
`document.write(***)` は  
document オブジェクトの  
write メソッド

の形で使用する。

- Math オブジェクトのメソッドは  
Math.メソッド名(引数1, 引数2, ... )

の形で使用する。

- 具体的なメソッドは例えば教科書の p164 の表を参照。
- 必要になったらその都度ネットで調べるのがよい。
- `Math.random()` は0以上1未満の数を生成する。
- `Math.floor(a)` は引数 a の小数点以下を切り捨てる。
- このふたつを組み合わせることで指定した範囲の乱数を発生させることができる。

# サイコロをふる

- 次のコード（サイコロをふって出た目を表示する）を入力してみよう。

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Prog_09-1</title>
7 </head>
8
9 <body>
10   <p>
11     サイコロの目は
12     <script>
13       let dice = Math.floor(Math.random()*6 + 1);
14       document.write(dice);
15     </script>
16     です。
17   </p>
18 </body>
19
20 </html>
```

サイコロの目は 4 です。

サイコロの目は 5 です。

サイコロの目は 2 です。

サイコロの目は 4 です。

ページを再読み込みするたびに  
出力が変わる。

$$0 \leq x < 1$$

$$1 \leq 6x + 1 < 7$$

- `Math.random()` で生成される数の範囲は 0以上1未満。
- `Math.random()*6 + 1` の範囲は 1以上7未満。
- よって、`Math.floor(Math.random()*6 + 1)` と入力すれば 1, 2, 3, 4, 5, 6 が等確率で出る。

# [演習]1次方程式を自動生成する

- 次の出力を参考に、1次方程式を自動生成し、その解（近似値でよい）を表示するプログラムを作成せよ。
- ただし、 $ax = b$  の  $a$  は 2以上9以下の整数、 $b$  は 1以上10以下の整数となるように乱数を発生させよ。

1次方程式  $2x=6$  を解くと、 $x=3$  である。

1次方程式  $5x=2$  を解くと、 $x=0.4$  である。

1次方程式  $7x=8$  を解くと、 $x=1.1428571428571428$  である。

# String オブジェクト

- String とは文字列のことを指す。
- 宣言した変数には文字列を代入することもできた。

```
let str1 = “やきにく”;
```

- 変数に文字列を代入すると、実質的に String オブジェクトが実体化される。
  - 厳密には違うのだが JavaScript ではあたかもそのようなものとして扱えるため、String オブジェクトのプロパティやメソッドを利用することができる。

## Stringオブジェクトのプロパティ

文字列の長さ

⋮

## Stringオブジェクトのメソッド

i 番目の文字を取り出す

特定の文字の位置を取り出す

指定した範囲の文字を取り出す

⋮

その他のメソッドは  
教科書の p171 を参照

# String オブジェクト

- 文字列の先頭の文字の位置は 0 番目 なので注意しよう。

や

0番目

き

1番目

に

2番目

く

3番目

- 次のコードを入力してみよう。なお、`str1.length` は文字列の長さを返す命令で、`str1.charAt(i)` は文字列の `i` 番目 を返す命令である。

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Prog_09-3</title>
7 </head>
8
9 <body>
10   <p>
11     <script>
12       let str1 = "やきにく";
13       document.write(str1 + "は" + str1.length + "文字の言葉です。<br>");
14       document.write(str1 + "という言葉の <br>");
15       for(let i=0; i < str1.length; i++){
16         document.write((i+1) + "文字目は「" + str1.charAt(i) +"」<br>");
17       }
18       document.write("です。");
19     </script>
20   </p>
21 </body>
22
23 </html>
```

やきにくは4文字の言葉です。  
やきにくという言葉の  
1文字目は「や」  
2文字目は「き」  
3文字目は「に」  
4文字目は「く」  
です。

# Date オブジェクト

- スペルに注意。
  - Date は日付、Data はデータ。
- Date オブジェクトでは、日付や時刻を取得・管理ができる。
  - 例えば、カレンダーやタイマー機能をもつプログラムを作ることができる。
- Date オブジェクトは、最初に扱った Math オブジェクトと違い **インスタンスの生成** を行う必要がある。
- (Date オブジェクトにおける) インスタンスの生成とは、大雑把に言えば「特定の瞬間の日時データを取得して、それを操作可能な形式にする」のような意味。

```
let now = new Date();
```

Date オブジェクトの  
インスタンスを生成

- 上のように入力すると、変数 now には、この命令を実行した瞬間の日時情報が入り、Date オブジェクトのプロパティやメソッドが使えるようになる。

# 習うより慣れろ

- オブジェクトという概念は便利であるが小難しい。
- まずは「細かい理屈は知らないけど使い方は分かるぞ」という水準を目指そう。
- Date オブジェクトの主なメソッドは教科書の p157 にある。
- 次のコードを入力してみよう。



現在時刻は13時59分です。

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Prog_09-4</title>
7 </head>
8
9 <body>
10  <p>
11    <script>
12      let now = new Date;
13      let hours = now.getHours();
14      let minutes = now.getMinutes();
15      document.write("現在時刻は" + hours + "時" + minutes + "分です。");
16    </script>
17  </p>
18 </body>
19
20 </html>
```

# [演習] じゃんけんのプログラム

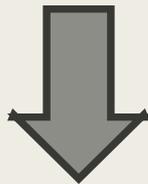
- 乱数を使うと簡単なゲームが作れる。
- 次の出力結果を参考に、コンピュータとじゃんけんの勝負をするプログラムを作ってみよう。

このページの内容

じゃんけんをします。グーなら「0」、チョキなら「1」、パーなら「2」と入力してください。

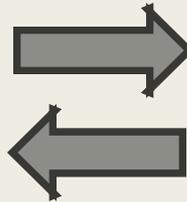
キャンセル OK

入力が正しい



あなたの手は チョキ です。  
コンピュータの手は グー です。  
結果はあなたの負けです。

入力が正しくない



最初の処理に戻る

このページの内容

不正な入力です。

OK

- Step.1 入力が不正かどうか判定する。
- Step.2 自分の手を表示させる。
- Step.3 コンピュータの手を表示させる。
- Step.4 勝敗を判定する。

# [演習]教科書を熟読しよう

- 今日の内容は教科書の p148～p174 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。
- Date オブジェクトの取り扱いは少し難しいので、Math オブジェクトや String オブジェクトから読むといいかもしれない。
- 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。