

# プログラミング

第14回

数式の表示 (TeXについて)

最終課題の案内

久保田 匠

# [連絡] 第15回はオンデマンド

- **来週（2月13日）** は久保田出張のためオンデマンド講義。



※政治的な意図はありません

- 最終課題に取り組んでください。今日の授業で詳細な採点基準などをお話します。

## 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	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) (オンデマンド)	●	(なし)
第10回	オブジェクト	●, ★	(なし)
第11回	配列	●, ★	<a href="#">Prog_11-1</a>
第12回	ユーザー定義関数	●, ★	<a href="#">Prog_12-1</a>
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) 最終課題の案内	▲	<a href="#">Prog_14-1</a>
第15回	ウェブツールを開発してみよう (オンデマンド)	▲	<a href="#">課題提出例</a>

**この回は授業なし**

# [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。



## 授業用ホームページ (久保田)

2025年度前期担当科目

	月曜	火曜	水曜	木曜	金曜
1限					
2限	確率統計II			確率統計II	
3限				線形数学演習I	確率統計I
4限	4年ゼミ				(オフィスアワー)
5限					

2025年度後期担当科目

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

その他のコンテンツ → ● ●

数学教育講座 久保田匠 (自然科学棟 521 研究室)  
Email: skubota [at] aeucc.aichi-edu.ac.jp

## 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	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		<a href="#">Prog_11-1</a>
第12回	ユーザー定義関数		<a href="#">Prog_12-1</a>
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)	●	<a href="#">Prog_14-1</a>
第15回	ウェブツールを開発してみよう		<a href="#">課題提出例</a>

# [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	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		<a href="#">Prog_11-1</a>
第12回	ユーザー定義関数		<a href="#">Prog_12-1</a>
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)	●	<a href="#">Prog_14-1</a>
第15回	ウェブツールを開発してみよう		<a href="#">課題提出例</a>



## Prog\_14-1

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Prog_14-1</title>
  <!-- HTML で TeX を使えるようにするための呪文 (ここから) -->
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/katex/dist/katex.min.css" crossorigin="anonymous">
  <script defer src="https://cdn.jsdelivr.net/npm/katex/dist/katex.min.js"
crossorigin="anonymous"></script>
  <script defer src="https://cdn.jsdelivr.net/npm/katex/dist/contrib/auto-
render.min.js" crossorigin="anonymous"></script>
  <script>
    window.onload = function() {
      renderMathInElement(document.body, {delimiters: [
        {left: "\\[", right: "\\]"}, display: true},
        {left: "$", right: "$"}, display: false}
      ]});
    };
  </script>
</head>
</html>
```

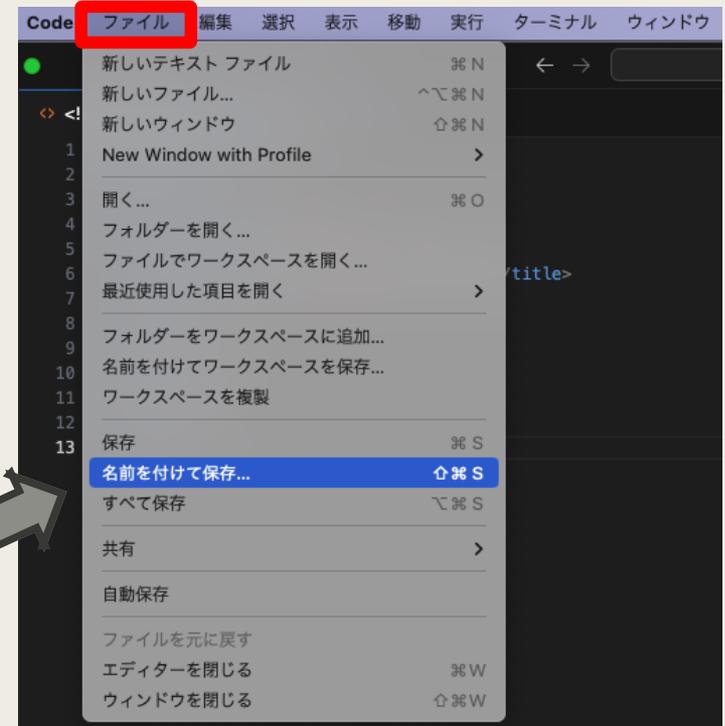
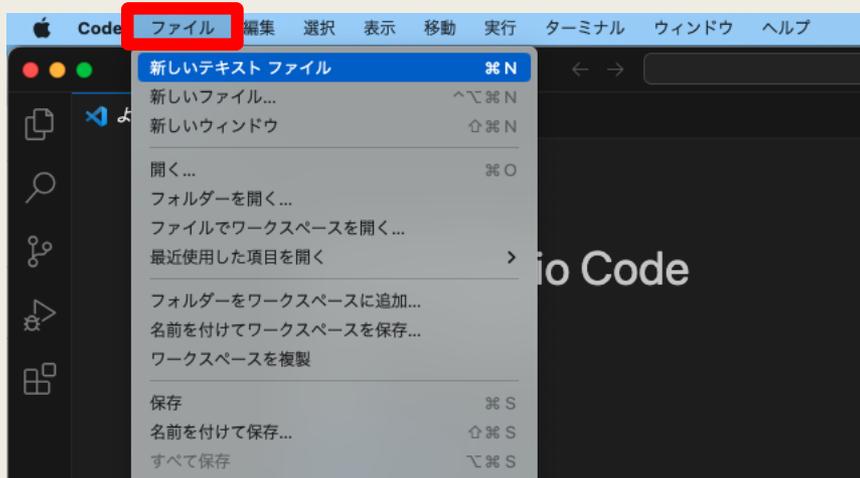
コピー

今日は「Prog\_14-1」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」。



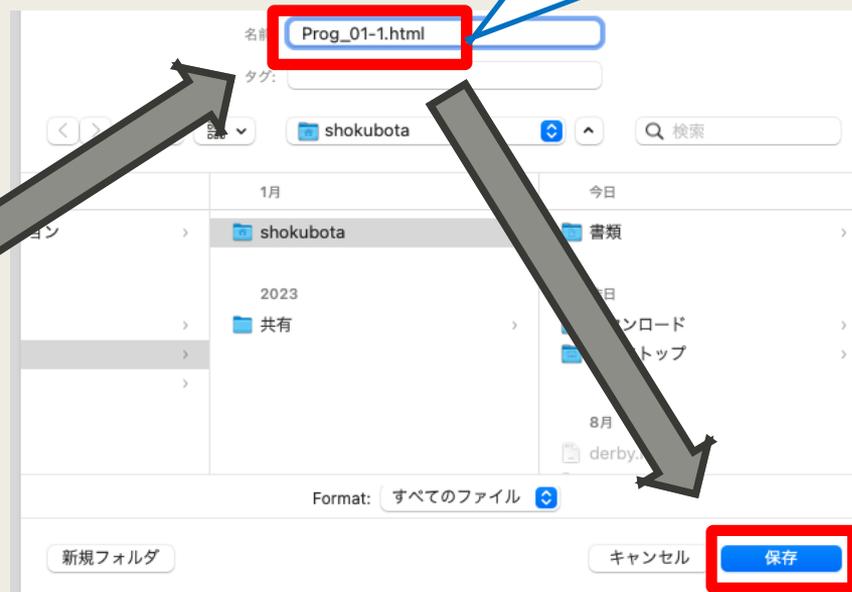
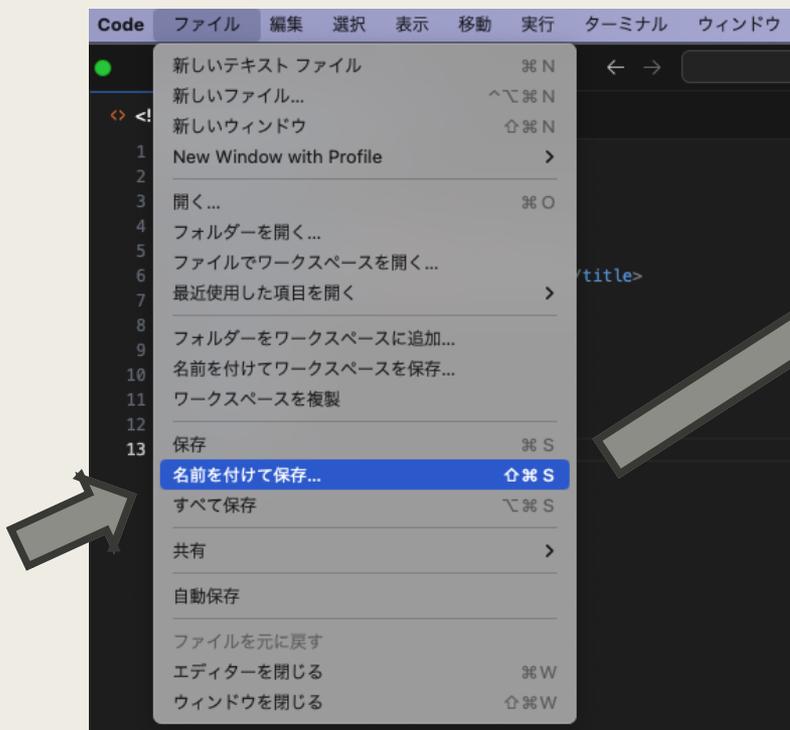
```
ようこそ << >> <!DOCTYPE html> Untitled-1 ●
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>ここにページのタイトルを書きます</title>
7 </head>
8
9 <body>
10   こんにちは！
11 </body>
12
13 </html>
```

# [準備]コードの新規作成②



- 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\_14-1.html」とつける。



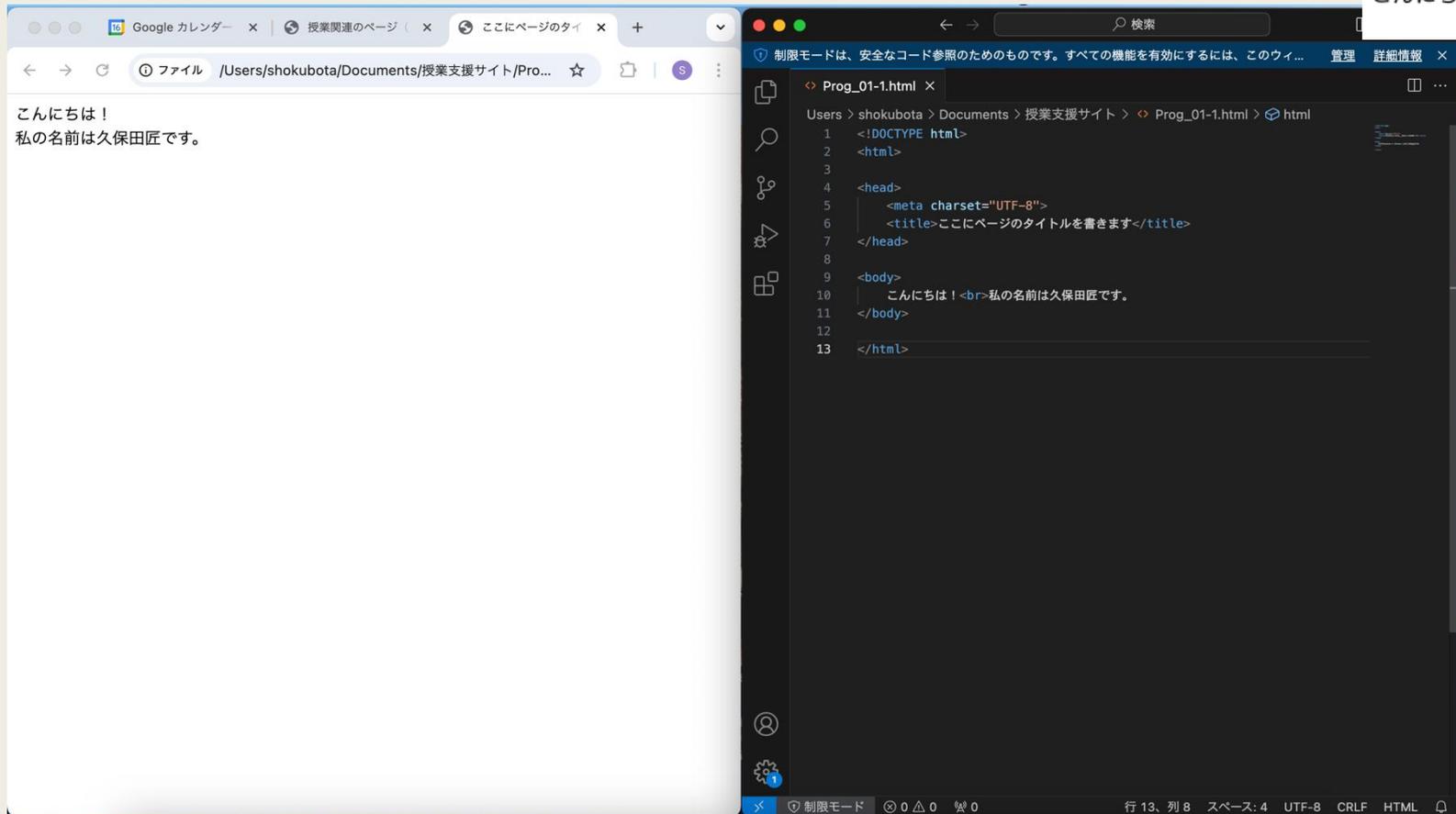
# [準備]作業環境を整える

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。

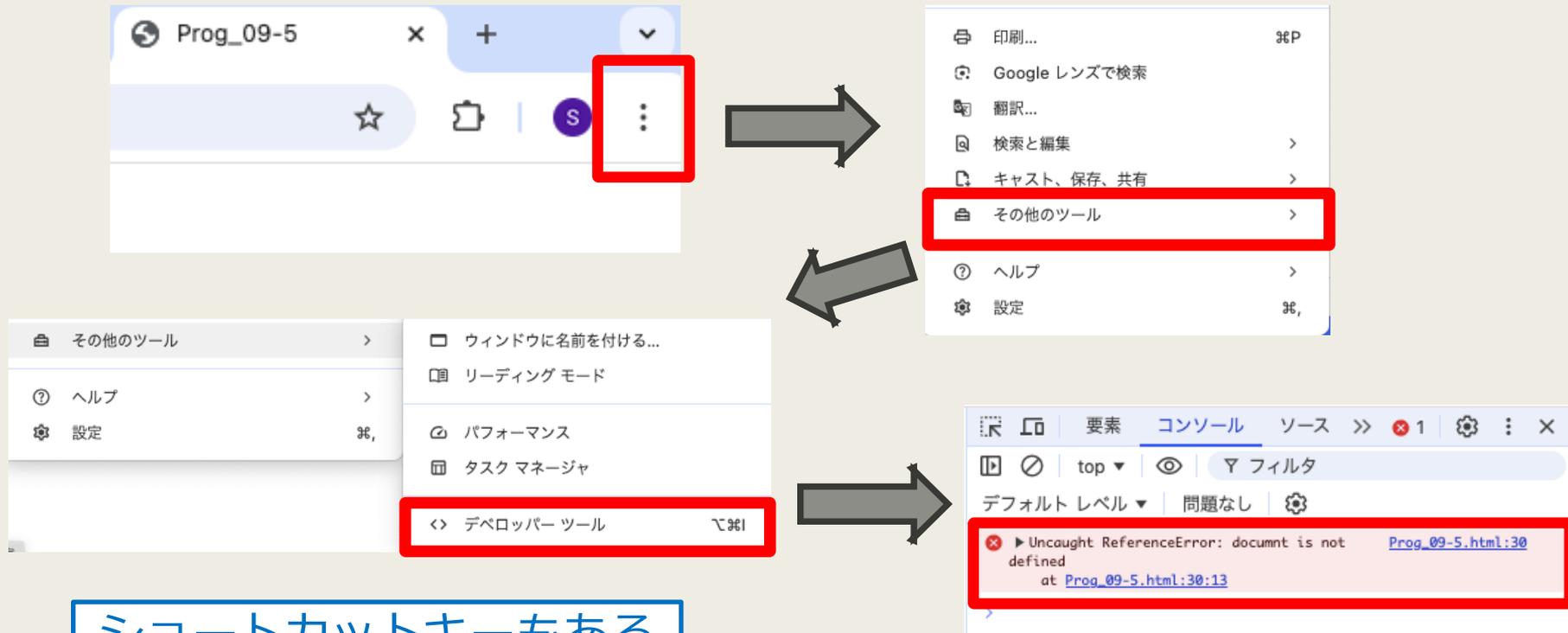
ダブル  
クリック

← → 🔄 ⓘ ファイル |  
こんにちは！



# [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) 8



## 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。

## 逆行列

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。

ボタンをクリックすると答えが表示される

- イベントとは、ユーザーの操作やブラウザの動作といった「何か起きたこと」を知らせる合図のこと。例えば...
  - ボタンをクリックする
  - マウスポインタが特定の領域に置かれる
  - ページが読み込まれる
- イベントハンドラは、イベントが発生したときに実行される関数。
- 上にある「ボタンをクリックすると答えが表示されるウェブサイト」の仕組みは、
  - ① 問題と答えを生成し、答えは透明色の字で表示しておく。
  - ② 「答え」ボタンがクリックされたら「透明色の字を黒字に変える関数」を発動。

# [復習]ボタンをクリックすると文字の色が変わる

- 次はボタンをクリックすると文字の色が変わるプログラム。

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Prog. 12-1</title>
7   <script>
8     function changeColor(){
9       document.getElementById("001").style.color = "red";
10    }
11  </script>
12 </head>
13
14 <body>
15   <p>
16     右にボタンを用意します。<button type="button" onclick="changeColor()">ボタン</button> <br>
17     このボタンを押すと
18     <span id="001"> この文字の色が </span>
19     変わります。
20   </p>
21 </body>
22
23 </html>
```

右にボタンを用意します。

ボタン

このボタンを押すと この文字の色が変わります。



右にボタンを用意します。 ボタン

このボタンを押すと **この文字の色が** 変わります。

- <button>タグでボタンが用意される。
- ブラウザ側でボタンが押されると関数 changeColor が呼び出される。
- 関数 changeColor は id が 001 であるタグ内の文字の色を赤色に変更する処理を行う。

# [復習]1次方程式を出題するプログラム

- ここまでの知識で「数学の問題を自動生成し、ボタンをクリックすると答えが表示されるウェブサイト」が作れる。
- 前回、1次方程式を自動生成して「答えを表示する」というボタンをクリックすると答えが表示されるプログラムを作ってもらった。

```
14 <body>
15 <!--
16 <p>
17     ... 右にボタンを用意します。<button type="button" onclick="changeColor()">ボタン</button> <br>
18     ... このボタンを押すと
19     ... <span id="001"> この文字の色が </span>
20     ... 変わります。
21 </p>
22 </body>
23 <p>
24     ... 1次方程式
25 <script>
26     ... let a = Math.floor(Math.random()*8 + 2);
27     ... let b = Math.floor(Math.random()*10 + 1);
28     ... document.write(a + "x=" + b);
29 </script>
30 </p>
31 <span style="color:transparent" id="001">
32 <script>
33     ... document.write("x=" + b/a);
34 </script>
35 </span>
36 <br>
37     ... である。<button type="button" onclick="changeColor()">答えを表示する</button>
38 </p>
39 </body>
```

id はひとつのタグに  
ひとつだけ対応するので  
先程の内容は <!-- --> で囲む

25~29行目は  
第10回でやった

答えも生成しておくが  
透明色で表示する

1次方程式  $4x=-6$  の解は  
である。



1次方程式  $4x=-6$  の解は  
 $x=-1.5$   
である。

# [復習]分数

1次方程式  $4x=-6$  の解は  
 $x=-1.5$   
である。

- 先のプログラムでは答え（1次方程式の解）が小数で表示された。
- 答えを分数で表示させることはできないだろうか？
- 1次方程式  $ax = b$  の解は  $x = b/a$  であるから、基本的には `document.write(b + "/" + a);` のように入力すればよさそうである。しかし...
  - 分数  $b/a$  は約分できる可能性がある。
  - 分母が 1 の場合は整数の形で表示する必要がある。
- 約分を実行するため、今回はユークリッドの互除法のプログラムを実装した。

# [再掲]もっときれいな数式を出力したい

1次方程式  $6x=4$  の解は  
 $x=2/3$   
である。

1次方程式  $9x = 6$  の解は  
 $x = \frac{2}{3}$   
である。

- 前回の内容で、1次方程式の解を分数の形で表示するところまではできるようになった。
- しかし、教科書で見えるような、もっと見栄えの良い分数（を含むあらゆる数式）を表示する方法はないだろうか？
- きれいな数式を表示させるためには TeX（てふ）と呼ばれるツールを使う。
- TeX は4年生になって卒論を書くときにも使用する可能性があるがあるので、この機会に雰囲気を感じておこう。

# [再掲]第10回以降に学ぶこと

[第14回]  
きれいな数式を  
表示する

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。

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

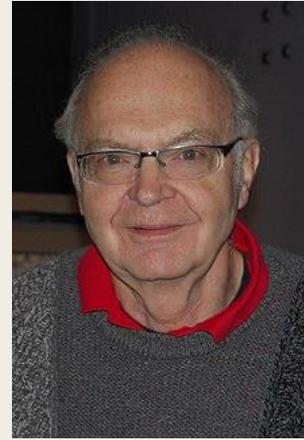
[第12回]  
処理のかたまりを  
定義する

- 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。

# TeX とは

[https://en.wikipedia.org/wiki/Donald\\_Knuth](https://en.wikipedia.org/wiki/Donald_Knuth)

- TeX (てふ) は、主に数学や科学の分野で使われる文書作成システム。
- 文字や数式をきれいに整えて印刷物のように表示できる。
- 開発者はアメリカの数学者・計算機科学者のドナルド・クヌース。



Donald E. Knuth  
(1938-)

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

Word による数式の表示  
(最近は Word でもきれいかも)

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

TeX による数式の表示

# HTML で TeX を使えるようにする

- 今日のサンプルコードの head部 にあるような呪文が書かれてあると html で TeX が使えるようになる。
- ただし、インターネットにつながっていないと数式に変換されないので注意。
- body部 に次を入力してみよう。

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Prog_13-1</title>
7   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/katex/dist/katex.min.css" crossorigin="anonymous">
8   <script defer src="https://cdn.jsdelivr.net/npm/katex/dist/katex.min.js" crossorigin="anonymous"></script>
9   <script defer src="https://cdn.jsdelivr.net/npm/katex/dist/contrib/auto-render.min.js" crossorigin="anonymous"></script>
10
11   <script>
12     window.onload = function() {
13       renderMathInElement(document.body, {delimiters: [
14         {left: "\\[", right: "\\]", display: true},
15         {left: "$", right: "$", display: false}
16       ]});
17     };
18   </script>
19 </head>
20
21 <body>
22   <p>
23     多項式  $x^2 + 2x + 1$  を因数分解すると  $(x+1)^2$  になります。
24   </p>
25 </body>
26
27 </html>
```

多項式  $x^2 + 2x + 1$  を因数分解すると  $(x + 1)^2$  になります。

# インライン数式とディスプレイ数式

- TeX を使って数式を表示する方法は大きく分けてふたつある。
- \$ (ドルマーク) で囲む。 [例:  $a+b=c$ ]
  - \$で囲むと数式が文中に表示される。文章の途中にちょっとした数式を入れたいときに使う。
  - これを **インライン数式** という。
- \[ と \] で囲む (円マークとバックスラッシュは同じ) 。
  - \[ と \] で囲むと数式が独立して表示され、行全体を使って中央に配置される。特に重要な数式や、長い数式を見やすく表示したいときに使う。
  - これを **ディスプレイ数式** という。
- 上添字は ^ を使い、下添字は \_ を使う。 [例:  $x^2$ ,  $a_1$ ]
- 添字が2文字以上ある場合は中括弧で囲む。 [例:  $a_{n+1}$ ]



# インライン数式とディスプレイ数式

- TeX を使って数式を表示する方法を大きく分けてふたつある。
- $\$$  (ドルマーク) で囲む。[例:  $\$a+b=c\$$ ]
- $\backslash[$  と  $\backslash]$  で囲む (円マークとバックスラッシュは同じ) 。
- 上添字は  $\wedge$  を使い、下添字は  $\_$  を使う。[例:  $x^2$ ,  $a_{1}$ ]
- 添字が2文字以上ある場合は中括弧で囲む。[例:  $a_{\{n+1\}}$ ]
- body部 に次を追加してみよう。

```
21 <body>
22   <p>
23     多項式  $x^2 + 2x + 1$  を因数分解すると  $(x+1)^2$  になります。
24   </p>
25   <p>
26     方程式
27      $\backslash[ x^2 - x - 2 = 0 \backslash]$ 
28     を解くと  $x = -1, 2$  です。
29   </p>
30 </body>
```

$\$$ の前後は半角スペースをあけることを推奨

ディスプレイ数式は画面の真ん中に表示される。  
見栄えはCSSでコントロールできるが今日の授業では省略。

多項式  $x^2 + 2x + 1$  を因数分解すると  $(x + 1)^2$  になります。

方程式

$$x^2 - x - 2 = 0$$

を解くと  $x = -1, 2$  です。

# [演習]分数とルート

¥

円マーク

\

バックスラッシュ

- 分数は  $\frac{1}{3}$  のように入力する。
- ルートは  $\sqrt{2}$  のように入力する。
- これ以外にも様々な数式が表現できる。すべてを頭に入れておく必要はなく、適宜調べて使えればよい。
- 出力が次になるように入力してみよう。授業で解説していないコマンドはネットで調べよう（「TeX プラスマイナス」など）。

2次方程式  $ax^2 + bx + c = 0$  の解は

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

で与えられます。

$a_1 = 1, a_{n+1} = \frac{a_n}{2a_n+1}$  で定められる数列  $\{a_n\}$  の一般項は

$$a_n = \frac{1}{2n-1}$$

です。

次の定積分を求めよ。

$$\int_0^{\frac{\pi}{2}} \sin^2 x \cos x dx$$

中括弧は  $\{\}$  のように入力する。

インライン数式で  $\frac{}{}$  を使うと分母分子が小さく表示される。これを回避したい場合は  $\dfrac{}{}$  を使うかディスプレイ数式で入力する。

# scriptタグ内での\の扱いについて

円マークと  
バックスラッシュは同じ



- scriptタグ内では\は別の意味を持っている。
- scriptタグ内で数式を入力するとき、「\」1個分の入力を反映させたいときは\を2個入力する。
  - `\frac{1}{2}` と入力したいときは `\\frac{1}{2}` と入力。
- 次を入力してみよう。

scriptタグ外なので\はひとつ

```
44 <p>
45   方程式  $\frac{2}{3}x = \frac{5}{2}$  の解は
46   <script>
47     document.write("$x= \\frac{15}{4}$");
48   </script>
49   です。
50 </p>
```

scriptタグ内なので\はふたつ

$$\text{方程式 } \frac{2}{3}x = \frac{5}{2} \text{ の解は } x = \frac{15}{4} \text{ です。}$$

# 1次方程式とその解を TeX 形式で表示

- 以下は、1次方程式を自動生成するプログラム（前回）を TeX 形式で表示するよう修正したものである。
- 先週のコードに加筆してみよう。

```
4 <head>
5 <meta charset="UTF-8">
6 <title>Prog 12-4</title>
7 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/katex/dist/katex.min.css" crossorigin="anonymous">
8 <script defer src="https://cdn.jsdelivr.net/npm/katex/dist/katex.min.js" crossorigin="anonymous"></script>
9 <script defer src="https://cdn.jsdelivr.net/npm/katex/dist/contrib/auto-render.min.js" crossorigin="anonymous"></script>
10
11 <script>
12 window.onload = function() {
13   renderMathInElement(document.body, {delimiters: [
14     {left: "\\(", right: "\\)", display: true},
15     {left: "$", right: "$", display: false}
16   ]});
17 };
18 </script>
19
20 <script>
21 function changeColor(){
22   document.getElementById("001").style.color = "red";
23 }
24
25 function findGCD(a,b){
26   while(b != 0){
27     let r = a%b;
28     a = b;
29     b = r;
30   }
31   return a;
32 }
33 </script>
34 </head>
```

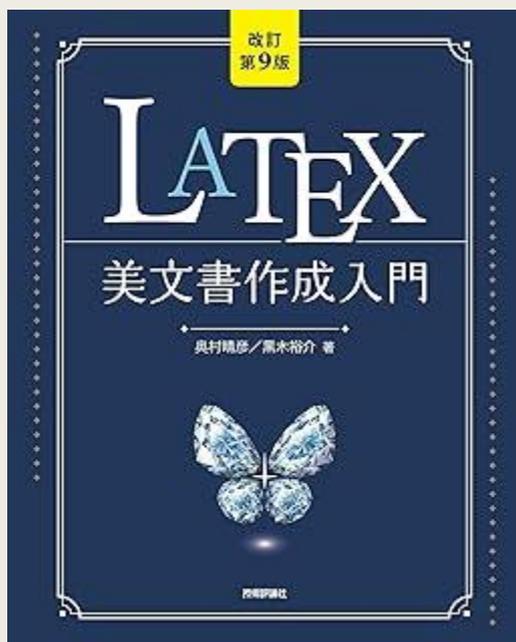
数式を呼び込むための呪文  
忘れずに書くこと

1次方程式  $6x = 10$  の解は  
 $x = \frac{5}{3}$   
である。

```
39 <body>
40 <p>
41   1次方程式
42   <script>
43     let a = Math.floor(Math.random()*8 + 2);
44     let b = Math.floor(Math.random()*10 + 1);
45     document.write("$" + a + "x=" + b + "$");
46   </script>
47   の解は <br>
48   <span style="color: black" id="001">
49     <script>
50       let gcd = findGCD(a,b);
51       if(a/gcd == 1){
52         document.write("$x=" + (b/gcd) + "$");
53       } else{
54         document.write("$x= \\dfrac{" + (b/gcd) + "}{" + (a/gcd) + "$"}");
55       }
56     </script>
57   </span>
58   <br>
59   である。 <button type="button" onclick="changeColor()">答えを表示する</button>
60 </p>
61 </body>
```

# 参考文献

- TeX についてもっと詳しく知りたい人は次の書籍がおすすめ。
- 奥村晴彦, 黒木裕介 『[改訂第9版] LaTeX美文書作成入門』



ゼミに配属されたら指導教員にねだると買ってくれるかも。久保田ゼミに配属された場合は4年前期くらいのタイミングでゼミ生全員に買い与えます。

- TeX と LaTeX は本来は区別されるべきものであるが初心者はあまり気にしなくて良いと思う。
- 違いを知りたい人は例えば次のサイトを参考のこと。

<https://texwiki.texjp.org/LaTeX>

# 最終課題

学籍番号と名前を書く

- 以上で本講義「プログラミング」の内容は終わり。
- 最終課題は第1回の授業でアナウンスした通り「数学の問題を自動生成し、『答え』ボタンをクリックすると答えが表示されるWebページを開発する」である。
- 問題を自動生成するページを作成し、その下に「制御構文の使用状況」と「自由記述欄」を記入する。自由記述欄ではプログラムの工夫した点や苦勞した点を書くこと。

2230999 久保田匠

逆行列

行列  $\begin{bmatrix} 0 & 3 & 1 \\ -4 & -3 & -2 \\ 1 & 2 & 1 \end{bmatrix}$  の逆行列は  $\begin{bmatrix} 1 & -1 & -3 \\ 2 & -1 & -4 \\ -5 & 3 & 12 \end{bmatrix}$  である。

制御構文の使用状況

この文章は  
書かなくてよい

以下の表において、自身で記述したコード（授業資料等の単純なコピーを除く）で使用したものは「○」を、使用していないものには「×」をつけよ。for文はwhile文と読み替えてもよい。例えば、課題の中でwhile文を一度でも使ったらfor文に「○」をつけてよい。また、「for文の中にif文」が「○」である場合は、「if文」にも「for文」にも「○」をつけてよい。

if文	for文	for文の中にif文	二重for文	二重for文の中にif文
○	○	×	○	×

自由記述欄

私がこのページを作るにあたり、工夫した点は○○である。プログラムの作成にあたり、△△という困難にあたったが、while文の条件を○○のように書き換えることで△△をうまく乗り越えることができた。←もちろんこれは少なすぎるのもっとたくさん書く。

```
<!-- 制御構文の使用状況の表（コピペ用） -->
<table border="1" style="text-align: center;">
<tr>
<th>if文</th><th>for文</th><th>for文の中にif文</th><th>二重for文
</th><th>二重for文の中にif文</th>
</tr>
<tr>
<td>○</td><td>○</td><td>×</td><td>○</td><td>×</td>
</tr>
</table>
```

# 最終課題（提出について）

- 2月23日（月）の23時59分までに以下のアドレスに html ファイルをメールで提出。

skubota@aecc.aichi-edu.ac.jp

- 提出物のファイル名は「学籍番号.html」にすること。
  - 例えば、学籍番号が「2230999」である学生は、提出ファイル名を「2230999.html」にする。
- ファイル管理の都合上、上記が守れていないと未提出扱いになる可能性があるので要注意。
- 提出が遅れた場合、受理はするが大幅に減点。

```
71 <h1>2024プログラミング採点</h1>
72 <h2>金曜3限クラス（義務数学1）</h2>
73 <p>
74   <script>
75     let counter = 0;
76     for(let i = 0; i<Fri3Member.length; i++){
77       document.write("<a href="+ Fri3Member[i] + ".html"> " + Fri3Member[i] + "</a> ");
78       counter++;
79       if(counter % 5 == 0){
80         document.write("<br>");
81       }
82     }
83   </script>
84 </p>
```

履修者の学籍番号の配列から  
採点用のサイトを構築できる



2024プログラミング採点

金曜3限クラス（義務数学1）

念の為、非公開

# 最終課題（詳細）

1次方程式の問題は  
授業で扱ったのでNG

## ■ 必須事項

- ページを再読み込みするごとに違う問題が生成されること。
- 答えボタンをクリックして答えが表示されること。

## ■ 加点事項

- if文 や for文（while文） を使っている。二重 for文ならさらに加点。
- TeX を使っている。
- ページを見やすくする工夫をしている。
- 数学であれプログラミングであれ難しい題材に挑戦している。
- 生成された問題に応じて解法（解き方）も作成されている。
- 自由記述欄は文章量が多ければ多いほど高得点とするが、無駄な文章や量を稼いでいると判断できる箇所は0文字としてカウント。

加点事項に取り組んだら  
自由記述欄に記入すること

## ■ 単位取得の基準

- 必須事項をクリアしていれば基本的には OK とするが、明らかに手を抜いている（自由記述欄が数十文字など）と判断されるものは再提出または不可とする。
- 動かないプログラムは0点。

# 最終課題（減点事項）

- 次の構文や記法を使用した場合は大きく減点する。使用状況によっては単位取得を認めない。
  - 三項演算子
  - テンプレートリテラル
  - 厳密等価演算子
  - 分割代入
  - アロー関数
- これらは本課題では扱う必要のない構文であり、理解せずに生成AIの出力をコピーした場合に混入しやすいため、原則として使用を認めない。
- ただし、最終課題の完成度が十分高いと判断される場合は「生成AIを有効活用している」（または「独学で高い水準まで学んでいる」とみなし、特に減点しない。
- 生成AIを使用する場合は、出力結果を鵜呑みにせず、自分が理解し、説明できる範囲でコードを書くこと。
  - 特に、プログラミングに苦手意識のある学生が生成AIを使用する場合は、慎重に使用すること。

独学で学んでいる人には  
申し訳ないけど...

# 問題を生成するときのアイデア①

- 「答え」を先に生成し、答えを材料に「問題」を作る。

## 連立方程式（解が一意に決まる）

連立方程式

$$\begin{cases} -5x - 3y + 3z = -29 \\ -4x - 5y + 3z = -23 \\ -x - 2y + z = -6 \end{cases}$$

を解け。

答え

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ -4 \end{bmatrix}$$

- 例えば連立方程式の問題は、係数や右辺を先に生成すると、基本的には汚い分数が答えになる。
- 最初に答えを整数値で生成しておくと、答えが必ず整数になる連立方程式が生成できる。

# 問題を生成するときのアイデア②

- ①の応用。「問題」でも「答え」でもないものを生成して、それをもとに「問題」と「答え」を作る。

**逆行列**

掃き出し計算で逆行列を計算する練習問題なので、分数が出ないようにしている

行列  $\begin{bmatrix} 1 & 2 & -2 \\ 2 & -3 & 2 \\ 2 & -2 & 1 \end{bmatrix}$  の逆行列は  $\begin{bmatrix} 1 & 2 & -2 \\ 2 & 5 & -6 \\ 2 & 6 & -7 \end{bmatrix}$  である。

- 逆行列の問題は「問題」「答え」の片方をランダムに生成すると、もう片方は基本的に汚い行列が出る。
- この問題では、対角成分がすべて1である上三角行列Pと下三角行列Qを生成している。
- 実は  $P^{-1}$  と  $Q^{-1}$  は簡単に計算できるので、問題の行列を  $PQ$  にして、答えの行列を  $Q^{-1}P^{-1}$  としている。

$$(PQ)^{-1} = Q^{-1}P^{-1}$$

$$\begin{bmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{bmatrix}$$

対角成分が1なので  
分数が出てこない

# 問題を生成するときのアイデア③

- 都合の悪い数値を生成したら生成をやりなおす。
- 都合の良い数値を生成するまで while文 で繰り返す。

## 1次合同方程式(1)

合同方程式  $ax \equiv b \pmod{m}$  は  $a$  と  $m$  が互いに素であるときただひとつの解をもつ。

[1] 合同方程式  $11x \equiv 9 \pmod{13}$  を解くと  $x \equiv 2 \pmod{13}$  である。

答え

[2] 合同方程式  $14x \equiv 15 \pmod{17}$  を解くと  $x \equiv 12 \pmod{17}$  である。

答え

- $p$  を素数とする。
- 合同方程式  $ax \equiv b \pmod{p}$  において、 $a$  と  $b$  を生成した結果、偶然  $a$  と  $b$  が等しくなると問題として面白くない。
- これを回避するため、 $a$  と  $b$  が異なる数を生成するまで  $b$  を生成しなおす。

```
let a = Math.floor(Math.random()*11) + 2;
let b = Math.floor(Math.random()*11) + 2;
while(a == b){
  b = Math.floor(Math.random()*11) + 2;
}
```

# [演習]最終課題に取り組もう

- 既に構想がある人は課題に着手してください。
- 何を作ればいいのか悩んでいる人は、練習として、簡単な問題を題材にして課題に取り組んでみるといいかも。
  - 一度作ってみると勝手が分かってくる。
- 課題の難易度は題材に大きく依存するので、すでにプログラミングに苦手意識のある学生は、どのような題材を選ぶべきかよく吟味すること。