プログラミング

第14回 数式の表示(TeXについて)

久保田 匠



5限

2





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)		<u>Prog_03-1</u> 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	●, ★	<u>Prog_11-1</u>
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	学習アプリを開発してみよう		課題提出例

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"</pre> crossorigin="anonymous"></script> <script defer src="https://cdn.jsdelivr.net/npm/katex/dist/contrib/auto-</pre> render.min.js" crossorigin="anonymous"></script> <script> window.onload = function() { renderMathInElement(document.body, {delimiters: [{left: "\\[", right: "\\]", display: true}, {left: "\$", right: "\$", display: false}]}); }; コピー

今日は「Prog_14-1」を

選択してください。





- 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」。

🗯 Code	e ファイル 編集 選択 表示	移動 実行	ターミナル	ウィンドウ	ヘルプ									
•••	新しいテキスト ファイル	ЖN	$\leftarrow \rightarrow$				Code	ファイル 編集	観 選択	表示	移動	実行	ターミナル	ウィンド
~ ×	新しいファイル	^~% N					•	新しいテキスト	ファイル			ЖN	$\leftarrow \rightarrow$	
	新しいウィンドウ	合 光 N						新しいファイル.			~	₹₩N		
\circ	開く	жO					· · · ·	新しいウィンド	ל			ሰ 		
	フォルダーを開く						1	New Window v	vith Profile	<u>,</u>		>		
የቀ	ファイルでワークスペースを開く						3	開く				жΟ		
5	最近使用した項目を開く	>	io Co	de			4	フォルダーを開く	<					
	フォルダーをワークスペースに追加.						5	ファイルでワーク	クスペース	を開く			/title>	
æ	名前を付けてワークスペースを保存.						7	最近使用した項目	目を開く			>		
но П	ワークスペースを複製						8	フォルダーをワ-	-クスペー.	スに追加.				
ш	保存	жS					9 10	名前を付けてワ-	-クスペー.	スを保存.				
	名前を付けて保存	企					11	ワークスペースを	を複製					
	すべて保存	₹₩ S					12 13	保存				₩S		
	×	🚺 ようこそ	↔ DO0</p	CTYPE html> U	ntitled-1 •			名前を付けて保ィ	字			<mark>ሰ</mark> ж s		
		1 DOCT</td <td>YPE html></td> <td></td> <td></td> <td></td> <td></td> <td>すべて保存</td> <td></td> <td></td> <td>-</td> <td>ΣЖS</td> <td></td> <td></td>	YPE html>					すべて保存			-	ΣЖS		
		2 <html></html> 3						共有				>		
		4 <head></head> 5 <	eta charset	="UTF-8">				自動保存						
			itle>ここに^	ページのタイトル	を書きます <td>></td> <td></td> <td>ファイルを元に</td> <td>戻す</td> <td></td> <td></td> <td></td> <td></td> <td></td>	>		ファイルを元に	戻す					
		7 <td></td> <td></td> <td></td> <td></td> <td></td> <td>エディターを閉し</td> <td>じる</td> <td></td> <td></td> <td>жw</td> <td></td> <td></td>						エディターを閉し	じる			жw		
	*	9 <body></body>						ウィンドウを閉し	じる			ΩжW		
	Ctrl + V	10 E.	んにちは!						_	-	-	_		
		11 12												
		13 <td></td>												





- 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」。









昨日

C

Prog_01-1.html

- 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。



ファイル /Users/s

6

 $a = -2 \text{ obs} a^2 = 4 \text{ coss}$ ○ 検索 IB Google カレンダー × S 授業関連のページ (× S ここにページのタイ × 000 + 制限モードは、安全なコード参照のためのものです。すべての機能を有効にするには、このウィ... 管理 詳細情報 ファイル /Users/shokubota/Documents/授業支援サイト/Pro. Prog_01-1.html × Ch こんにちは! Users > shokubota > Documents > 授業支援サイト > ◇ Prog_01-1.html > ◇ html 私の名前は久保田匠です。 <!DOCTYPE html> <meta charset="UTF-8"> <title>ここにページのタイトルを書きます</title> </head> こんにちは!

br>私の名前は久保田匠です。 8 ⑦ 制限モード ⊗ 0 △ 0 ㎏ 0 行 13、列 8 スペース: 4 UTF-8 CRLF HTML 🗘

[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。



[復習]イベント



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、ユーザーの特定の操作(ボタンをクリックする等)やブラウザの特定の動作(ページが読み込まれる等)に応じて発生する信号のこと。
- イベントハンドラとは、発生したイベントに応じて実行される関数のこと。
- 上にある「ボタンをクリックすると答えが表示されるweb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。





1 2 4 5 6 7	<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"/> <title>Prog 12-1</title> <th>右にボタンを用意します。 このボタンを押すと ここの文子の色が 変わります。</th></head></html></pre>	右にボタンを用意します。 このボタンを押すと ここの文子の色が 変わります。
8 9 10 11 12 13 14	<pre>function changeColor(){ document.getElementById("001").style.color = "red"; } </pre>	
15 16 17 18	右にボタンを用意します。 <button onclick="changeColor()" type="button">ボタン</button> このボタンを押すと ここの文字の色が 	右にボタンを用意します。 ボタン このボタンを押すと <mark>ここの文字の色が</mark> 変わります。
19 20 21 22 23	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を押すと関数 changeColor が呼び出され る。
 - 関数 changeColor は id が 001 である タグ内の文章の
 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

- ここまでの知識を応用することで「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プログラム」が書ける。
- 次は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である。



※ head部はさっきのプログラムと同じ



1次方程式 4x=-6 の解は x=-1.5 である。 答えを表示する

- 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。
- したがって基本的には document.write(b + "/" + a);のように命 令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、前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



- 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 は卒論を書くときにも使用する可能性があるので、この 機会に雰囲気を掴んでおこう。



- 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

- 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-)

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

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

Word による数式の表示 (最近は Word でもきれいかも)

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 を入力しておくと 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 に次を入力してブラウザで閲覧してみよう。

```
<!DOCTYPE html>
<head>
             <meta charset="UTF-8">
             <title>Prog_13-1</title>
             <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></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></sc
             <script>
                           window.onload = function() {
                           renderMathInElement(document.body, {delimiters: [
                                        {left: "\\[", right: "\\]", display: true},
                                        {left: "$", right: "$", display: false}
                           }__
             </script>
</head>
<bodv>
                            多項式 $x^2 + 2x + 1$ を因数分解すると $(x+1)^2$ になります。
             </body>
                                                                                                       多項式 x^2 + 2x + 1 を因数分解すると (x + 1)^2 になります。
 </html>
```

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\$]
- \[と\]で囲む(円マークとバックスラッシュは同じ)。
- 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}]
- head部に次を入力してブラウザで閲覧してみよう。

21	<body></body>	
22		
23	多項式 \$x^2 + 2x + 1\$ を因数分解すると \$(x+1)^2\$ になります。	ふの前後は半角スペースを
24		
25		あけることを推奨
26	方程式	
27	$\left[x^2 - x - 2 = 0 \right]$	
28	を解くと \$x = −1, 2\$ です。	
29		
30		

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

- 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

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

で与えられます。

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

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

- 1次方程式 6x = 10の解は 以下は、1次方程式を自動生成するプロ $x=rac{1}{3}$ 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。入力してみよう。 である。 答えを表示する
- ただし、scriptタグ内では \ は別の意味を持っている。数式 を入力するときに「\」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。





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』



- 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



■ 以上で本講義「プログラミング」の実質的な内容は終わり。

- 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、その下にプログラムの 工夫した点や苦労した点を書くこと。工夫した点では着想の 経緯、苦労した点ではそれをどう乗り越えたかなど、頑張っ たアピールを自由に記述すること。



最終課題(詳細)

- 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
 - 来週に詳細を説明。
- 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 だが明らかに手を抜い ている(自由記述欄が数十文字とか)と判断されるものは再提出ま たは不可。
 - 動かないプログラムは0点。

最終課題(提出について)

 2月23日(日)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@auecc.aichi-edu.ac.jp

- 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」の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。



- 次回の授業では、問題を生成するときのちょっとしたコツや、 成績評価における加点事項や採点基準の詳細を紹介。
- 演習の時間を多めにとる予定。