

【オンライン無料体験会】 MATLAB を使い始めるための入門ハンズオン

2026/02/26 14:00-16:00

MathWorks Japan

アプリケーションエンジニアリング部 テクニカルコンピューティング

齊藤 甲次郎 井上 道雄

アジェンダ

- 事前準備
- MATLAB 概要
- デスクトップの基礎
 - ウォーミングアップ MATLAB で計算
 - 演習 1 MATLAB の画面に慣れる
- エディターとライブエディター
 - 演習 2 ファイルの読み込み、可視化
 - スクリプトの関数化
- MATLAB Copilot の活用
 - 演習 3 MATLAB Copilot にコードを書いてもらう
- 手軽にGUIアプリ作成
 - 演習 4 GUI アプリの作成
 - GUI アプリ作成のコツ
- 学習リソースの紹介

事前準備

事前準備

ワークショップライセンスの確認

1. <https://www.mathworks.com/licensecenter/classroom/5182050/> にアクセス (画像1)
2. 「サインイン」をクリックして、MathWorks アカウントでサインイン (画像2)
アカウントをお持ちでない方は左下の「アカウントを作成する」から作成してください。



画像1



画像2

事前準備

ワークショップライセンスの確認

3. MathWorks アカウントに登録されているメールアドレスにMATLAB Copilot ライセンスがリンクされたというお知らせが届きます (画像3)



画像3

事前準備

ワークショップライセンスの確認

4. 「MATLAB Online へのアクセス」をクリックします (画像4)



MATLAB ヘルプ センター コミュニティ 学習

Hands-On Workshop 向け MATLAB へのアクセス

MathWorks では、Hands-On Workshop でお使いいただく特別ライセンスをコース受講者に提供しています。この限定ライセンスは 14 3月 2026 まで利用可能であり、コースの課題にのみご使用いただくことを目的としています。官公庁、研究機関、民間企業、その他の組織で使用することはできません。

コース名:	2/26 MATLAB入門ハンズオン【オンライン無料体験会】
所属団体:	MathWorks
開始:	26 2月 2026
終了:	27 2月 2026

MATLAB Online へのアクセス

画像4

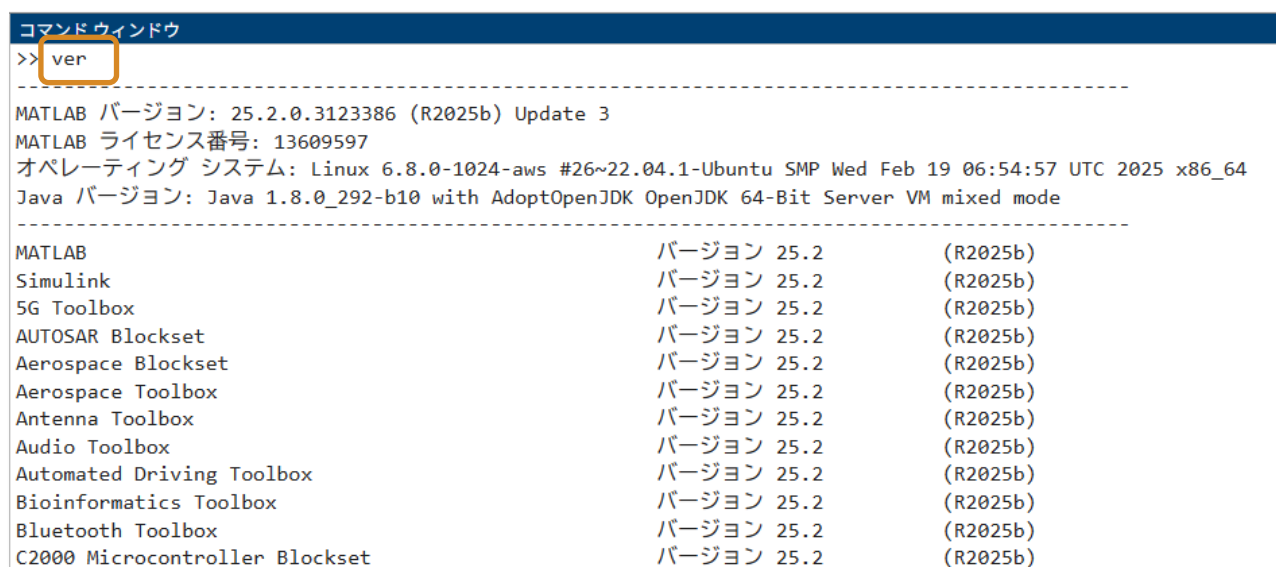
事前準備

ワークショップライセンスの確認

5. 「MATLAB Online を開く」 ボタンをクリックします (画像5)
6. コマンドウィンドウに「ver」と入力してエンターキーを実行します。画像6 のような表示になればワークショップライセンスの取得は問題ありません



画像5



画像6

ワークショップライセンスについて

環境

MATLAB Online のみ

使用できる製品

MATLAB
MATLAB Online で使用できる全Toolbox
MATLAB Copilot

有効期限

3月12日(木)まで



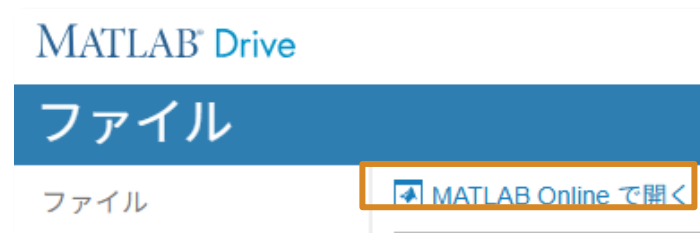
事前準備

ファイルのコピー

7. Web ブラウザーでリンク (<https://drive.mathworks.com/sharing/50dba5ca-9125-4cdc-ba50-31ab2848832c>) を開き「自分のファイルに追加」→「フォルダーのコピー」をクリックします (画像7)
8. フォルダーがコピーされます。「**MATLAB Online** で開く」をクリックします (画像8)

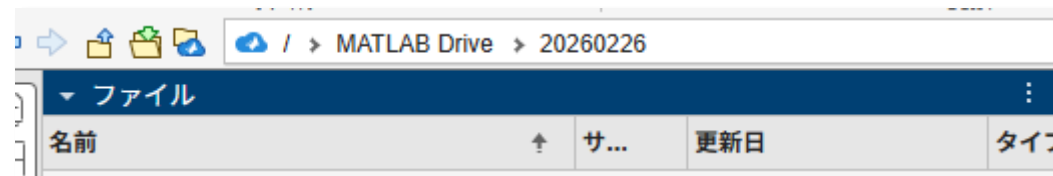


画像7



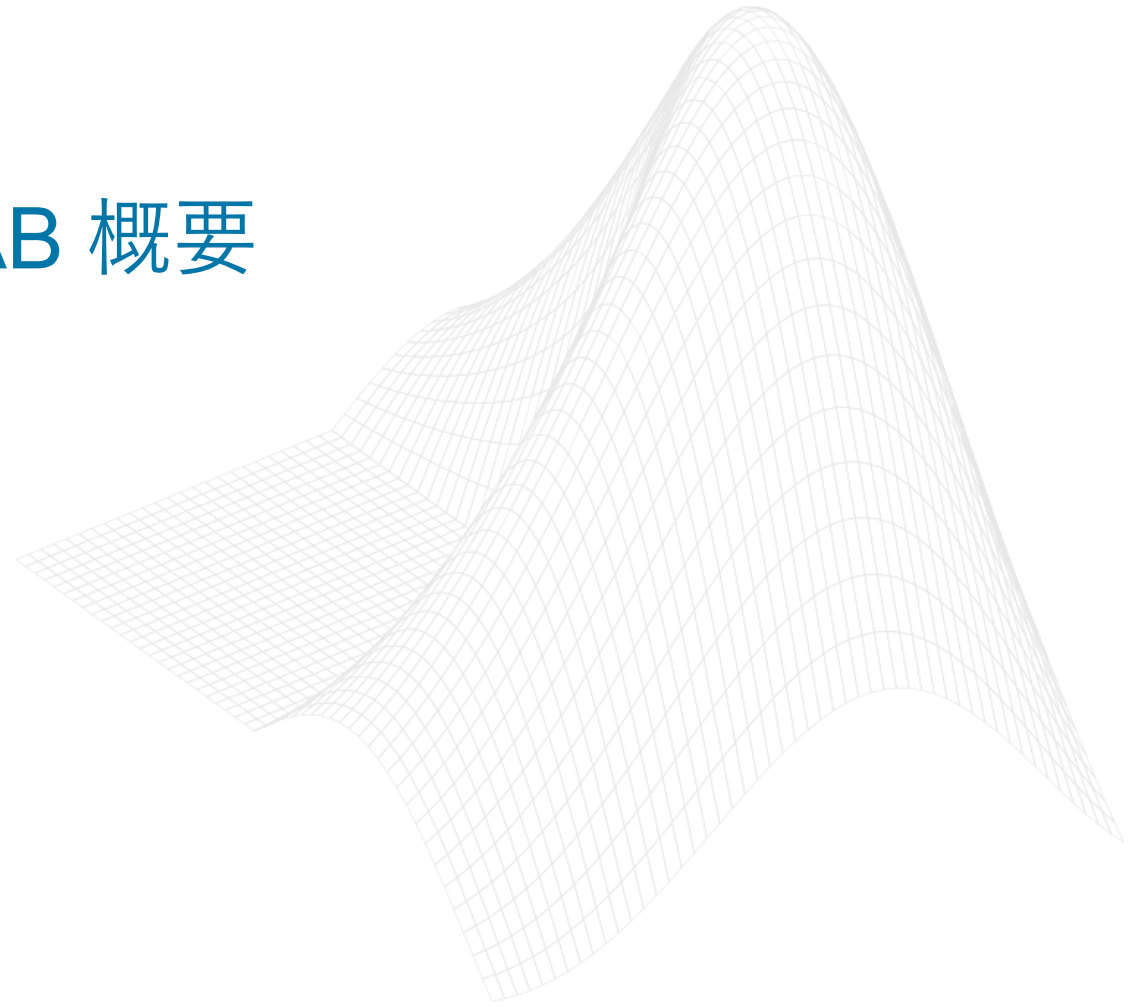
画像8

9. MATLAB Online で「**20260226**」のフォルダーが表示されれば準備完了です。ダブルクリックして現在のフォルダーを「**20260226**」に移動します (画像9)



画像9

MATLAB 概要



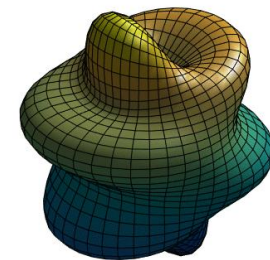
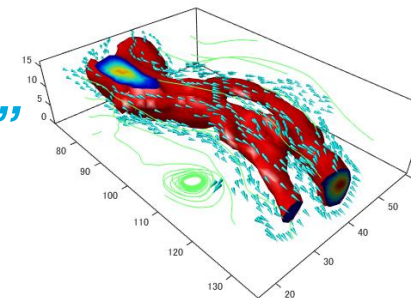
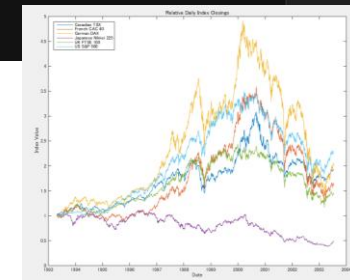
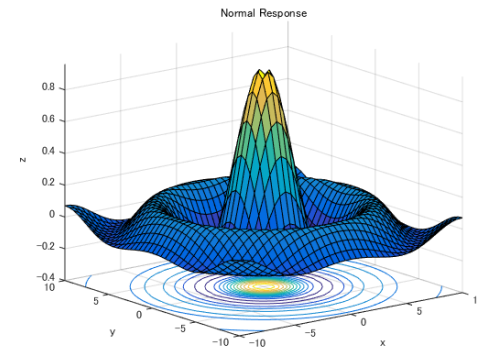
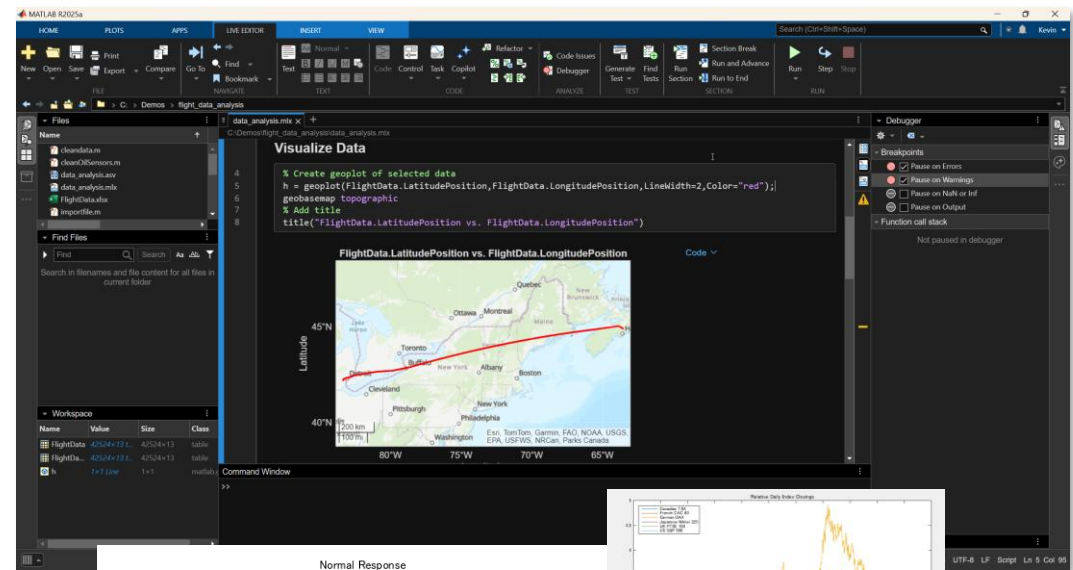
MATLAB

Matrix Laboratory

- 対話的な開発環境
- 豊富な拡張ライブラリ
- 使用目的
 - 数値解析
 - データ解析、可視化
 - アルゴリズム開発、プログラミング
 - 外部アプリケーションとの連携

“Accelerating the pace of engineering and science”

本業に専念できるよう、プログラミングを楽に

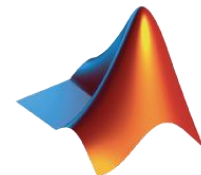


MATLAB の標準機能



120 以上のToolbox

MATLAB [®] 製品ファミリ	SIMULINK [®] 製品ファミリ
MATLAB	Simulink
並列処理	イベントベースモデリング
Parallel Computing Toolbox MATLAB Parallel Server	Stateflow SimEvents
AI、データサイエンス、統計	物理モデリング
Deep Learning Toolbox Statistics and Machine Learning Toolbox Curve Fitting Toolbox Text Analytics Toolbox	Simscape Simscape Battery Simscape Driveline Simscape Electrical Simscape Fluids Simscape Multibody
数学および最適化	リアルタイム シミュレーションおよびテスト
Optimization Toolbox Global Optimization Toolbox Symbolic Math Toolbox Mapping Toolbox Partial Differential Equation Toolbox	Simulink Real-Time Simulink Desktop Real-Time
レポートとデータベースアクセス	レポート作成
Database Toolbox MATLAB Report Generator	Simulink Report Generator
コード生成	システム エンジニアリング
MATLAB Coder Embedded Coder HDL Coder HDL Verifier Filter Design HDL Coder Fixed-Point Designer	System Composer Requirements Toolbox
	コード生成
	Simulink Coder Embedded Coder DDS Blockset AUTOSAR Blockset G2000 Microcontroller Blockset



MATLAB



Simulink

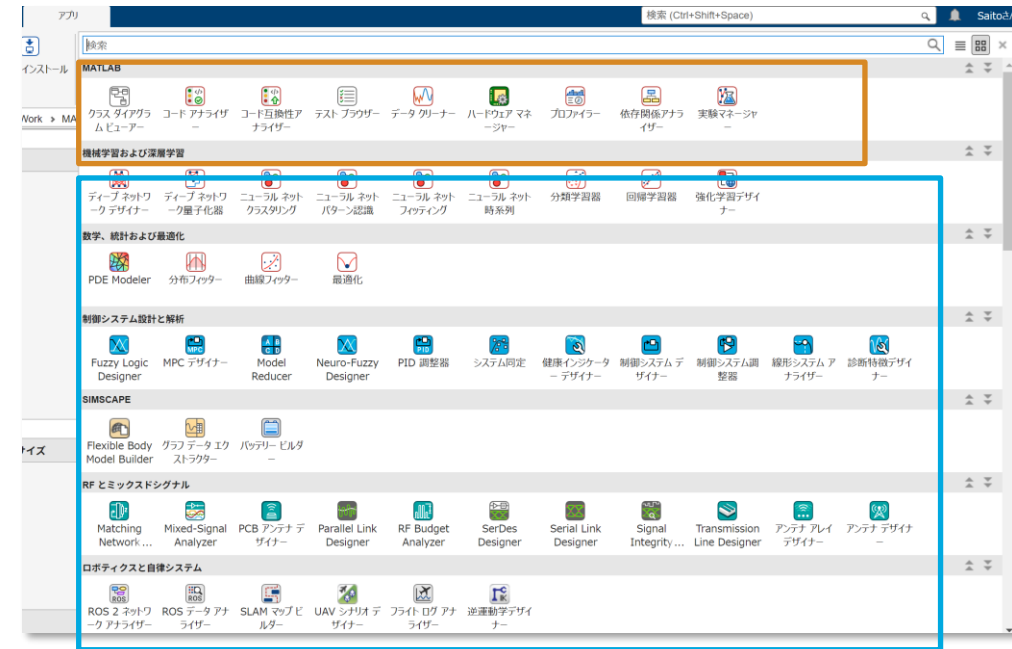
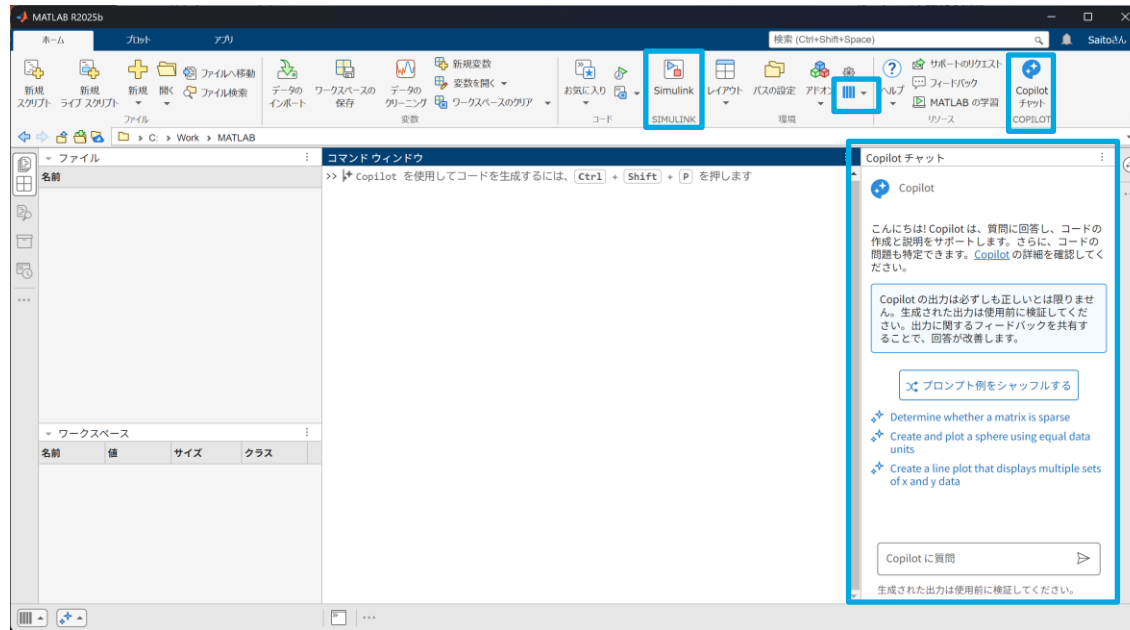


Toolbox

用途		
信号処理	RF、ミックスドシグナル	自動車
Signal Processing Toolbox DSP System Toolbox Audio Toolbox Wavelet Toolbox DSP HDL Toolbox	Antenna Toolbox RF Toolbox RF PCB Toolbox RF Blockset Mixed-Signal Blockset SerDes Toolbox Signal Integrity Toolbox	Model-Based Calibration Toolbox Powertrain Blockset Vehicle Dynamics Blockset Automated Driving Toolbox IEC Certification Kit (for ISO 26262 61508) Vehicle Network Toolbox AUTOSAR Blockset RoadRunner RoadRunner Asset Library RoadRunner Scenario RoadRunner Scene Builder Simulink 3D Animation
画像処理およびコンピューター ビジョン	無線通信	
Image Processing Toolbox Computer Vision Toolbox Lidar Toolbox Medical Imaging Toolbox Vision HDL Toolbox Image Acquisition Toolbox	Communications Toolbox 5G Toolbox LTE Toolbox WLAN Toolbox Bluetooth Toolbox Satellite Communications Toolbox Wireless HDL Toolbox Wireless Testbench	
制御システム	レーダー	航空宇宙
Control System Toolbox System Identification Toolbox Predictive Maintenance Toolbox Robust Control Toolbox Model Predictive Control Toolbox Fuzzy Logic Toolbox	Radar Toolbox Phased Array System Toolbox	Aerospace Blockset Aerospace Toolbox UAV Toolbox DO Qualification Kit (for DO-178) Simulink 3D Animation
		金融工学

<https://jp.mathworks.com/products.html>

120 以上のToolbox



MATLAB 本体だけでできること

- ファイル入出力 `readtable`, `writetable`, `imread`, `imwrite` など
- 三角関数、行列演算 `sin`, `cos`, `ones`, `sum` など
- カスタムGUI アプリ作成 `appdesigner`, `uifigure` など
- グラフィックス `plot`, `scatter`, `bubblechart`, `geoplot` など
- 単体テスト `assert`, `functiontests` など
- 他言語連携 `Python`, `C/C++`, `.NET`, `Java`, `Fortran`

Toolbox があるとできること

- 専門的な関数(コマンド)
- GUI アプリ (Simulink も)

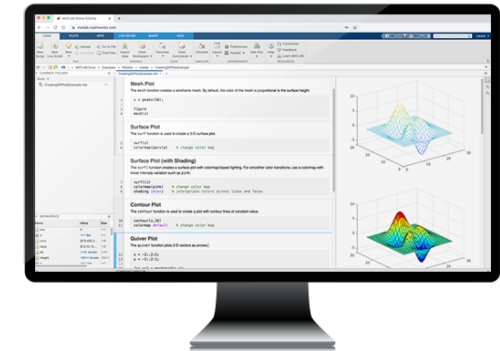
MATLAB の利便性を飛躍的に高めるアクセス・サービス



MATLAB デスクトップ



MATLAB Drive
クラウドストレージを利用し
どこからでもファイル共有

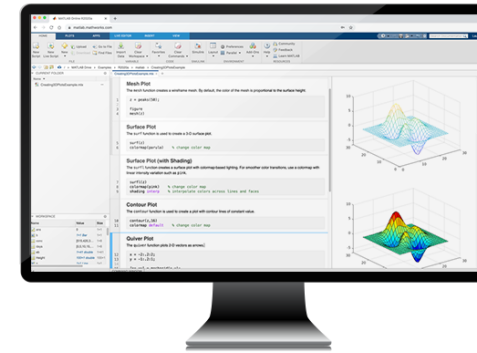


MATLAB Online
Web ブラウザで
MATLAB およびSimulink の利用



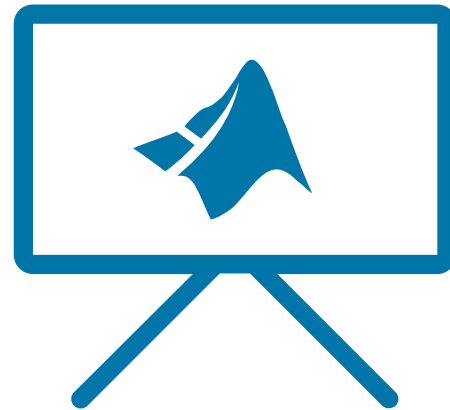
MATLAB Mobile
iPad / iPhone / Android デバイスで
MATLAB 利用

MATLAB と MATLAB Online の違い



	MATLAB	MATLAB Online
インストールの必要	マシンにインストール	インストール不要。Web ブラウザでアクセスして使う
バージョン	任意のバージョン	最新版 (現時点ではR2025b Update 4)
OS	インストールしたマシンのOS (Windows, macOS, Linux)	Linux (Ubuntu)
実行環境	インストールしたマシン	MathWorks のクラウド環境
ファイルアクセス	<ul style="list-style-type: none"> ローカルフォルダー MATLAB Drive OneDrive 	<ul style="list-style-type: none"> MATLAB Drive OneDrive

デスクトップの基礎



タブ

エディター

Figure ウィンドウ

現在のフォルダー

ワークスペース
(変数を格納するところ)

コマンドウィンドウ
(コマンドを1つずつ実行するところ)

Figure 1

コマンドウィンドウ

```
>> I = imread('coins.png');
>> figure; imshow(I);
>>
```

ワークスペース

名前	値	サイズ	クラス
ans	2.3819e+03	1×1	double
areas	[2724;1917;...	10×1	double
I	246×300 ui...	246×300	uint8
I1	246×300×3 ...	246×300×3	uint8
I2	246×300×3 ...	246×300×3	uint8
I3	246×300×3 ...	246×300×3	uint8
lbw	246×300 lo...	246×300	logical
lbwc	246×300 lo...	246×300	logical
lbwf	246×300 lo...	246×300	logical
stats	10×4 table	10×4	table

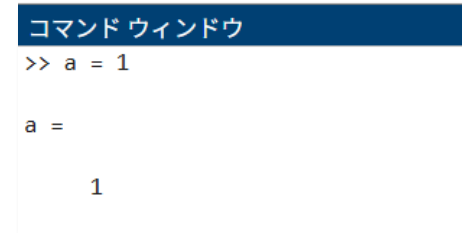
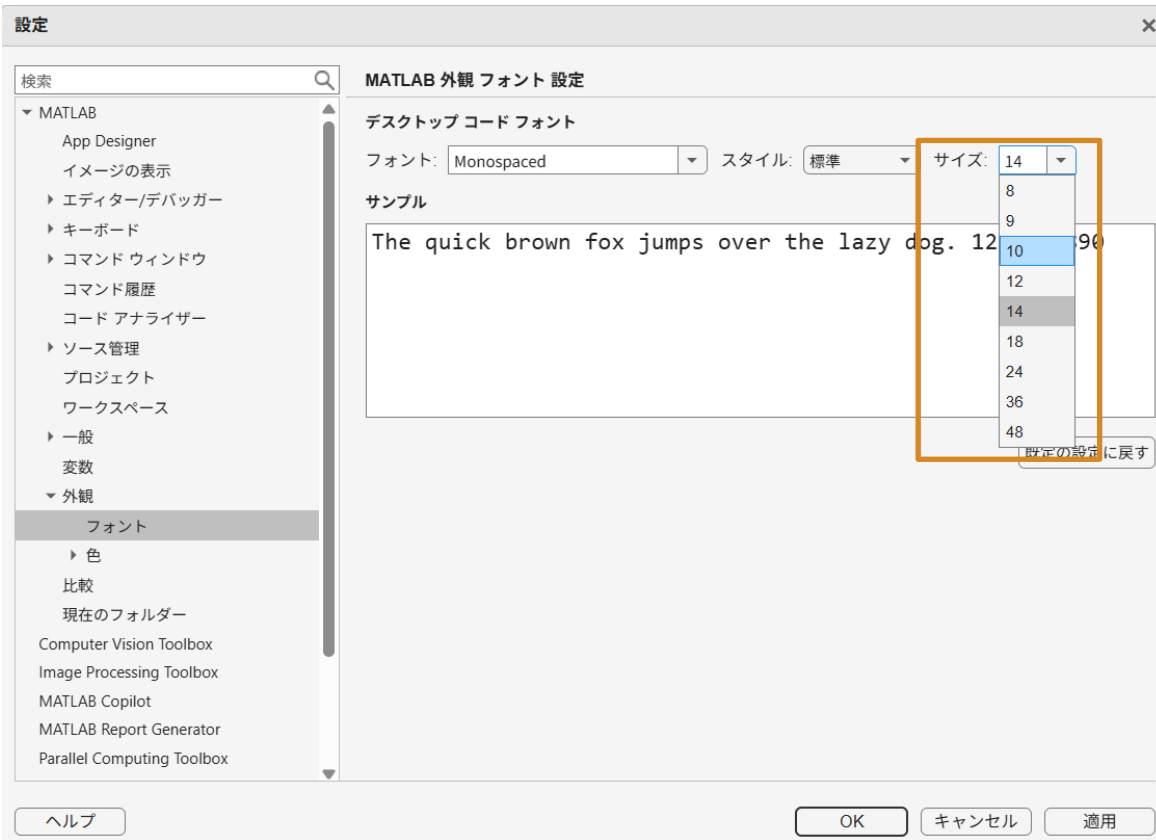
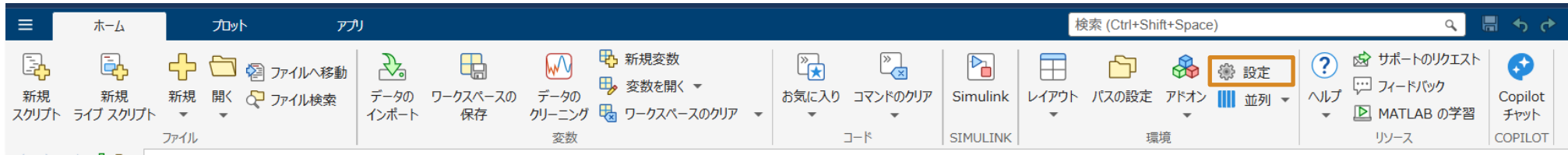
オブジェクトの定量評価 (面積・中心点・直径・周囲長)

```
1 clear;clc;close all;imtool close all
2 I = imread('coins.png'); % 画像の読み込み
3 figure; imshow(I); % 表示
```

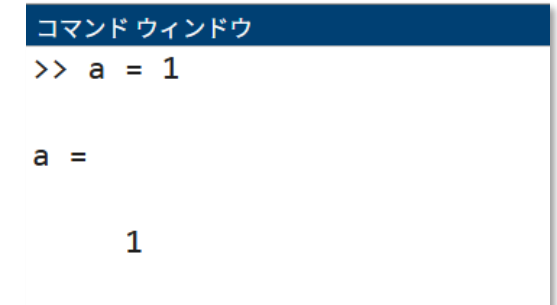
ズーム: 100% UTF-8 LF スクリプト 行 22 列 15

【設定】 フォントサイズを変更する

フォントが小さいと感じたら、設定からサイズを変更できます



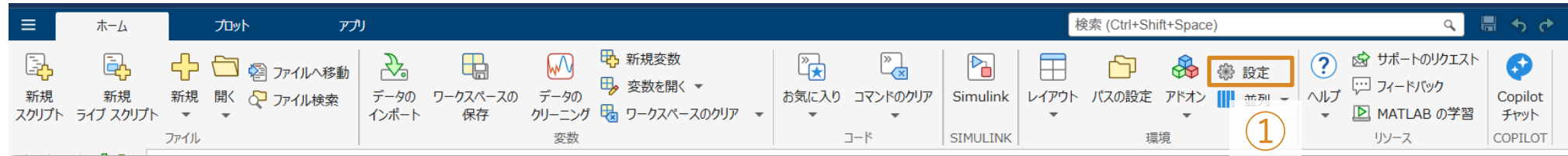
フォントサイズ10 pt



フォントサイズ14 pt

【設定】 MATLAB Copilot をOFF にする

前半はMATLAB Copilot を使用しないので設定からOFF に変更します



【ウォーミングアップ】 MATLAB で計算

- コマンドウィンドウでa からf を計算してみましょう

$a = 1$

$b = 2$

$c = a + b$

🚩 コード例

$a = 1$

$b = 2$

$c = a + b$

$d = 1.2$

$e = 1.6$

$f = d + e$

🚩 コード例

$d = 1.2$

$e = 1.6$

$f = d + e$

MATLAB では数値がデフォルトでdouble 型で定義されるため、データ型の宣言が不要です

【ウォーミングアップ】 MATLAB で計算

- コマンドウィンドウで2行2列の配列matA を作成してみましょう

$$matA = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

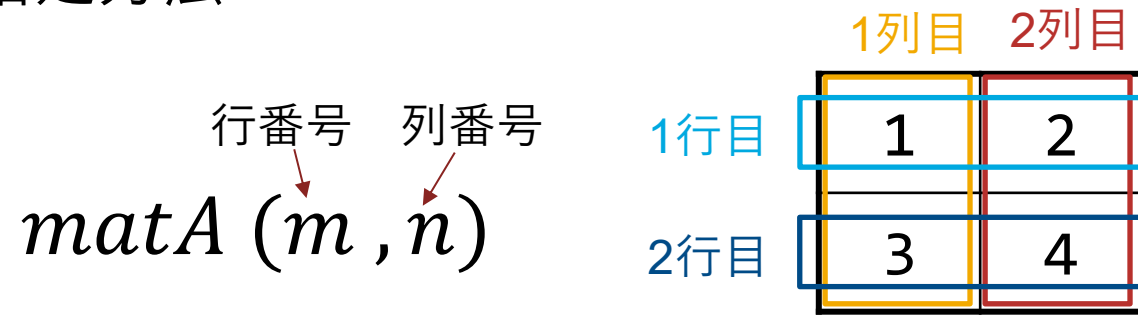
🚩 コード例

```
matA = [1, 2; 3, 4]
```

大括弧[]で配列を定義できます
セミコロン;で行の区切りを入れます

【ウォーミングアップ】 MATLAB で計算

- 要素の指定方法



- 配列matA の2行1列の値を5 に変更してみましょう

コード例

```
matA(2, 1) = 5
```

演習 1 MATLAB の画面に慣れる

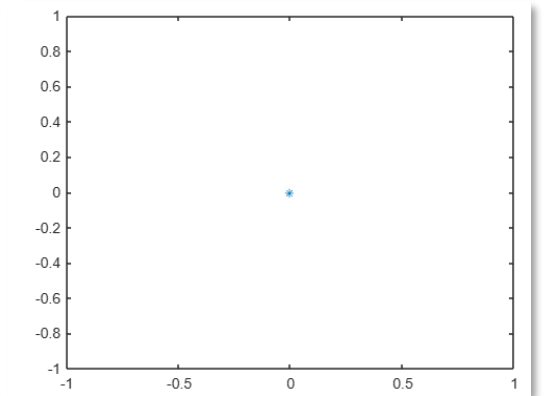
コマンドウィンドウ

コマンドウィンドウに書いて実行してみましょう

(1) x が0 の整数、
 y が $\sin\left(\frac{x}{2\pi}\right)$ の変数を定義し、
プロットしてみましょう

🚩 コード例

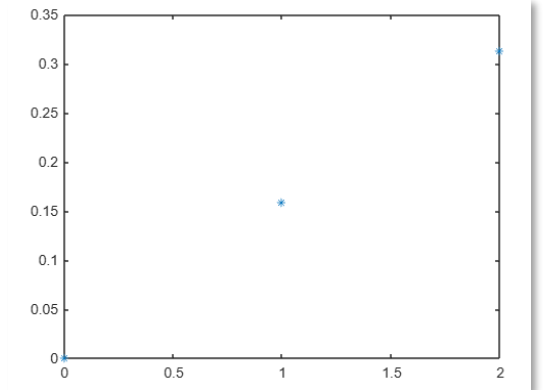
```
x = 0;  
y = sin(x/(2*pi));  
plot(x, y, '*')
```



(2) x が0, 1, 2 の整数、
 y が $\sin\left(\frac{x}{2\pi}\right)$ の変数を定義し、
プロットしてみましょう

🚩 コード例

```
x = [0 1 2];  
y = sin(x/(2*pi));  
plot(x, y, '*')
```



大括弧[]で配列を定義

演習 1 MATLAB の画面に慣れる

ワークスペース

MATLAB で作成した変数や、データファイルや他のプログラムからインポートした変数が表示

ワークスペース			
名前	値	サイズ	クラス
x	[0,1,2]	1×3	double
y	[0,0.1585,0.3130]	1×3	double

ダブルクリックすると値の詳細を確認や、編集、グラフ作成などができます

The image shows the MATLAB interface with the '変数' (Variables) tab selected. The workspace table on the left lists variables 'x' and 'y'. An orange arrow points from this table to the 'y' variable's details on the right. The details pane shows the variable 'y' as a 1x3 double array with the following values:

	1	2	3
1	0	0.1585	0.3130
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			

演習 1 MATLAB の画面に慣れる

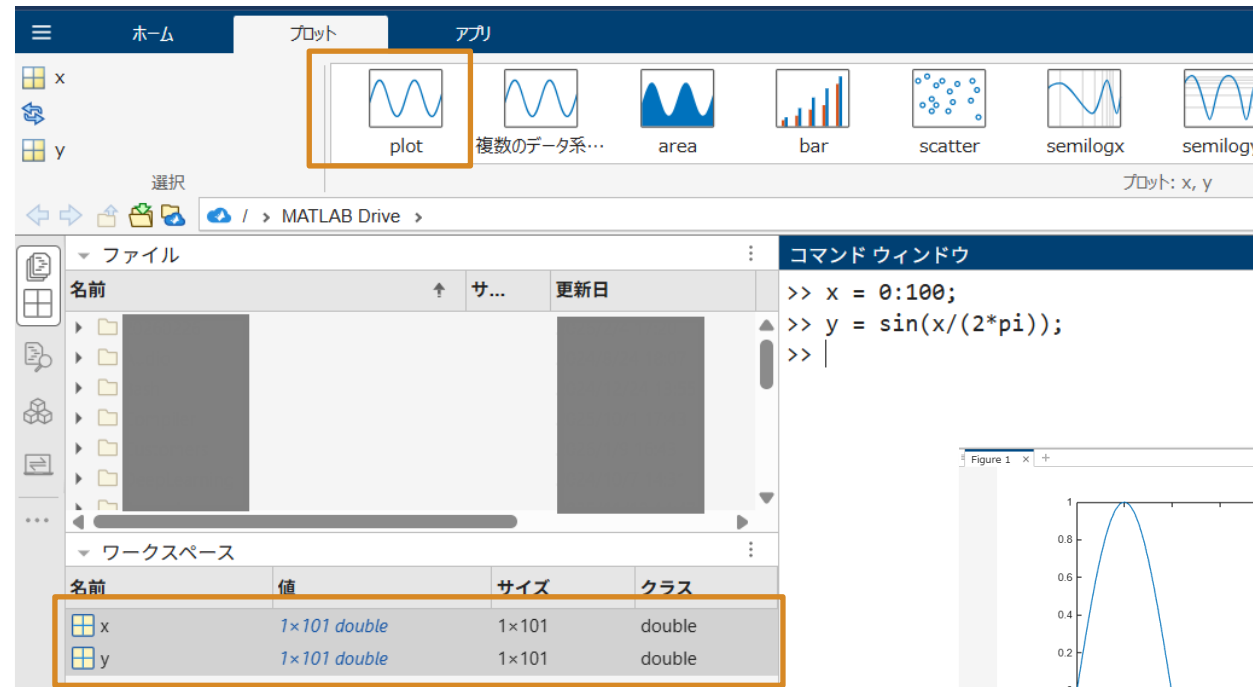
ワークスペース

(3) x が0, 1, 2, ..., 100の整数、 y が $\sin\left(\frac{x}{2\pi}\right)$ の変数を定義し、プロットしてみましょう

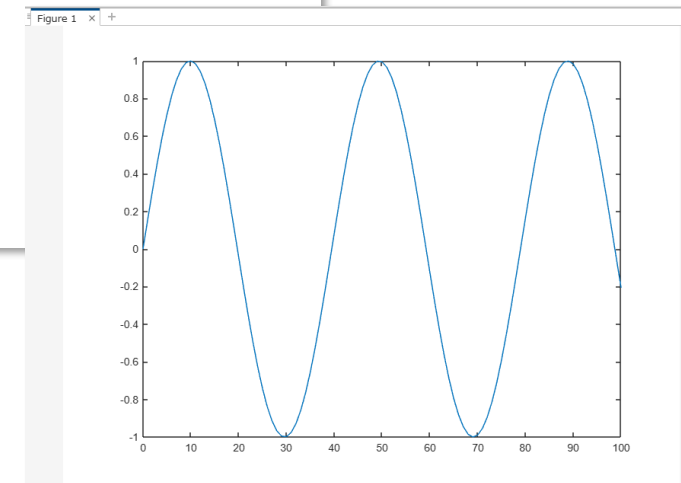
コード例

```
x = 0:100;  
y = sin(x/(2*pi));
```

コロン:を使って連続の整数を定義



ワークスペースでマウสดラッグで x と y を選択して、「プロット」タブから「plot」をクリック



演習 1 MATLAB の画面に慣れる ドキュメントを開く

(4) プロットの色を青から赤に変えてみましょう

コード例
doc **plot**

▼ LineSpec — ラインスタイル、マーカー、色
string スカラー | 文字ベクトル

ラインスタイル、マーカー、および色。記号を含む string スカラーまたは文字ベクトルとして指定します。記号は任意の順序で表記できます。3 つの特性
ロットはラインなしでマーカーのみを表示します。

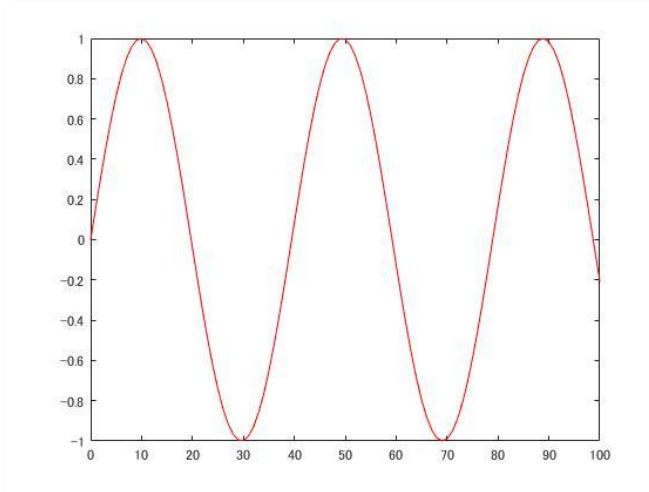
例: "--or" は円形マーカー付きの赤い破線です。

ラインスタイル	説明
"_"	実線
"--"	破線
"_."	点線
"-."	一点鎖線

マーカー	説明
"o"	円
"+"	プラス記号
"*"	アスタリスク
"."	点
"x"	十字
"_"	水平線
" "	垂直線
"square"	正方形
"diamond"	菱形
"^"	上向き三角形
"v"	下向き三角形
">"	右向き三角形
"<"	左向き三角形
"pentagram"	星形五角形
"hexagram"	星形六角形

色名	省略名	RGB 3 成分
"red"	"r"	[1 0 0]

コード例
`plot(x, y, 'r')`



ドキュメントはWeb ページからも閲覧できます
<https://jp.mathworks.com/help/>

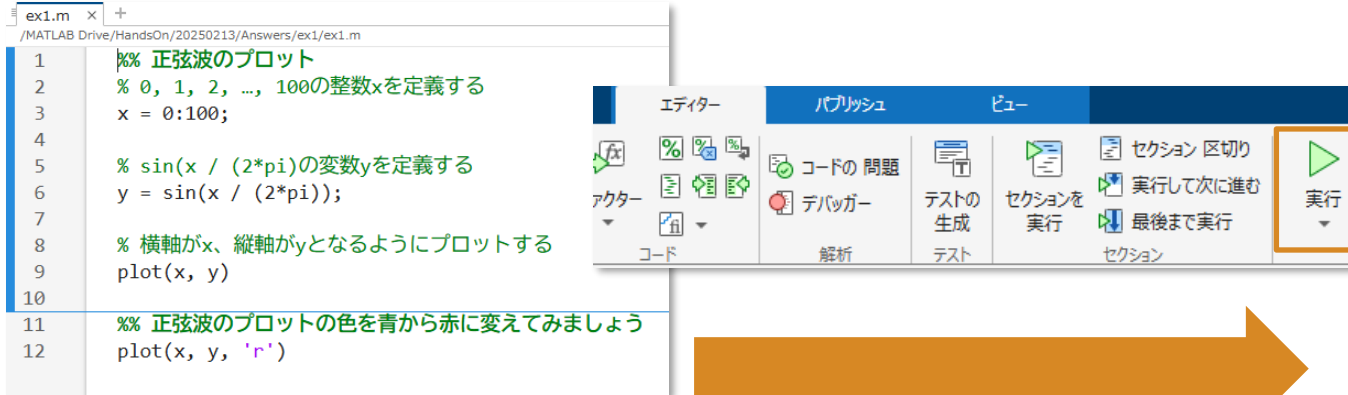
エディターとライブエディター



スクリプトとライブスクリプト

スクリプト = 実行させたいコマンドを並べたファイル

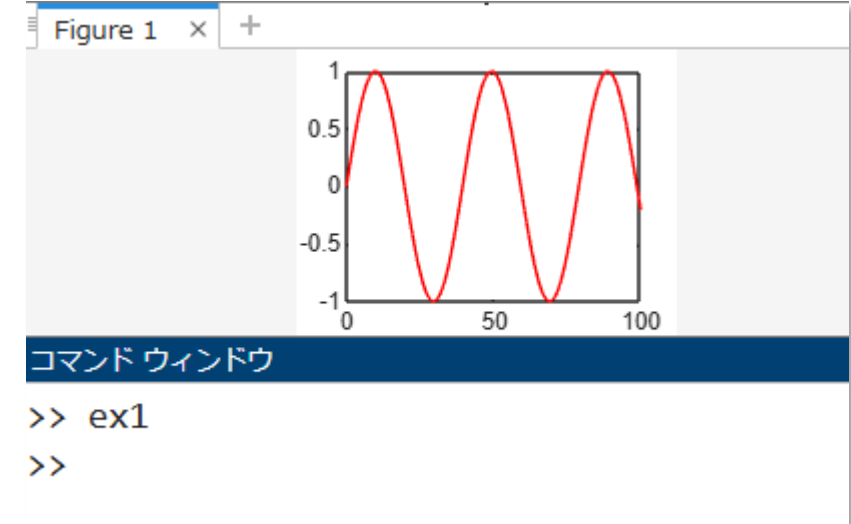
セクション




```

1  %% 正弦波のプロット
2  % 0, 1, 2, ..., 100の整数xを定義する
3  x = 0:100;
4
5  % sin(x / (2*pi))の変数yを定義する
6  y = sin(x / (2*pi));
7
8  % 横軸がx、縦軸がyとなるようにプロットする
9  plot(x, y)
10
11 %% 正弦波のプロットの色を青から赤に変えてみましょう
12 plot(x, y, 'r')
  
```

実行すると



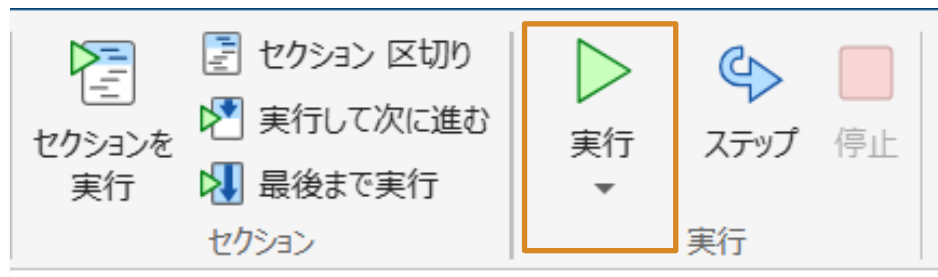
コマンドウィンドウで実行され、
別ウィンドウでFigureが表示されます

 **スクリプト (.m ファイル)**
テキストエディターで編集できる平文のファイル

「%」から始めるとコメント文
「%%」から始めるとセクション区切り

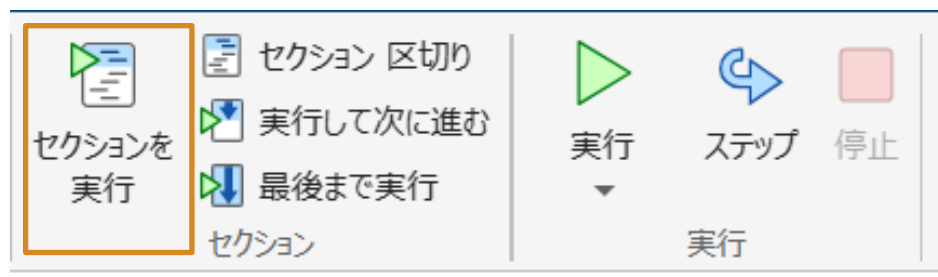
スクリプトのセクション

- 「実行」 = スクリプトの最初から最後まで実行



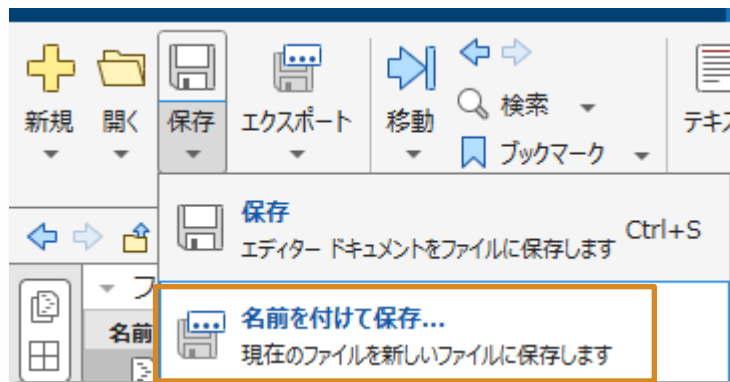
```
ex1.m x +
1 %% 正弦波のプロット
2 % 0, 1, 2, ..., 100の整数xを定義する
3 x = 0:100;
4
5 % sin(x / (2*pi))の変数yを定義する
6 y = sin(x / (2*pi));
7
8 % 横軸がx、縦軸がyとなるようにプロットする
9 plot(x, y)
10
11 %% 正弦波のプロットの色を青から赤に変えてみましょう
12 plot(x, y, 'r')
```

- 「セクションを実行」 = 現在のセクションのみ実行

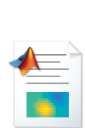
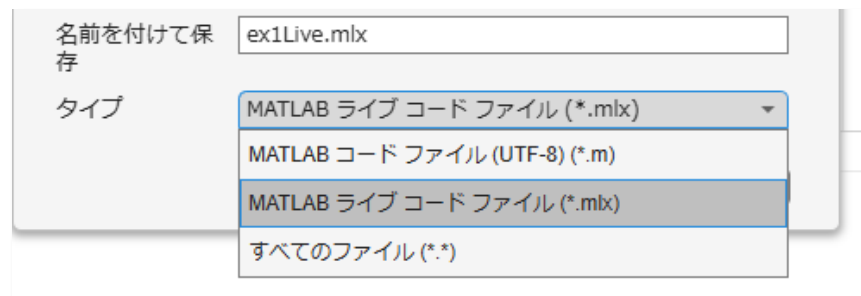


```
ex1.m x +
1 %% 正弦波のプロット
2 % 0, 1, 2, ..., 100の整数xを定義する
3 x = 0:100;
4
5 % sin(x / (2*pi))の変数yを定義する
6 y = sin(x / (2*pi));
7
8 % 横軸がx、縦軸がyとなるようにプロットする
9 plot(x, y)
10
11 %% 正弦波のプロットの色を青から赤に変えてみましょう
12 plot(x, y, 'r')
```

スクリプトとライブスクリプト









ex1.m を「保存」→「名前を付けて保存」で「.mlx」で保存



ライブスクリプト (.mlx ファイル)
エディター、コマンドライン、
Figure が統合されたスクリプト



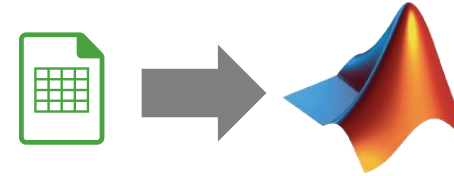
MATLAB ファイルの種類

アイコン	拡張子	説明
	.m	スクリプト。テキストエディターで編集できる平文のファイル
	.mlx	ライブスクリプト。エディター、コマンドライン、Figure が統合したもの
	.mat	mat ファイル。ワークスペースのデータをファイルに保存したもの
	.mlapp	App Designer で作ったGUI アプリファイル
	.fig	Figure をエクスポートしたもの
	.slx	Simulink のモデルファイル

よく使うコマンド

	説明
clc	コマンドウィンドウのコマンド履歴を削除
clear	ワークスペースを削除
close	Figure を閉じる
save	ワークスペースをmat ファイルに保存
doc	ドキュメントを開く

CSV・Excel ファイルの読み取り



readtable 関数を使う方法

```
data = readtable('ball_bouncing.csv');
head(data)
```

Time	Position	Velocity
0	10	15
0.0011423	10.017	14.989
0.0022845	10.034	14.978

テーブルとして読み取り

他にも

readtimetable: タイムテーブルとして読み取り

readmatrix: 数値配列として読み取り

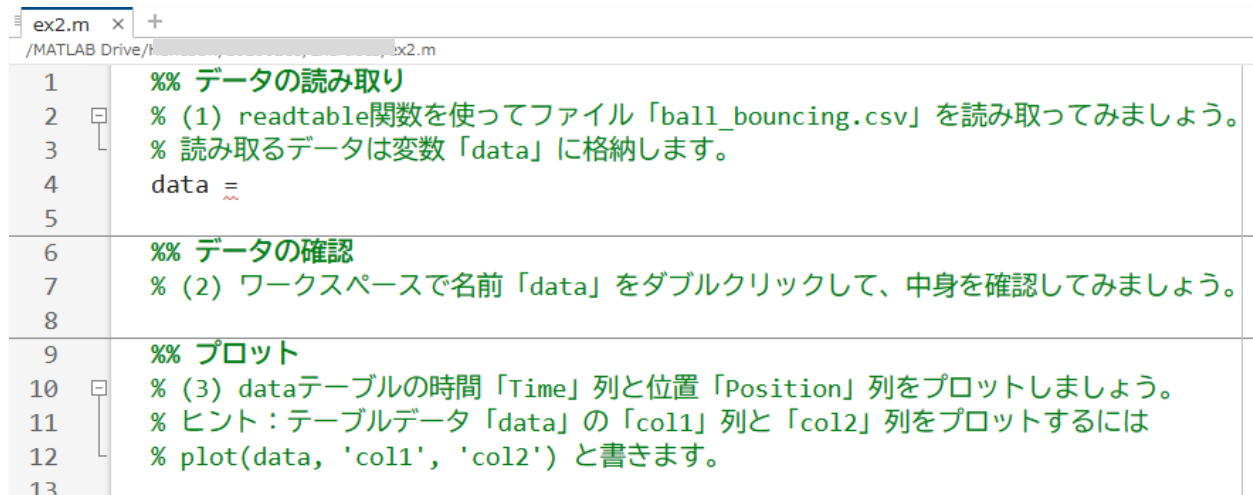
readcell: セル配列として読み取り

readvars: 列毎に別変数として読み取り

インポートウィザードを使う方法

ライブエディターのタスクを使う方法

演習 2 ファイルの読み込み、可視化



```
1 %% データの読み取り
2 % (1) readtable関数を使ってファイル「ball_bouncing.csv」を読み取ってみましょう。
3 % 読み取るデータは変数「data」に格納します。
4 data =
5
6 %% データの確認
7 % (2) ワークスペースで名前「data」をダブルクリックして、中身を確認してみましょう。
8
9 %% プロット
10 % (3) dataテーブルの時間「Time」列と位置「Position」列をプロットしましょう。
11 % ヒント：テーブルデータ「data」の「col1」列と「col2」列をプロットするには
12 % plot(data, 'col1', 'col2') と書きます。
13
```

編集するファイル：ex2.m

(1) readtable 関数を使って
ball_bouncing.csv を読み込むコードを
書きます

ヒント

- readtable('ファイル名.csv')でCSV ファイル
をテーブルとして読み込めます

(2) ワークスペースで名前「data」をダブル
クリックして、中身を確認してみま
しょう

(3) data テーブルの時間「Time」列と位
置「Position」列をプロットしましょう

ヒント

- テーブルデータ「data」の「col1」列と
「col2」列をプロットするにはplot(data,
'col1', 'col2') と書きます。

演習 2 ファイルの読み込み、可視化

```
14 %% プロットの重ね描き
15 % (4) 座標軸をホールドして、
16 % 時間「Time」列と速さ「Velocity」列のプロットを重ね描きします。
17 % ヒント：座標軸をホールドするにはhold onと書きます。
18
19
20 %% ホールドの解除
21 hold off
22
23 % 凡例表示
24 legend({'位置', '速さ'})
```

編集するファイル：ex2.m

(4) 座標軸をホールドして、時間「Time」列と速さ「Velocity」列のプロットを重ね描きします。

ヒント

- 座標軸をホールドするにはhold onと書きます。
- テーブルデータ「data」の「col1」列と「col2」列をプロットするにはplot(data, 'col1', 'col2') と書きます。

演習 2 ファイルの読み込み、可視化【解答】

```

ex2.m
/MATLAB Drive/.../s/ex2.m

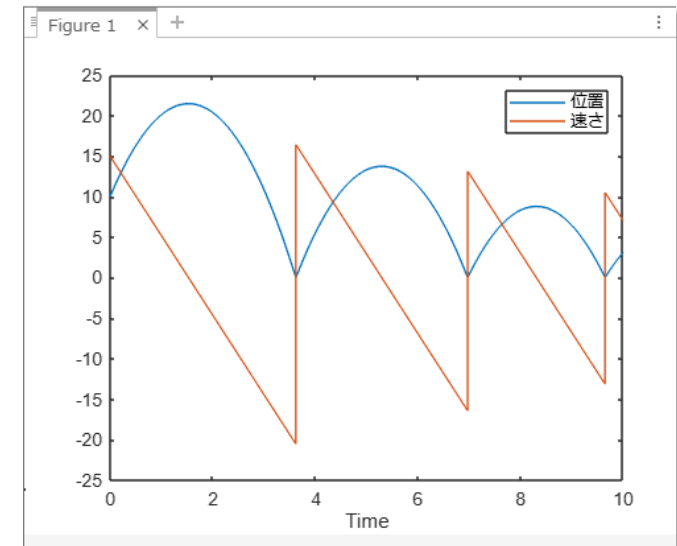
1 %% データの読み取り
2 % (1) readtable関数を使ってファイル「ball_bouncing.csv」を読み取ってみましょう。
3 % 読み取るデータは変数「data」に格納します。
4 data = readtable('ball_bouncing.csv');
5
6 %% データの確認
7 % (2) ワークスペースで名前「data」をダブルクリックして、中身を確認してみましょう。
8
9 %% プロット
10 % (3) dataテーブルの時間「Time」列と位置「Position」列をプロットしましょう。
11 % ヒント：テーブルデータ「data」の「col1」列と「col2」列をプロットするには
12 % plot(data, 'col1', 'col2') と書きます。
13 plot(data, 'Time', 'Position')
14
15 %% プロットの重ね描き
16 % (4) 座標軸をホールドして、
17 % 時間「Time」列と速さ「Velocity」列のプロットを重ね描きします。
18 % ヒント：座標軸をホールドするにはhold onと書きます。
19 hold on
20 plot(data, 'Time', 'Velocity')
21
22 %% ホールドの解除
23 hold off
24
25 % 凡例表示
26 legend({'位置', '速さ'})

```

data 146×3 table

	1 Time	2 Position	3 Velocity
最小値	0	0	-20.5231
最大値	10	21.4576	16.4185
平均値	5.0899	7.99	3.0216
標準偏差	3.1503	6.6481	10.6153
欠損値	0	0	0
クラス	double	double	double

	0	10	15
1			
2	0.0011	10.0171	14.9888
3	0.0023	10.0342	14.9776
4	0.0034	10.0513	14.9664
5	0.0084	10.1259	14.9174
6	0.0134	10.2003	14.8684



解答例：ex2_ans.m

スクリプトの関数化



スクリプトと関数

- スクリプト
 - コードの羅列
 - 再利用性が低い

```
ex2.m +
/MATLAB Drive/ s/ex2.m
1 %% データの読み取り
2 % (1) readtable関数を使ってファイル「ball_bouncing.csv」を読み取ってみましょう。
3 % 読み取るデータは変数「data」に格納します。
4 data = readtable('ball_bouncing.csv');
5
6 %% データの確認
7 % (2) ワークスペースで名前「data」をダブルクリックして、中身を確認してみましょう。
8
9 %% プロット
10 % (3) dataテーブルの時間「Time」列と位置「Position」列をプロットしましょう。
11 % ヒント：テーブルデータ「data」の「col1」列と「col2」列をプロットするには
12 % plot(data, 'col1', 'col2') と書きます。
13 plot(data, 'Time', 'Position')
14
15 %% プロットの重ね描き
16 % (4) 座標軸をホールドして、
17 % 時間「Time」列と速さ「Velocity」列のプロットを重ね描きします。
18 % ヒント：座標軸をホールドするにはhold onと書きます。
19 hold on
20 plot(data, 'Time', 'Velocity')
21
22 %% ホールドの解除
23 hold off
24
25 % 凡例表示
26 legend({'位置', '速さ'})
```

ファイル名がスクリプトにハードコード

スクリプトと関数

関数

- 冒頭にfunction を付けたスクリプト
- 入力引数と出力引数を定義する
- 再利用性が高い
- 関数ファイルとローカル関数
- ベースワークスペースと

関数ワークスペースの範囲に注意

関数ファイル

```
mysin.m  x  +
1  function y = mysin(x)
2      y = sin(x/(2*pi));
3  end
```

コマンドウィンドウ

```
>> x = 0:100;
>> y = mysin(x);
>> plot(x, y)
```

ローカル関数

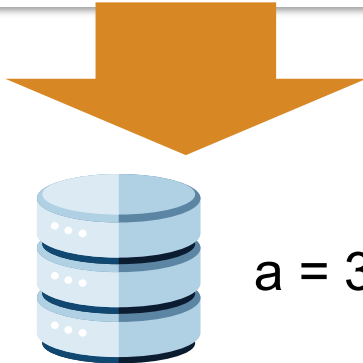
スクリプトの中に関数を定義する(=ローカル関数)
同じファイルからのみ呼び出せる

```
callLocalFunction.m  x  +
1      x = 0:100;
2
3      % ローカル関数
4      function y = mysin2(x)
5          y = sin(x/(2*pi));
6      end
7
8      % ローカル関数の呼び出し
9      y = mysin2(x);
10     plot(x, y)
```

ワークスペースの種類

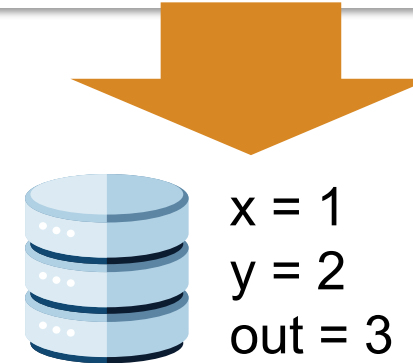
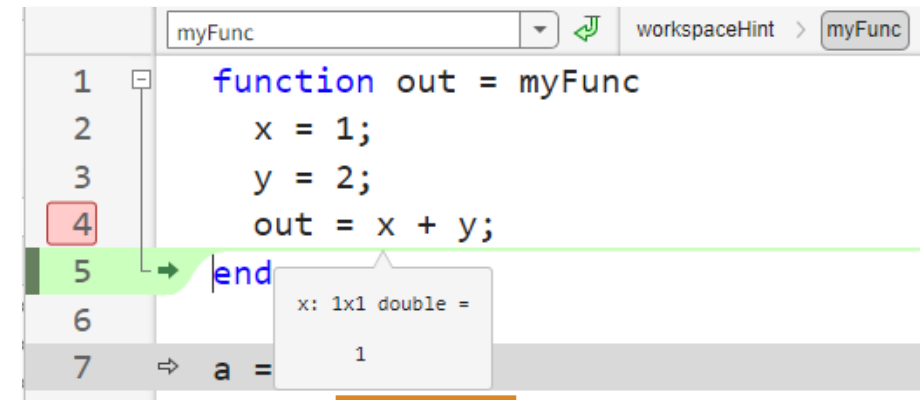
関数の内部で定義した変数はベースワークスペースに値が格納されない出力変数 (ここではout) のみリターンされる

```
1 function out = myFunc
2     x = 1;
3     y = 2;
4     out = x + y;
5 end
6
7 a = myFunc;
```



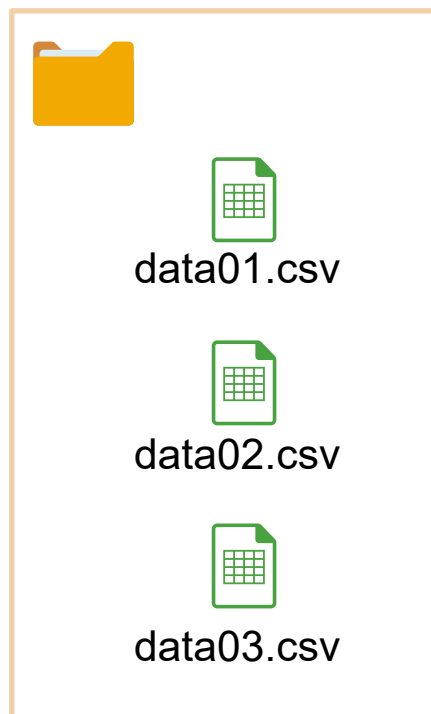
ベースワークスペース

関数内でブレークポイントを設定すると、関数ワークスペースが見える

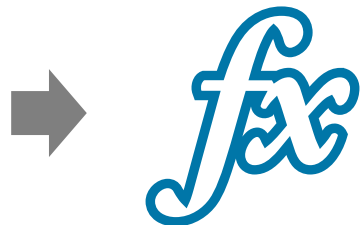


関数ワークスペース

スクリプトを関数にする



ファイルパスを入力引数にすれば再利用できる



```
ex2Func.m × +
/MATLAB Drive/ 6/ex2Func.m
1 function ex2Func(filePath)
2     %% データの読み取り
3     data = readtable(filePath);
4
5     %% プロット
6     % 時間と位置のプロット
7     plot(data, 'Time', 'Position')
8
9     % 座標軸をホールドして重ね描きする
10    hold on
11
12    % 時間と速さのプロット
13    plot(data, 'Time', 'Velocity')
14
15
```

- (1) ex2.m を開き、冒頭にfunction ex2Funcを付けます
- (2) function に小括弧()を追加し、その中にfilePath を追加します
- (3) readtable の中身をfilePath に変更します
- (4) ファイル名「ex2Func.m」で保存します
- (5) コマンドウィンドウからex2Func('ファイル名.csv') で実行します

コマンドウィンドウ

```
>> ex2Func('ball_bouncing.csv')
>> |
```

MATLAB Copilot の活用



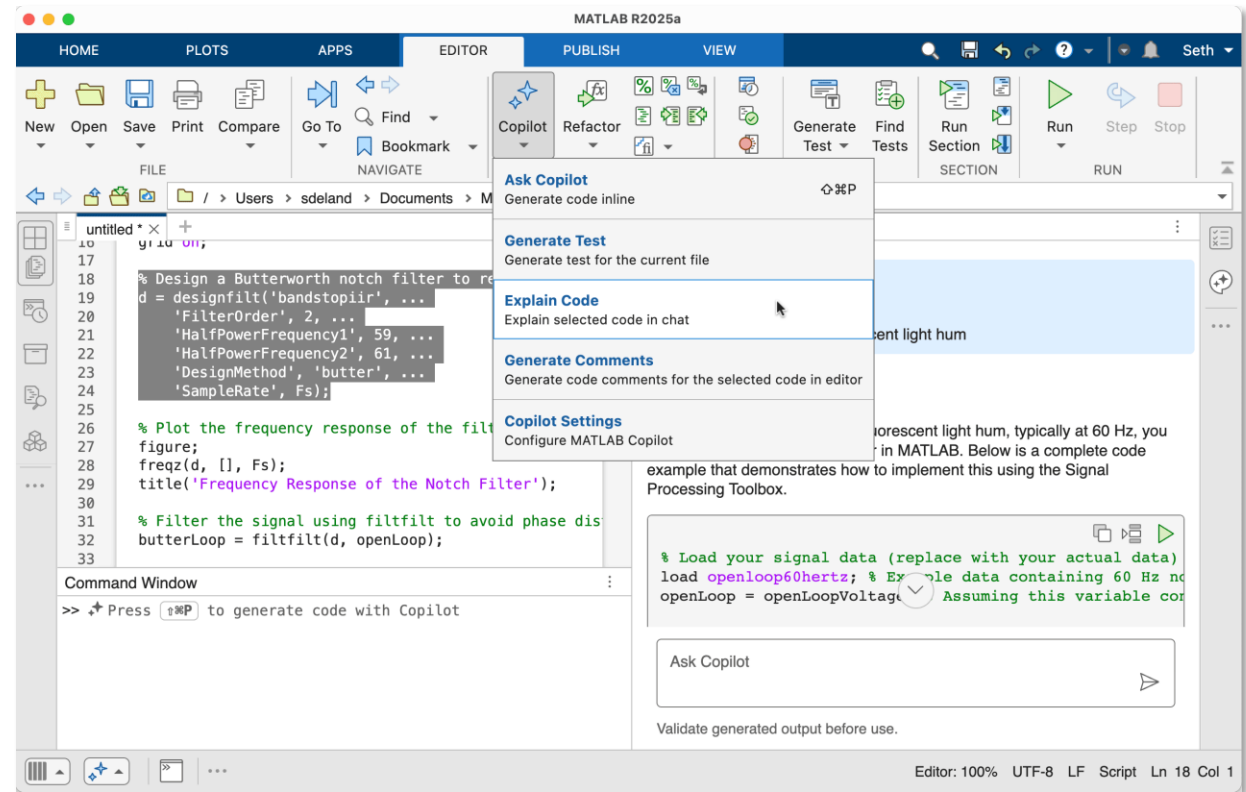
コーディングを楽にするMATLAB Copilot

MATLAB Copilot は、生成AI モデルをMATLAB のワークフローに統合します

R2025a

主な機能

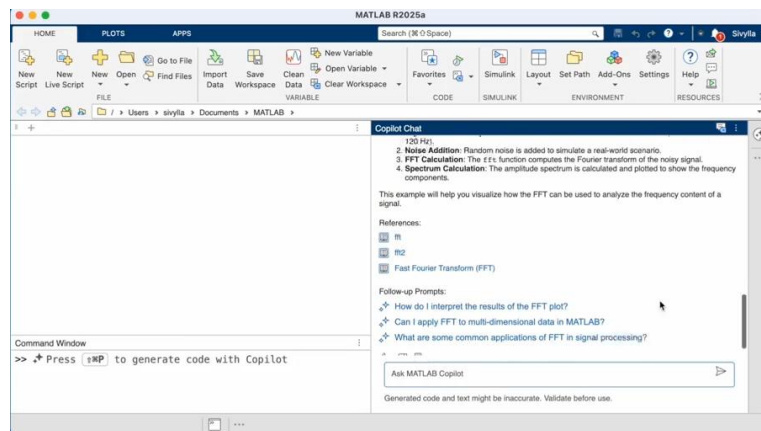
- ドキュメントに基づいたチャット
- チャットでコードの生成
- エディター上でのコード自動生成
- コードやエラーメッセージの説明
- ソフトウェアテストの自動生成



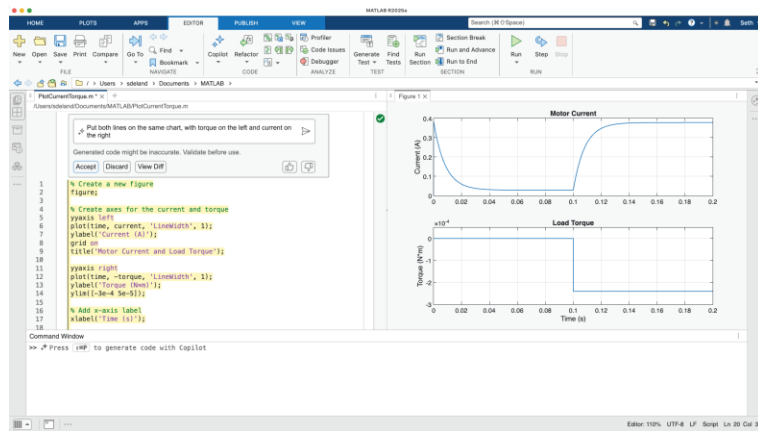
MATLAB Copilot 製品ページ

<https://jp.mathworks.com/products/matlab-copilot.html>

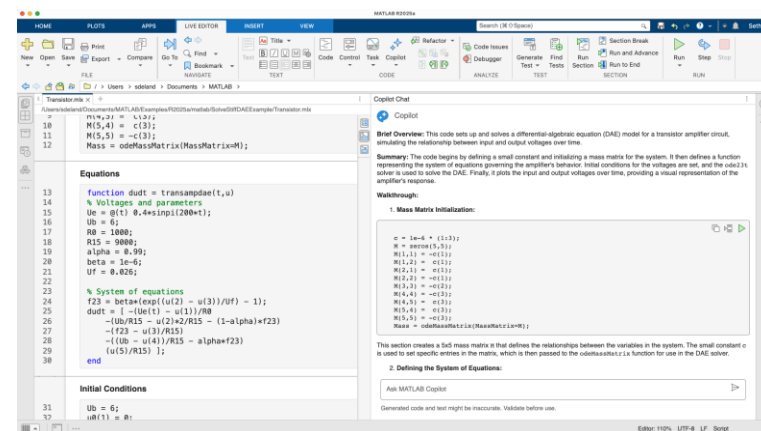
MATLAB Copilot の 6 つの使い方



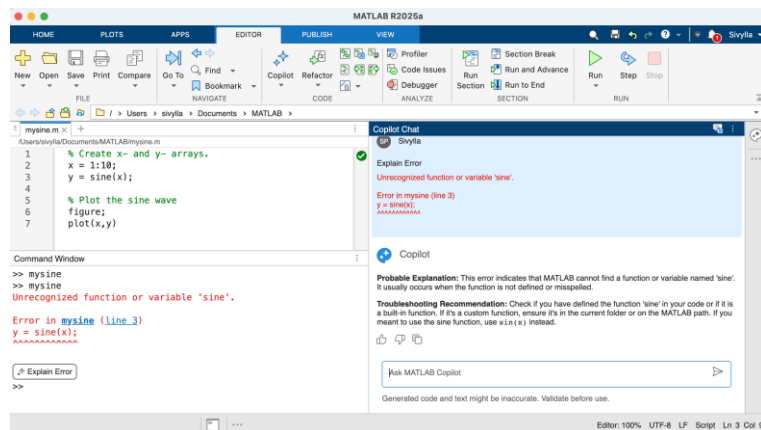
チャットで質問



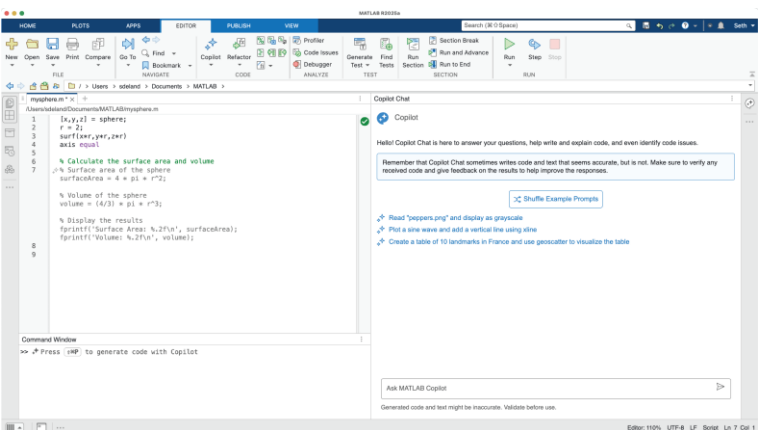
コードの生成・修正



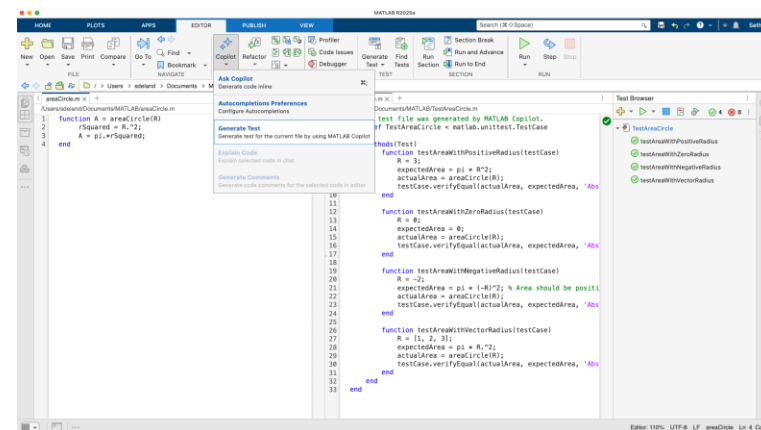
コードの解説・コメント挿入



エラーの解説



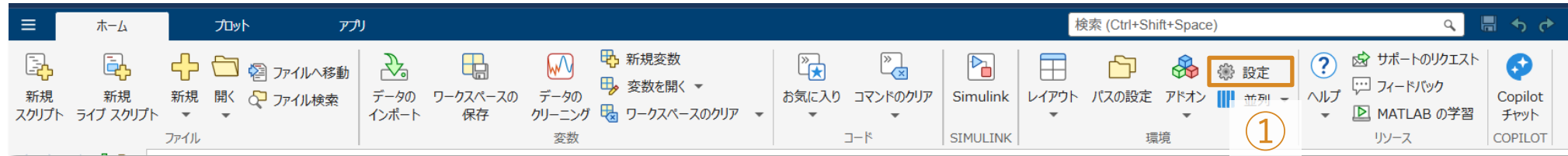
コードの予測入力



テストコードの自動生成
with MATLAB Test

【設定】 MATLAB Copilot をON にする

MATLAB Copilot を使用するので設定からON に変更します



演習 3 MATLAB Copilot にコードを書いてもらう

- コードの要件をCopilot のチャットに入れてみてください

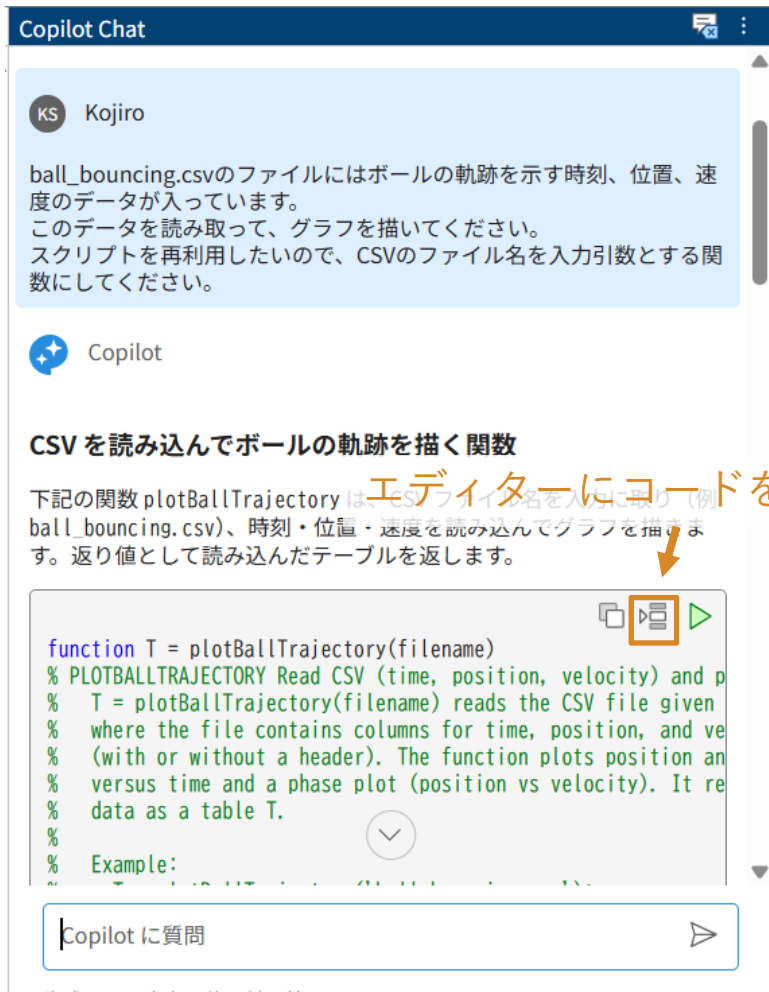
ball_bouncing.csvのファイルにはボールの軌跡を示す時刻、位置、速度のデータが入っています。

このデータを読み取って、グラフを描いてください。
スクリプトを再利用したいので、CSVのファイル名を入力引数とする関数にしてください。

ex3_Copilot_チャット内容.txt

演習 3 MATLAB Copilot にコードを書いてもらう

MATLAB Copilot が返したコードを、中央のボタンをクリックしてエディターに挿入します



Copilot Chat

KS Kojiro

ball_bouncing.csvのファイルにはボールの軌跡を示す時刻、位置、速度のデータが入っています。
このデータを読み取って、グラフを描いてください。
スクリプトを再利用したいので、CSVのファイル名を入力引数とする関数にしてください。

Copilot

CSVを読み込んでボールの軌跡を描く関数

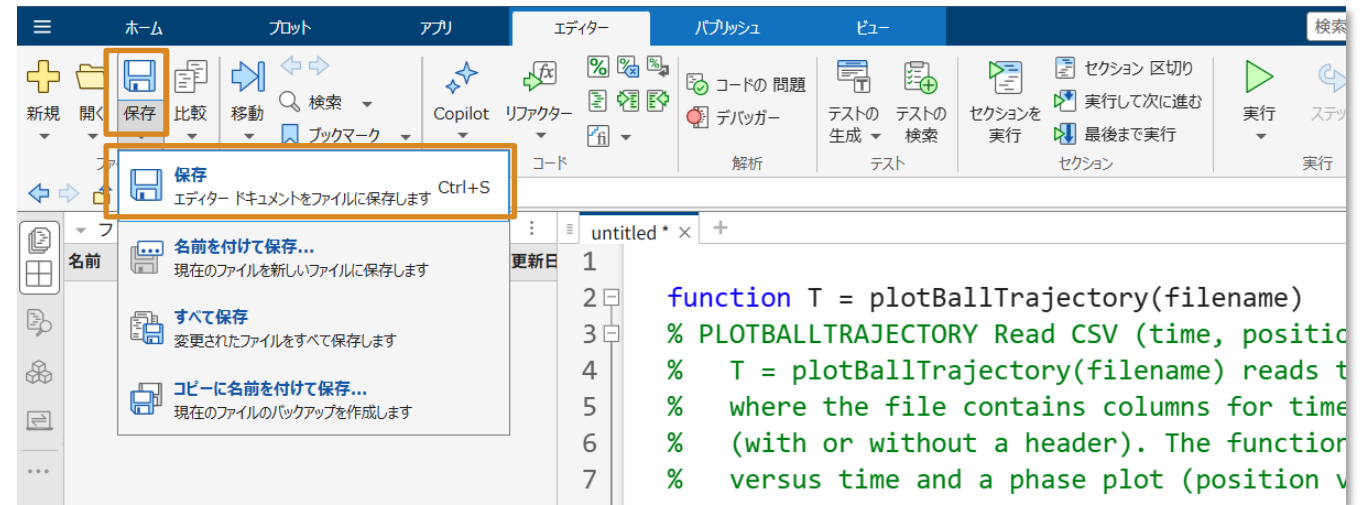
下記の関数 plotBallTrajectory は、CSVファイル名を入力に取って (例: ball_bouncing.csv)、時刻・位置・速度を読み込んでグラフを描きます。返り値として読み込んだテーブルを返します。

```
function T = plotBallTrajectory(filename)
% PLOTBALLTRAJECTORY Read CSV (time, position, velocity) and plot
% T = plotBallTrajectory(filename) reads the CSV file given
% where the file contains columns for time, position, and velocity
% (with or without a header). The function plots position versus time
% and a phase plot (position vs velocity). It returns the data as a table T.
%
% Example:
% T = plotBallTrajectory('ball_bouncing.csv');
```

Copilot に質問

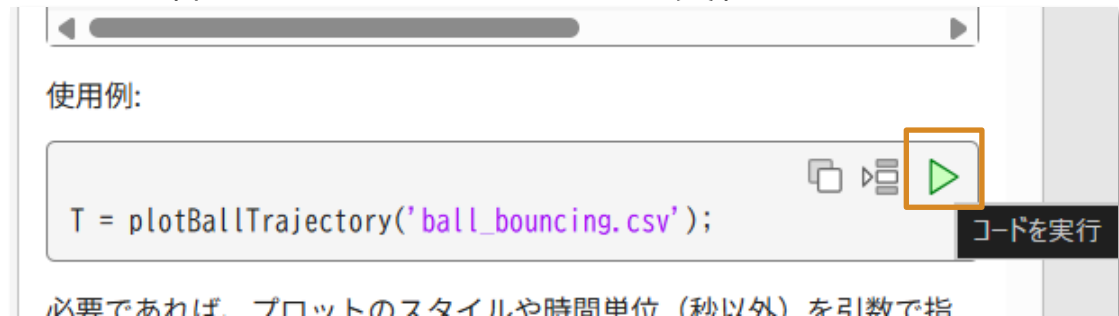
エディターにコードを挿入

保存ボタンをクリックしてデフォルトのファイル名 (例: plotBallTrajectory.m) で保存します



保存ボタンをクリックしてデフォルトのファイル名 (例: plotBallTrajectory.m) で保存します

MATLAB Copilot チャットの使用例のコードの右のボタンをクリックして実行します



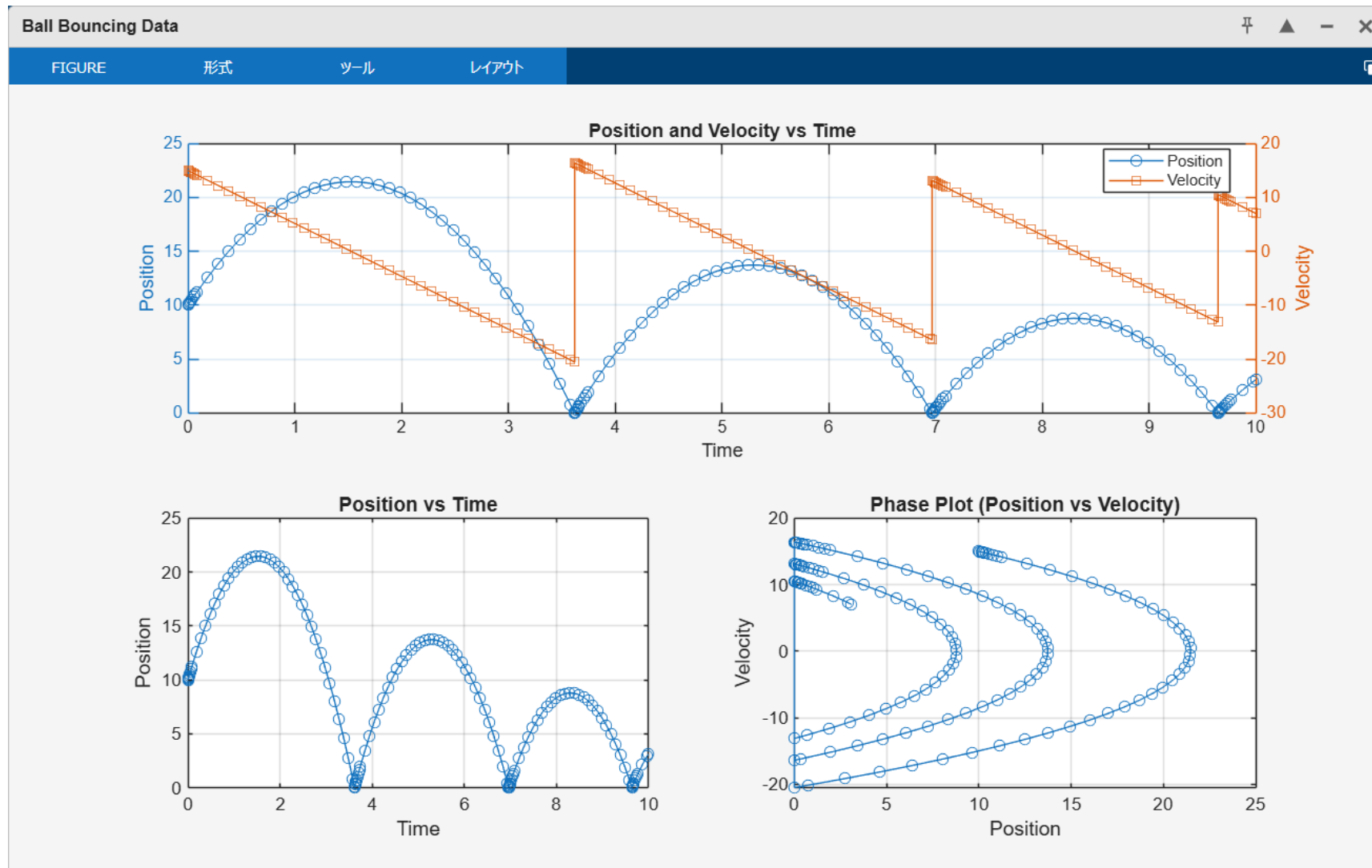
使用例:

```
T = plotBallTrajectory('ball_bouncing.csv');
```

コードを実行

必要であれば、プロットのスタイルや時間単位 (秒以外) を引数で指定

演習 3 MATLAB Copilot にコードを書いてもらう



※生成AI の回答は毎回変わることがあります

演習 3 MATLAB Copilot にコードを書いてもらう

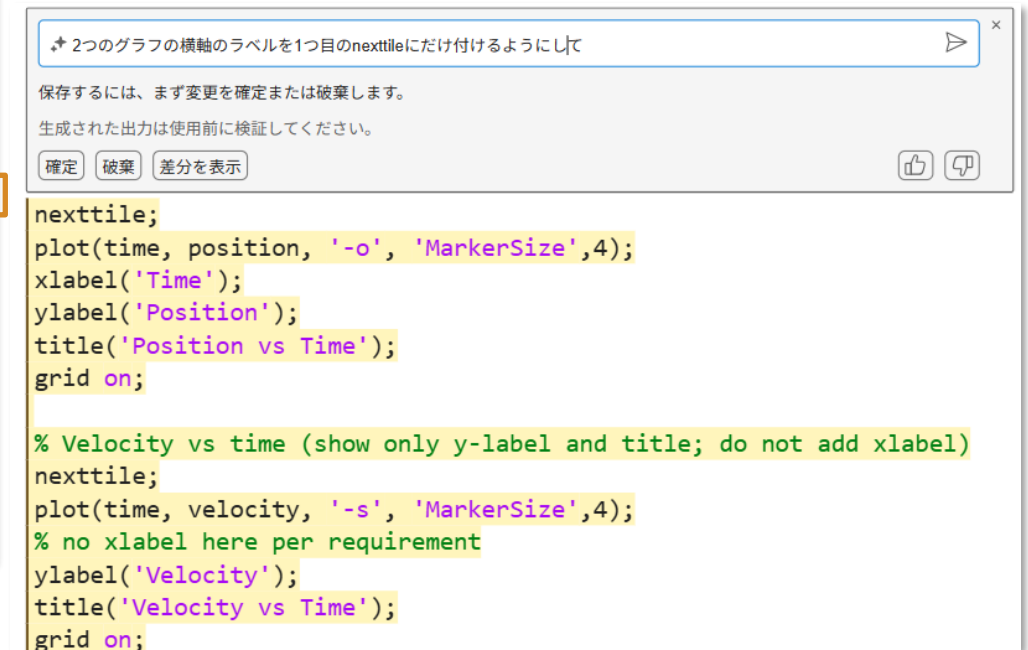
コードを変更

エディターで関数ファイル (例: plotBallTrajectory.m) を開き、
 任意のコードをマウสดラッグで選択し、
 右クリック「Copilot に質問」をクリックします



変更したい内容を書いて実行します

例： 2つのグラフの横軸のラベルを1つ目のnexttileにだけ付けるようにして



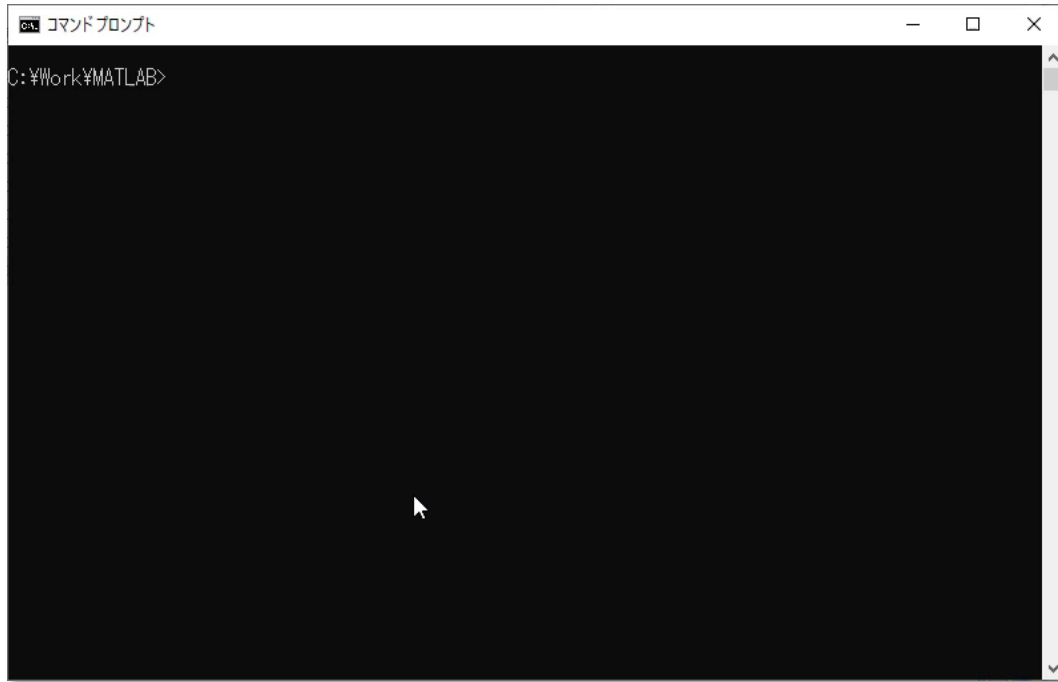
問題なければ「確定」をクリックして反映させます

手軽にGUI アプリ作成

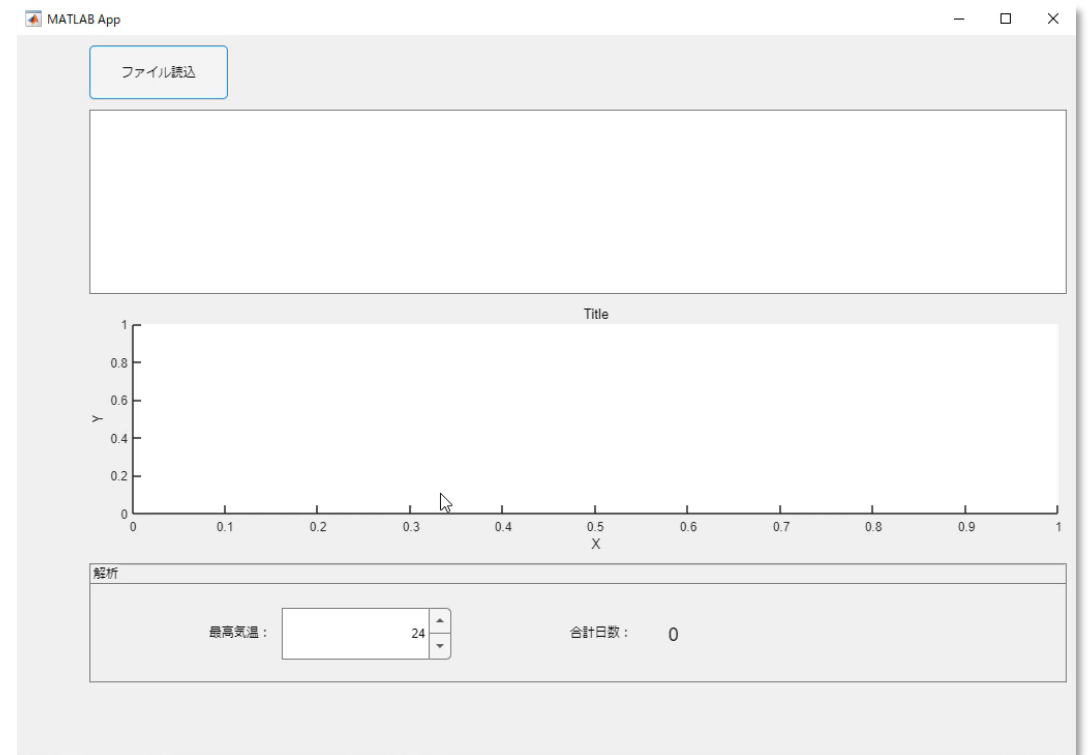


どちらが使ってみたいアプリですか？

①

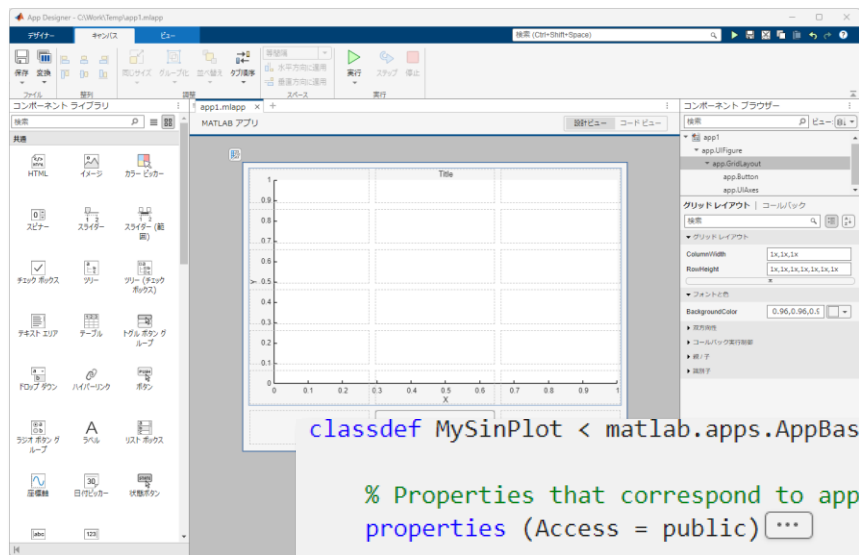


②



MATLAB でのGUI アプリの作成

App Designer



```
classdef MySinPlot < matlab.apps.AppBase
```

```
% Properties that correspond to app components
properties (Access = public) ...
```

```
% Callbacks that handle component events
methods (Access = private)
```

```
% Button pushed function: Button
function ButtonPushed(app, event)
```

```
    x = 0:100;
    y = sin(x / (2*pi));
    plot(app.UIAxes, x, y)
```

```
end
```

```
end
```

拡張子 .mlapp

uifigure コマンド

```
% uifigureを作成
```

```
uif = uifigure(Name = 'My App', Position = [500 500 420 270]);
```

```
% ボタンのコールバック関数
```

```
function buttonPushed(src, event, ax)
```

```
    x = 0:100;
```

```
    y = sin(x / (2*pi));
```

```
    plot(ax, x, y)
```

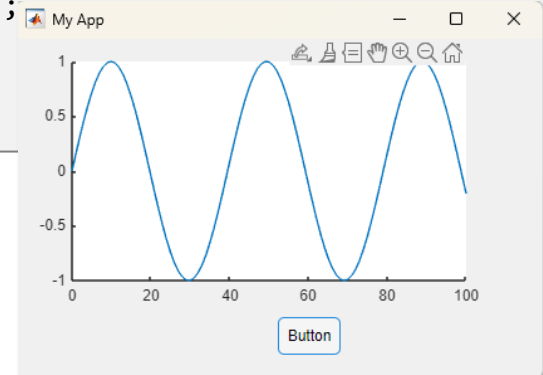
```
end
```

```
% 座標軸を追加
```

```
ax = uiaxes(uif, Position = [20 60 350 200]);
```

```
% ボタンを追加
```

```
uib = uibutton(uif, ButtonPushedFcn = {@buttonPushed, ax}, ...
    Position = [210 20 50 30]);
```

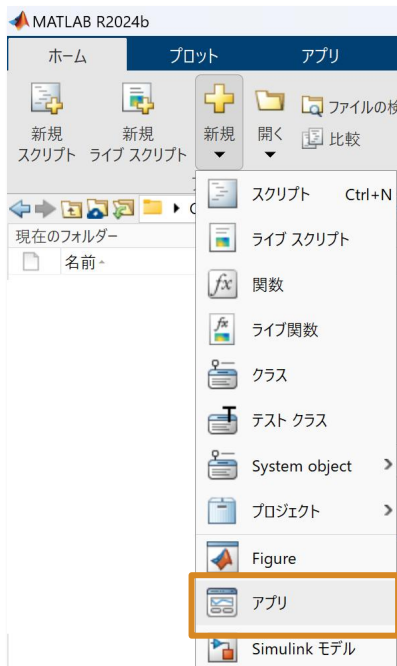


App Designer の起動方法

コマンドウィンドウで「appdesigner」を入力してエンターキーで実行

コマンド ウィンドウ
>> appdesigner

「ホーム」タブの「新規」→
「アプリ」をクリック

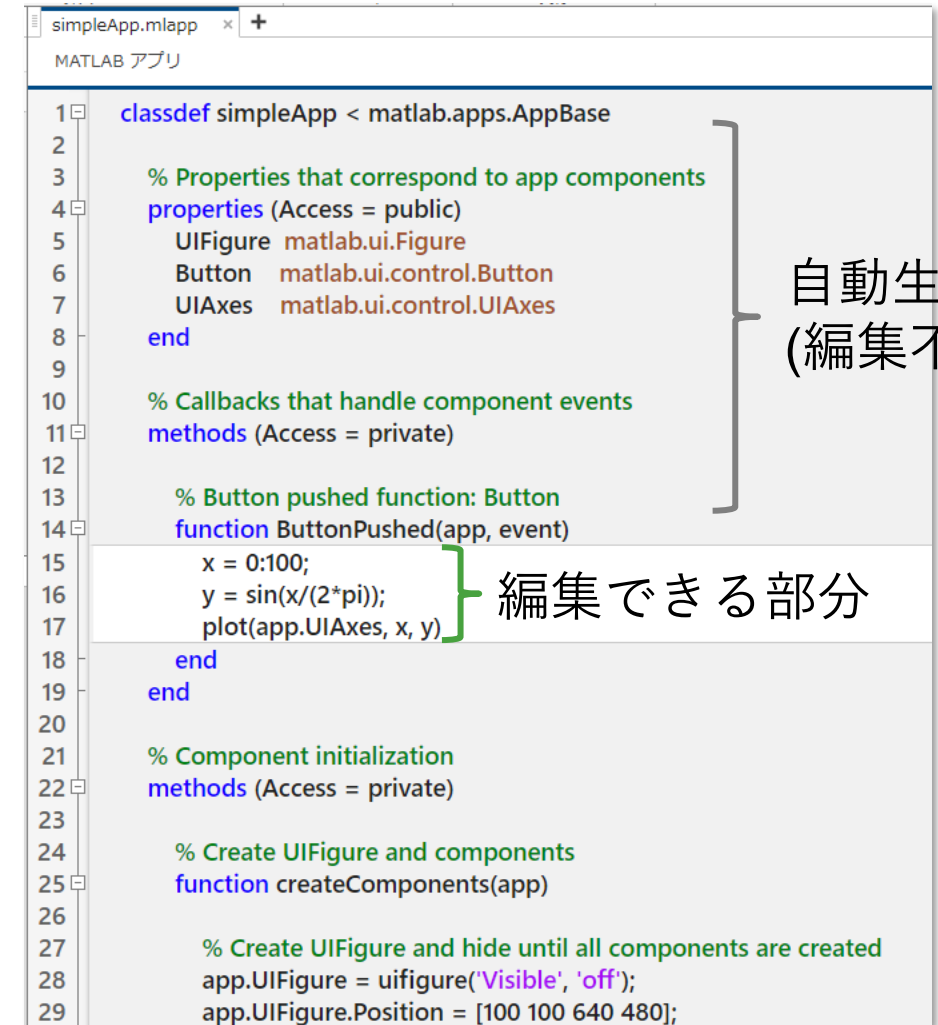
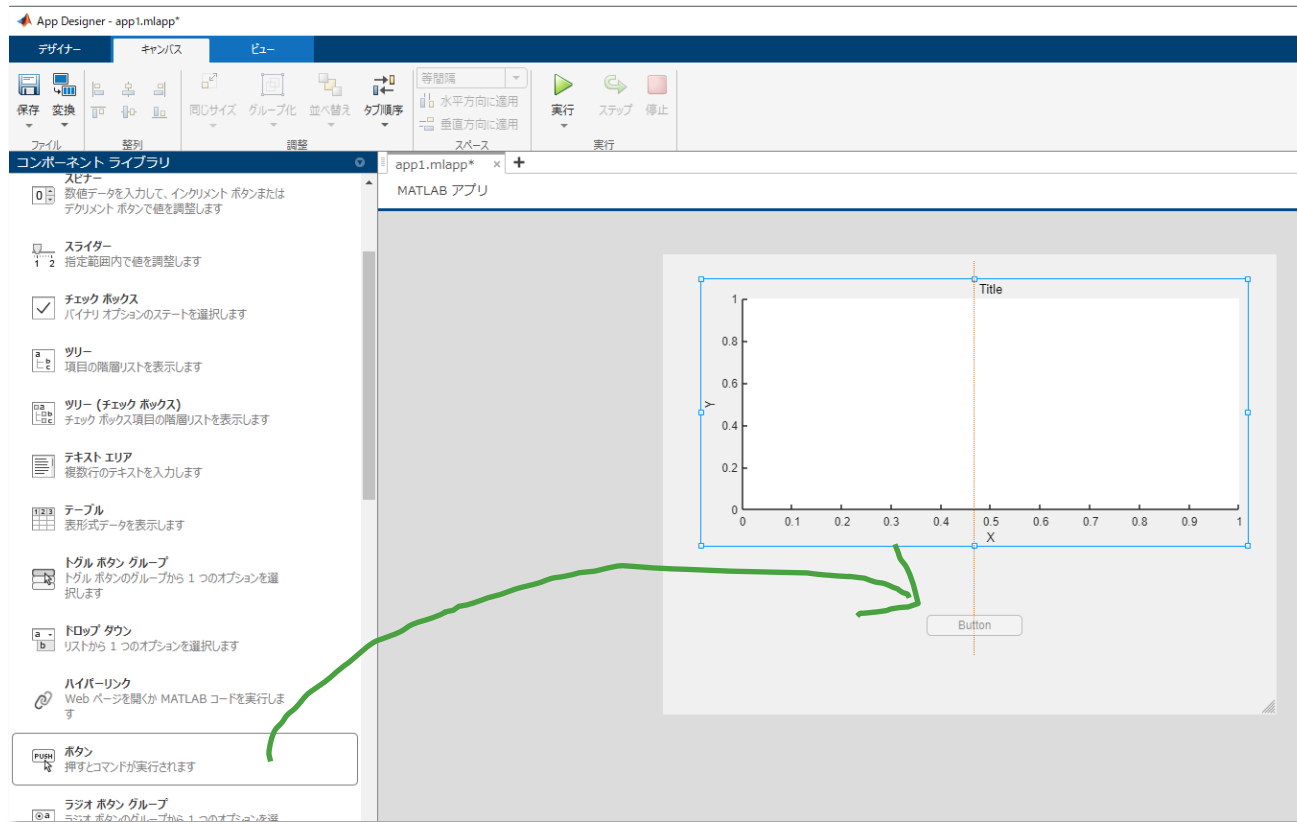


App Designer が起動し、テンプレートからアプリを作成できます
「空のアプリ」でスクラッチから作成できます



App Designer でのGUI アプリ作成

- ドラッグ&ドロップでコンポーネントを置き、コールバックに最小限のMATLAB コードを書いてGUI アプリを作成



App Designer の使い方

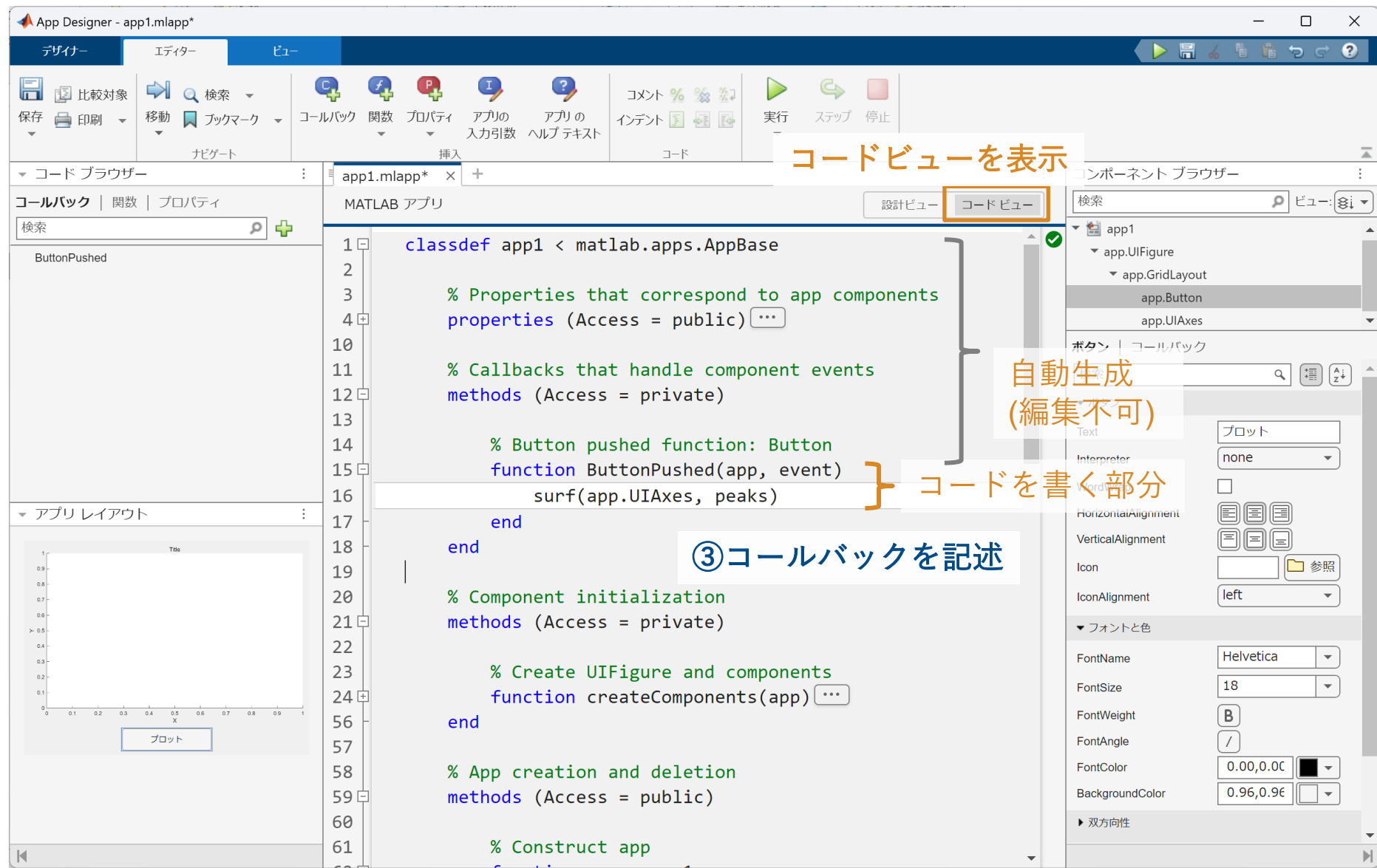
The screenshot displays the MathWorks App Designer interface for a file named 'app1.mlapp*'. The top toolbar includes tabs for 'デザイナー' (Designer), 'キャンパス' (Canvas), and 'ビュー' (View), with 'デザイナー' currently selected. Below these are icons for saving, undo, redo, and other design tools. The main workspace is divided into three sections:

- コンポーネント ライブラリ (Component Library):** Located on the left, it contains a grid of various UI components such as 'スピナー' (Spinner), 'スライダー' (Slider), 'チェック ボックス' (Checkbox), 'ツリー' (Tree), 'テキスト エリア' (Text Area), 'テーブル' (Table), 'ドロップ ダウン' (Dropdown), 'ラジオ ボタン グループ' (Radio Button Group), '座標軸' (Axes), '日付ピッカー' (Date Picker), and '状態ボタン' (Status Button). A blue arrow points from the 'ボタン' (Button) component in this library to the 'プロット' (Plot) component on the canvas.
- キャンパス (Canvas):** The central workspace where the app is designed. It features a grid and a plot area with axes labeled 'X' and 'Y'. A blue box highlights a 'プロット' (Plot) component on the canvas, with the label 'キャンバス' (Canvas) written in orange above it.
- コンポーネント ブラウザー (Component Browser):** Located on the right, it shows the hierarchy of components in the app. The tree structure is: 'app1' > 'app.UIFigure' > 'app.GridLayout' > 'app.Button' > 'app.UIAxes'. Below the tree, there are settings for 'ボタン' (Button) and 'コールバック' (Callback), including 'Text', 'Interpreter', 'WordWrap', 'HorizontalAlignment', 'VerticalAlignment', 'FontSize', 'FontWeight', 'FontAngle', 'FontColor', and 'BackgroundColor'. A blue box highlights the 'ボタン' settings section.

Annotations and instructions are provided in Japanese:

- 設計ビューを表示 (Show Design View):** A label pointing to the '設計ビュー' (Design View) button in the top toolbar.
- ①ドラッグ&ドロップでコンポーネントを置きレイアウトを設計 (Design layout by dragging and dropping components):** A blue instruction at the bottom.
- ②コンポーネントブラウザーからプロパティの編集 (Edit properties from the Component Browser):** A blue instruction at the bottom.
- コンポーネント一覧 (Component List):** A label pointing to the Component Library on the left.

App Designer の使い方



App Designer のコンポーネント

共通

- HTML**
簡易マークアップまたは組み込み HTML ファイルを表示します
- イメージ**
アイコンまたはロゴを表示します
- カラー ピッカー**
色を選択して表示します
- スピナー**
数値データを入力して、インクリメント ボタンまたはデクリメント ボタンで値を調整します
- スライダー**
指定範囲内で値を選択します
- スライダー (範囲)**
指定範囲内で値の範囲を選択します
- チェック ボックス**
バイナリ オプションのステートを選択します
- ツリー**
項目の階層リストを表示します
- ツリー (チェック ボックス)**
チェック ボックス項目の階層リストを表示します
- テキスト エリア**
複数行のテキストを入力します
- テーブル**
表形式データを表示します

- トグル ボタン グループ**
トグル ボタンのグループから 1 つのオプションを選択します
- ドロップ ダウン**
リストから 1 つのオプションを選択します
- ハイパーリンク**
Web ページを開くか MATLAB コードを実行します
- ボタン**
押すとコマンドが実行されます
- ラジオ ボタン グループ**
ラジオ ボタンのグループから 1 つのオプションを選択します
- ラベル**
コンポーネントを説明するテキストを表示します
- リスト ボックス**
リストから 1 つ以上のオプションを選択します
- 座標軸**
グラフィカル データの可視化を表示します
- 日付ピッカー**
日付を選択して指定した形式で表示します
- 状態ボタン**
2 つのステート間で切り替えます
- 編集フィールド (テキスト)**
テキスト データを入力します
- 編集フィールド (数値)**
数値データを入力します

コンテナー

- グリッドレイアウト**
指定したサイズ変更動作でグリッド内のコンポーネントを配置します
- タブ グループ**
さまざまなタブにコンポーネントをグループ化して管理します
- パネル**
コンポーネントをグループ化します

Figure ツール

- コンテキスト メニュー**
関連するコンポーネントが右クリックされたときにコンテキスト メニューを表示します
- ツール バー**
ツールをグループ化してアプリの上部に表示します
- メニュー バー**
アプリケーション コマンドとオプションを機能別にグループ化して表示します

計器

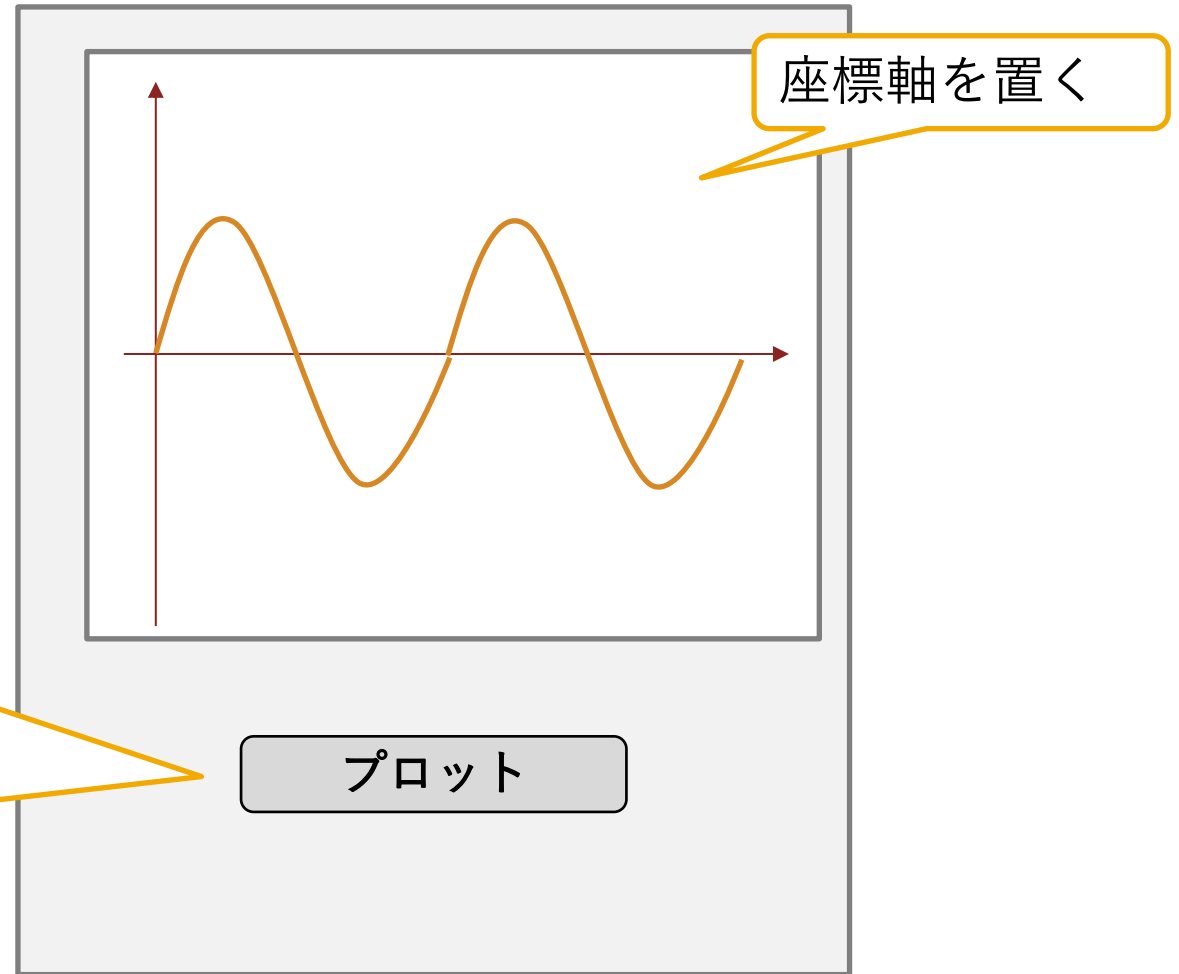
- ゲージ**
放射状スケールで値を表示します
- ゲージ (4 分円)**
90 度の放射状スケールで値を表示します
- ゲージ (半円)**
180 度の放射状スケールで値を表示します
- ゲージ (直線)**
線形スケールで値を表示します
- スイッチ**
2 つの排他的ステート間で切り替えます
- スイッチ (トグル)**
2 つの排他的ステート間で切り替えます
- スイッチ (ロック)**
2 つの排他的ステート間で切り替えます
- ノブ**
指定範囲内で値を調整します
- ノブ (離散型)**
複数の別個の状態の 1 つに値を調整します
- ランプ**
光ることによりステータスを示します

演習 4 GUI アプリの作成

学べること

- コンポーネントの置き方
- コールバックの書き方
- アプリ内の座標軸にプロットする方法

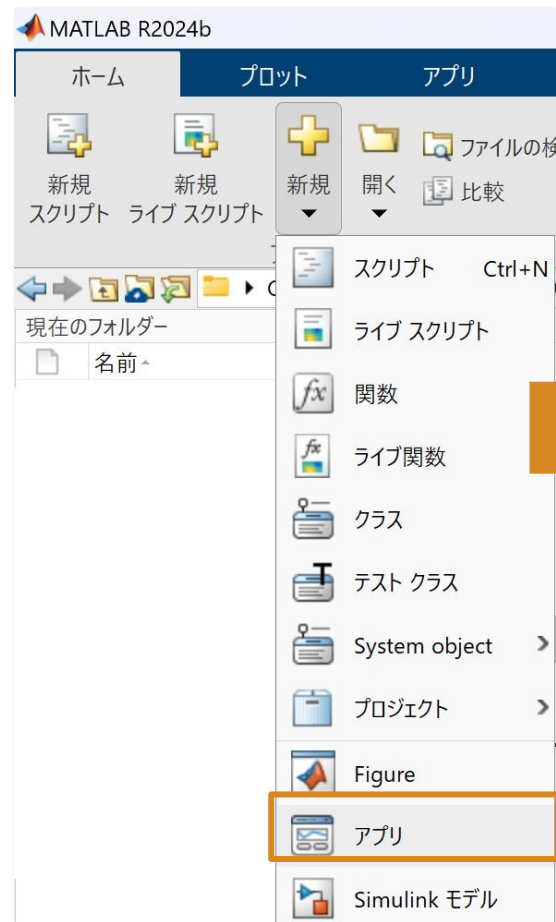
ボタンをクリックすると
 $x=0, 1, 2, \dots, 100$ の整数
 $y=\sin(x/2\pi)$
が計算され、 x と y が座標軸
にプロットされる



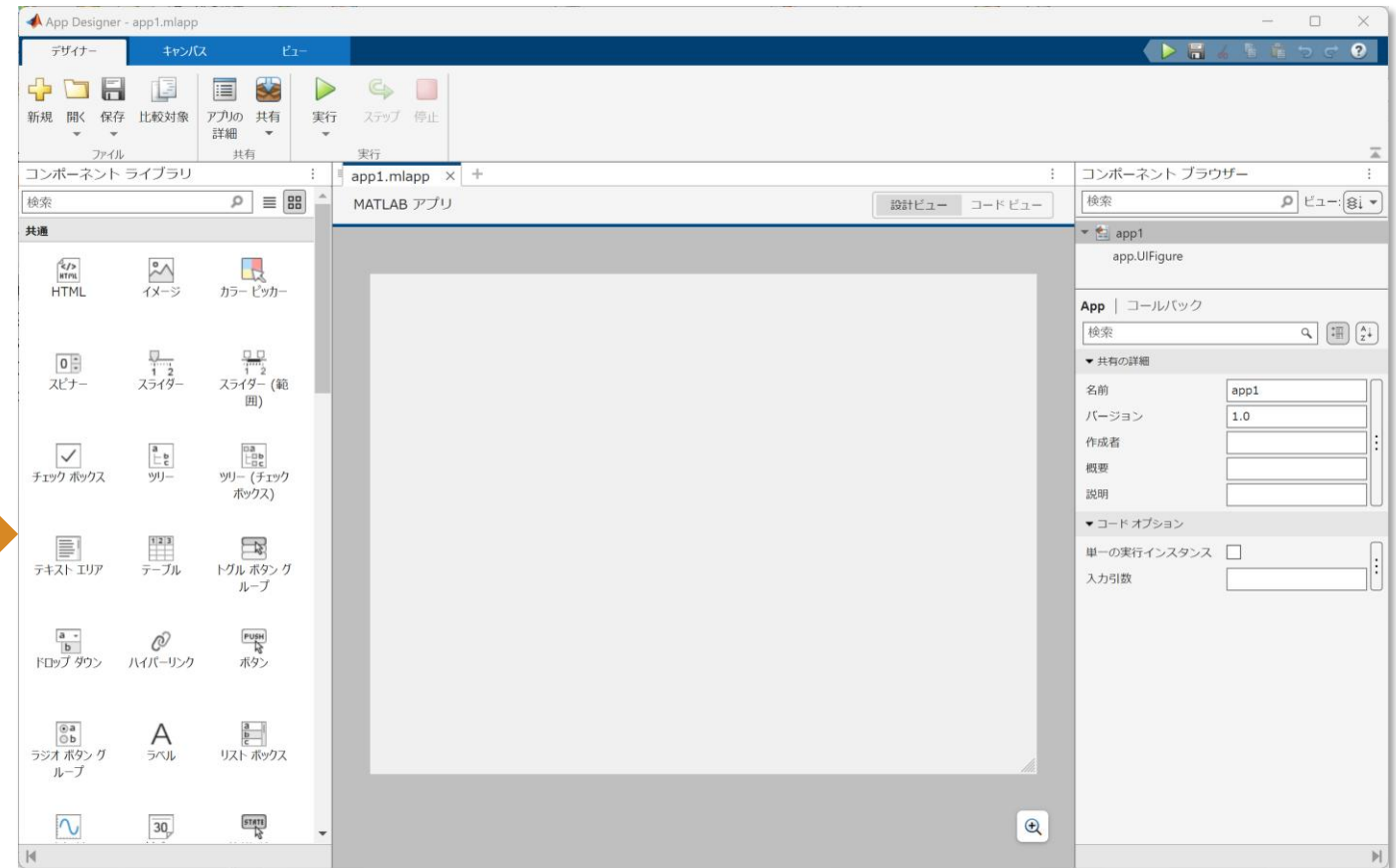
演習 4 GUI アプリの作成

新規アプリの作成

「ホーム」タブの「新規」→
「アプリ」をクリック

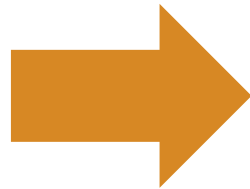


空のアプリが作成されます

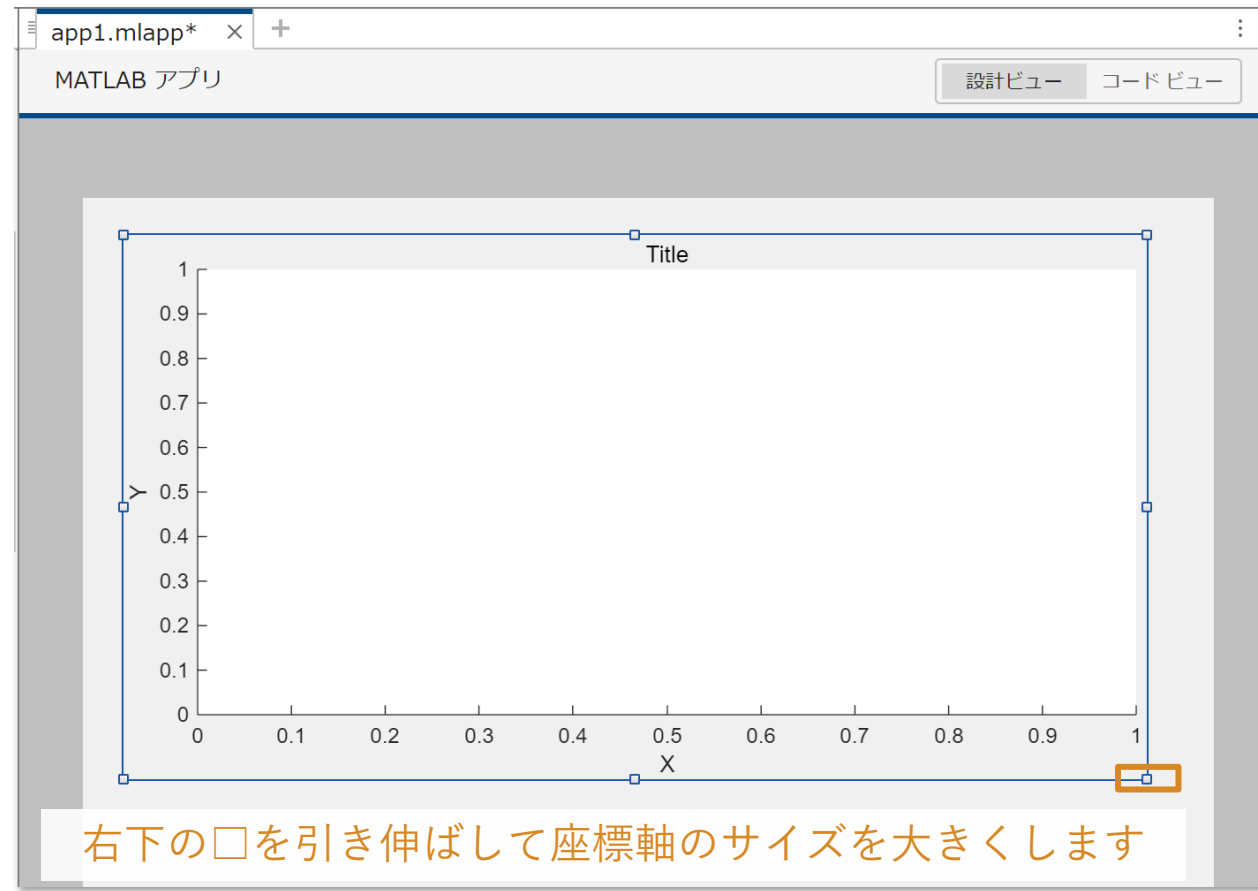


演習 4 GUI アプリの作成

設計ビューでレイアウトを作成

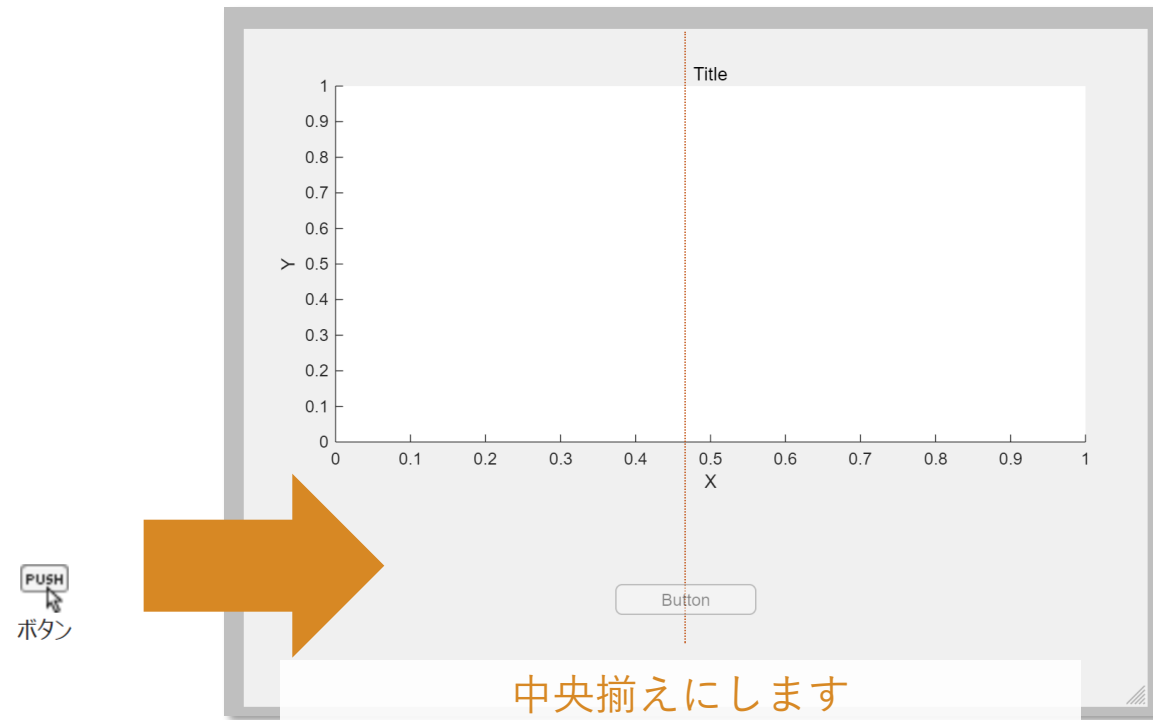


キャンバスに座標軸
コンポーネントを
ドラッグ&ドロップで配置



演習 4 GUI アプリの作成

設計ビューでレイアウトを作成



キャンバスの下側にPUSH ボタン
コンポーネントを
ドラッグ&ドロップで配置

演習 4 GUI アプリの作成

設計ビューでレイアウトを作成



コンポーネントブラウザで
座標軸のコンポーネントを選択

Title.String: 正弦波のプロット

XLabel.String: x

YLabel.String: y

ZLabel.String: 空白



コンポーネントブラウザで
ボタンのコンポーネントを選択

Text: プロット

演習 4 GUI アプリの作成

ボタンにコールバックを追加

設計ビューをクリック

正弦波のプロット

プロット

右クリック

ボタンコンポーネントを右クリックして
「ButtonPushedFcn コールバックの追加」
をクリック

コールバック

- ButtonPushedFcn コールバックの追加
- 既存のコールバックを選択...

並べ替え

コンテキスト メニュー

選択のヘルプ

置き換え

VerticalAlignm

Icon

Fontname

FontSize

FontWeight

FontAngle

FontColor

演習 4 GUI アプリの作成

コードビューで動作を定義

コードビューに移動され、関数ButtonPushedが追加されます

```
% Callbacks that handle component events
methods (Access = private)

    % Button pushed function: Button
    function ButtonPushed(app, event)

end
end
```

コードビューの「ButtonPushedFcn」の中にex4_コールバックの中身.txt をペーストします

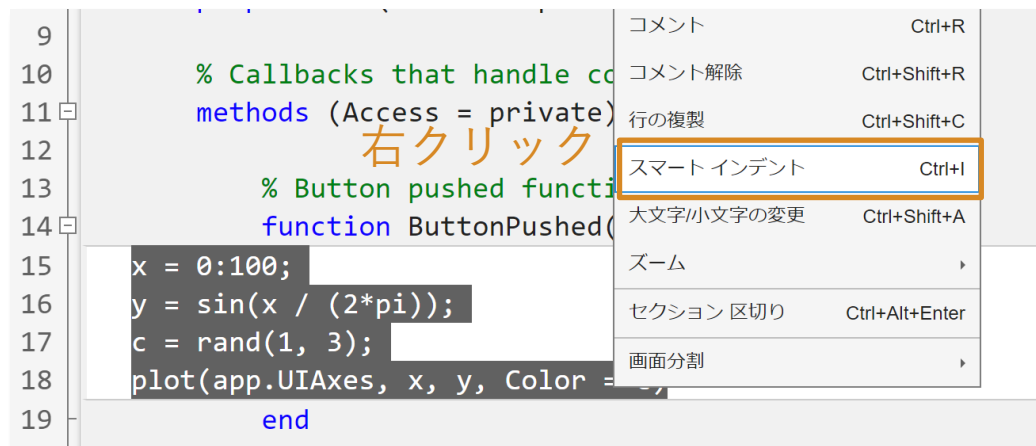
```
function ButtonPushed(app, event)
    x = 0:100;
    y = sin(x / (2*pi));
    c = rand(1, 3);
    plot(app.UIAxes, x, y, Color = c)
end
```

演習 4 GUI アプリの作成

コードビューで動作を定義


行を「Ctrl」キーと「A」キーを同時に押して全行を選択して、
右クリック「スマートインデント」をクリックします

行のインデントが整列されました



The screenshot shows a MATLAB code editor with a right-click context menu open. The menu options include: コメント (Ctrl+R), コメント解除 (Ctrl+Shift+R), 行の複製 (Ctrl+Shift+C), スマートインデント (Ctrl+I), 大文字/小文字の変更 (Ctrl+Shift+A), ズーム, セクション 区切り (Ctrl+Alt+Enter), and 画面分割. The 'スマートインデント' option is highlighted with an orange box. The code in the background is as follows:

```
9  
10 % Callbacks that handle component events  
11 methods (Access = private)  
12 % Button pushed function: Button1  
13 function ButtonPushed(app, event)  
14 x = 0:100;  
15 y = sin(x / (2*pi));  
16 c = rand(1, 3);  
17 plot(app.UIAxes, x, y, Color = c);  
18  
19 end
```

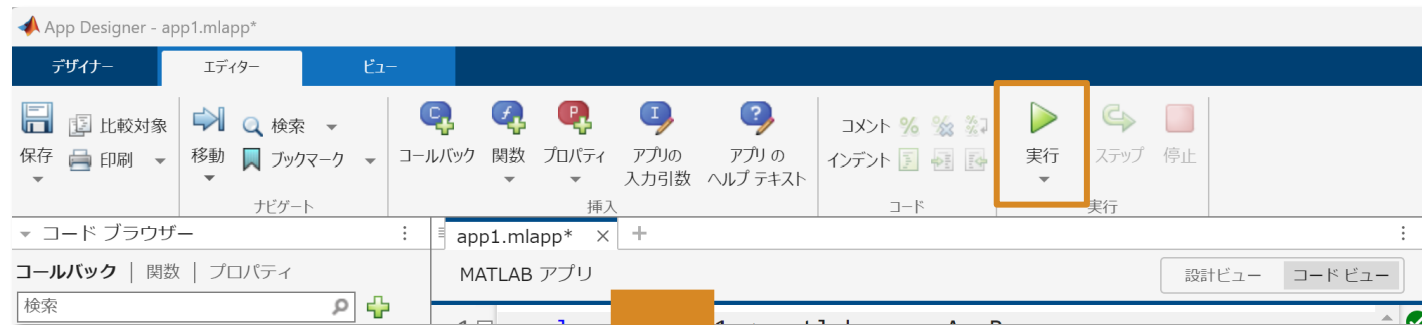


```
function ButtonPushed(app, event)  
    x = 0:100;  
    y = sin(x / (2*pi));  
    c = rand(1, 3);  
    plot(app.UIAxes, x, y, Color = c)  
end
```

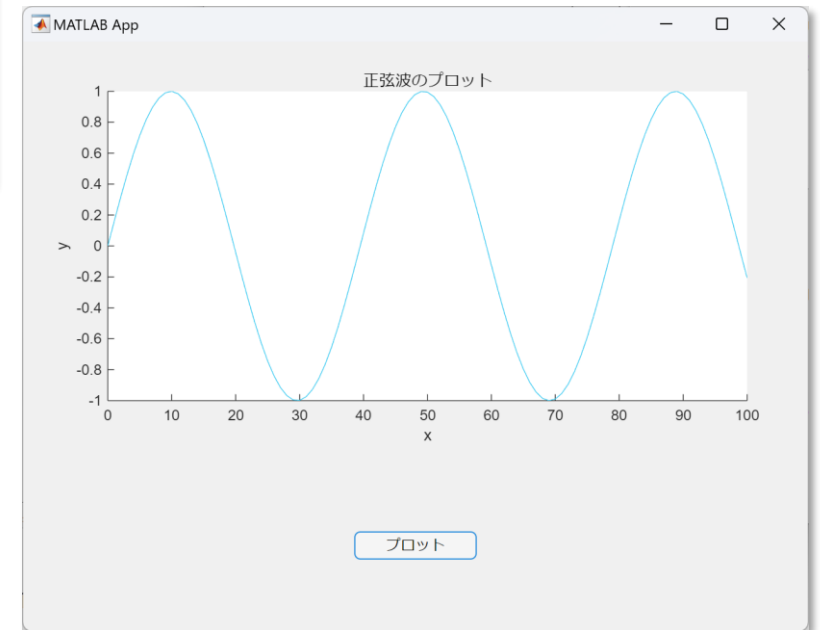
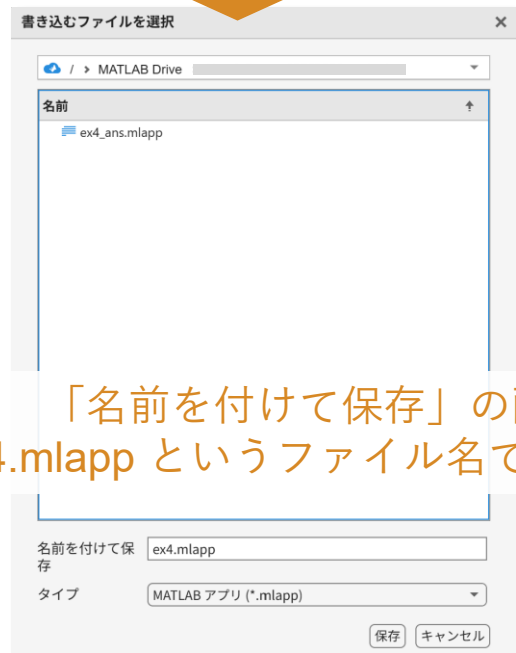
演習 4 GUI アプリの作成

アプリを実行

実行ボタンをクリックします



「名前を付けて保存」の画面で
ex4.mlapp というファイル名で保存します

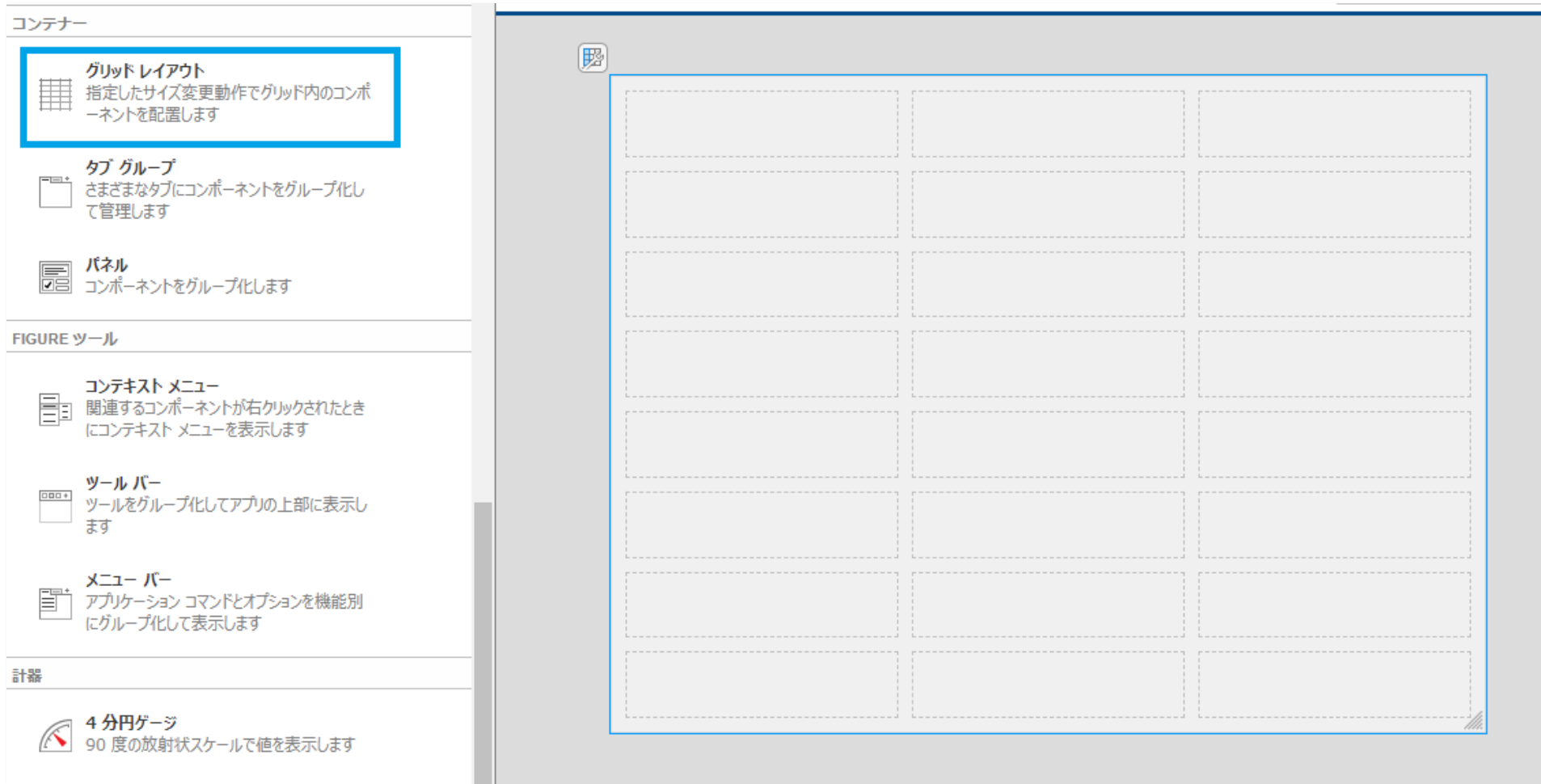


プロットボタンを繰り返しクリック
してみましょう

GUI アプリ作成のコツ

1. レスポンシブデザインにしたい

グリッドレイアウトを使用することで画面サイズに応じて動的に配置を変えられます

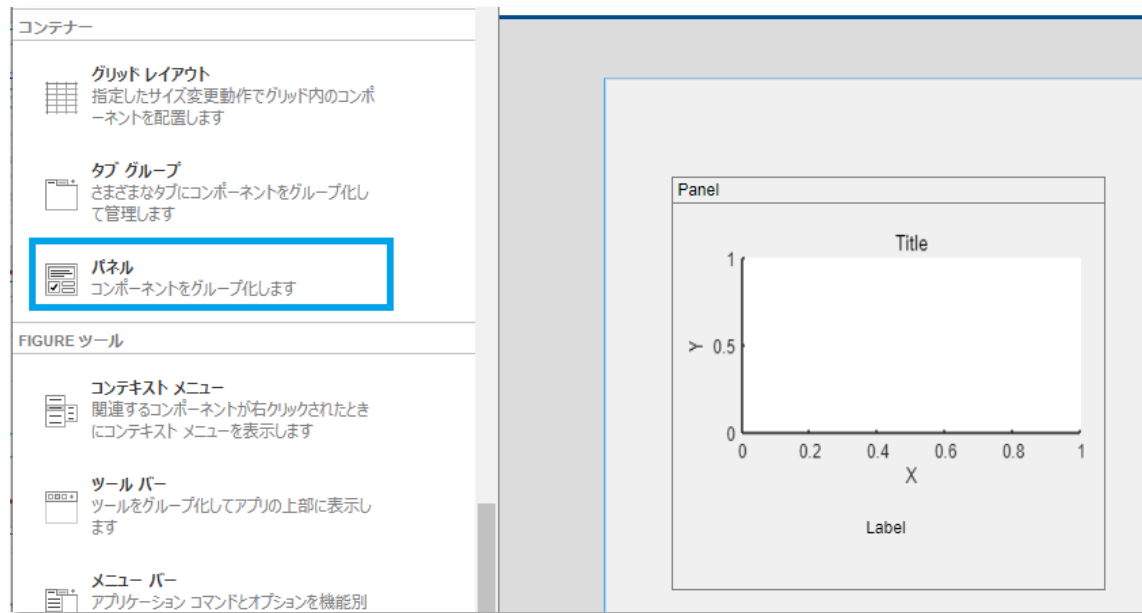


最初から入れても後から入れても

GUI アプリ作成のコツ

2. 導線を良くしたい

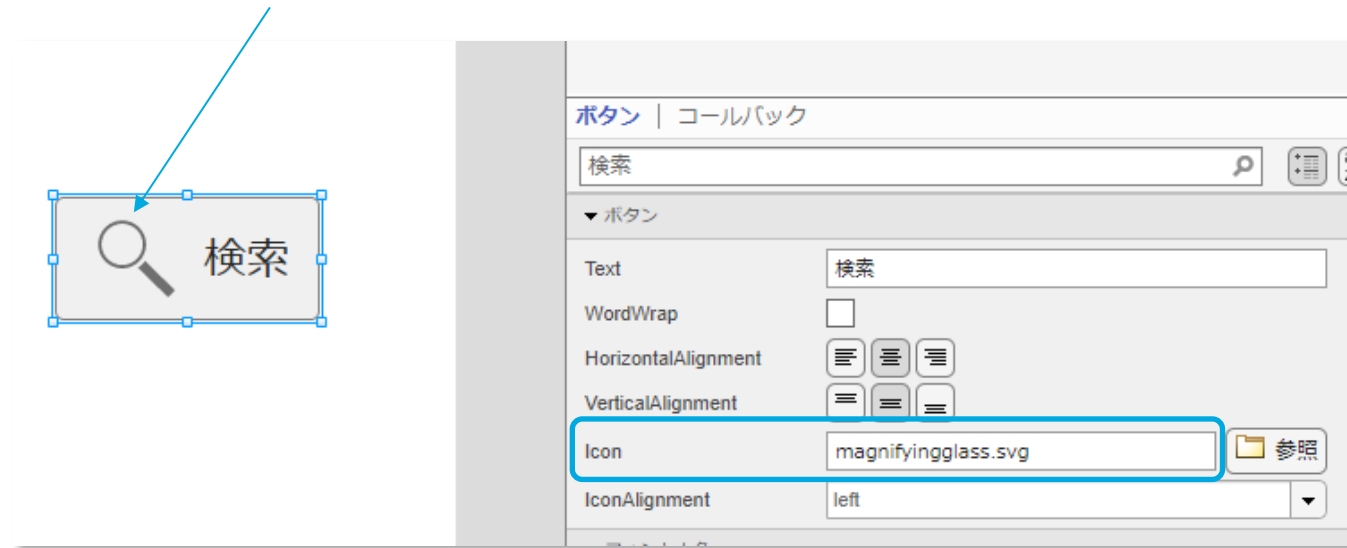
パネルでコンポーネントをまとめられます



GUI アプリ作成のコツ

3. ボタンにアイコンを入りたい

ボタンにアイコンを挿入できます



使用した製品

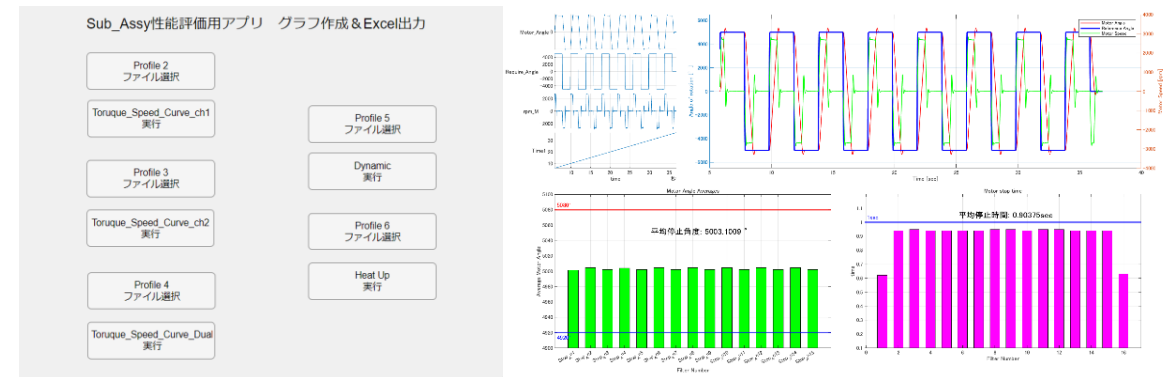
使用した製品	内容
MATLAB	<ul style="list-style-type: none">■ CSV ファイルの読み取り■ スクリプト・ライブスクリプト■ 関数の作成■ 可視化■ GUI アプリの作成 (App Designer)
MATLAB Copilot	<ul style="list-style-type: none">■ 生成AI を使ったコーディング支援

ナブテスコオートモーティブ、データ解析を GUI アプリ化し 作業効率が12倍に

電動化製品の開発段階における複雑かつ膨大化した実験データ解析において、従来は表計算ソフトのマクロを使い、時間をかけて手作業で行っていました。これらの解析作業を MATLAB のコードで実装し、自動化しました。さらに App Designer と MATLAB Compiler を用いて、スタンドアロンアプリとして現場へ展開しました。

主な成果/利点：

- マクロや手作業で行っていた作業を自動化し、作業スピードが12倍以上向上
- 大容量のデータを高速に処理することができ、計算時間自体も大幅に削減
- アプリ化により業務の属人化を防ぎ、ワンクリックで解析が実行可能に
- MATLAB 言語は可読性が良く、解析アプリのメンテナンスが容易に



測定データを選択すると自動で解析が実行され、結果がレポートアウトされるアプリ

“12時間以上要していた解析作業が1時間程度で行えるようになり、作業者の拘束時間が大幅に減りました。プログラム実行中に別の作業を行えるようになったのが嬉しいです。”

- ナブテスコオートモーティブ 第二開発課、片岡 氏

“「MATLAB 初心者の私でも、短期間で自由度の高いカスタム GUI アプリを作成できるようになり、驚きました。データ解析も含めた自作アプリの可能性を感じています。”

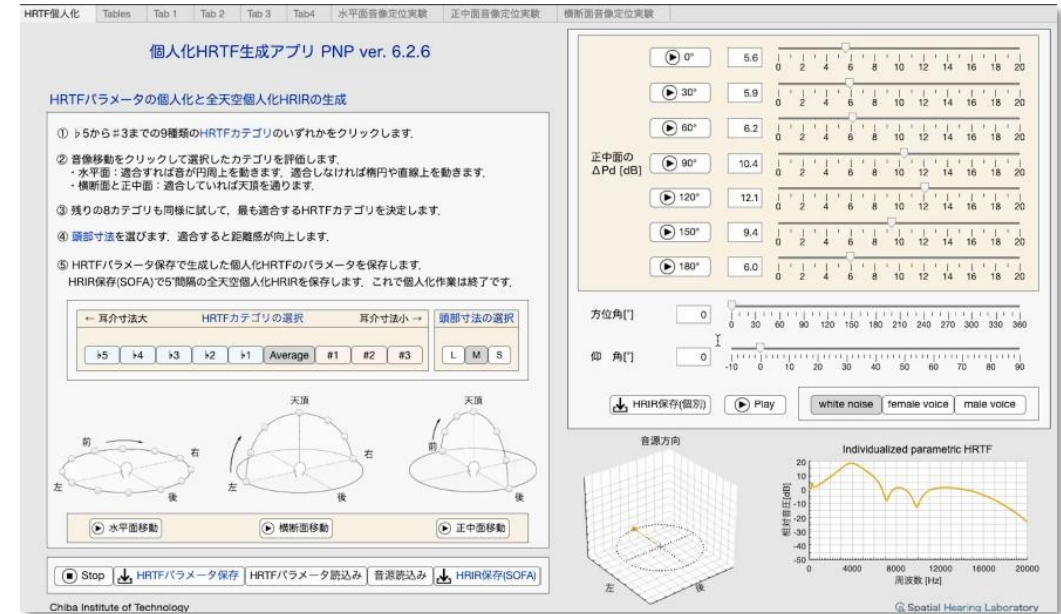
- ナブテスコオートモーティブ 第二開発課、橋本 氏

千葉工業大学、個人化頭部伝達関数生成アプリを開発し、ライブコンサート、スポーツ観戦などでの3次元音響XRに応用

3次元音響再生は、頭部伝達関数(HRTF)をヘッドホンで再生することで実現できます。HRTFには顕著な個人差があり、精度の高い3次元音響の実現には、個人化HRTFを簡単に生成する方法の確立が必要でした。千葉工業大学では、MATLABを利用し、聴取者が簡単な試聴で自分に適したHRTFを生成できるGUIアプリを作成。Webアプリとして公開し、研究開発に役立てました。

主な成果

- 個人に適したHRTFを生成するための音響解析アルゴリズムを手軽に実装
- 聴取者が簡単な視聴により個人化HRTFを取得できるGUIアプリを作成
- Webアプリ化によりMATLABが無い環境でも動作するアプリを作成し、研究成果を広く公開
- アプリから出力されたHRTFを使用してスポーツ観戦やライブコンサートなどの実証実験へ応用



Webアプリとして公開している個人化HRTFの生成アプリ

「MATLABは信頼性の高いライブラリが豊富で音響の解析に役立ちます。また、App DesignerによるGUI設計が非常に容易です。コンサルティングやコーチングのサポート体制も充実しています。世界初の個人化HRTF生成のWebアプリを公開し、多くの方が利用できるようになりました。」

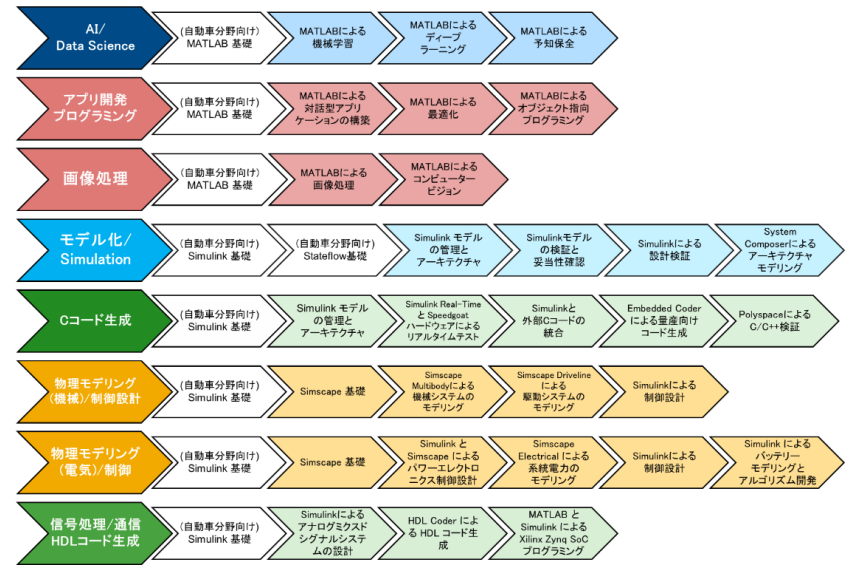
- 千葉工業大学 先進工学部 知能メディア工学科 飯田 一博、中村 風香

学習リソースの紹介



【トレーニング】 目的に合わせた多種多様なコース

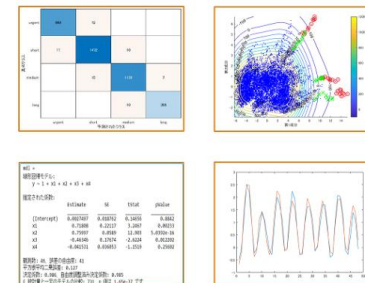
定期開催コースのカリキュラム例



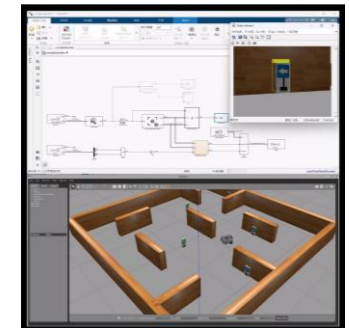
特別開催コースの例

AI・データサイエンス講座

- ①機械学習アルゴリズム・理論入門
- ②回帰分析～統計・機械学習・ディープラーニング



SimulinkとROSを用いたロボットソフトウェア開発



- 定期開催：基礎から応用まで効率的に習得
 - オンライン・対面どちらも開催がございます
 - コース一覧は [こちら](#)
- 特別開催：最新のトレンド技術を習得
 - 特に国内で需要の高い技術分野のコースを日本のトレーニングチームで開発、提供しています
 - 理論・実践を合わせて学習いただけます
 - 詳細は [こちら](#)

【トレーニング】特徴と受講のメリット

- **講師はMathWorks 社のトレーニング専任エンジニア**
 - 製品を使用した開発経験をキャリアとして持つ講師陣です
 - 現場で躓きやすいポイントや、習得のコツをお伝えするのに長けています
- **ツールの主要な機能を効率的に習得**
 - 最新のアップデートに対応した機能・知っていると便利な機能を順序良く学べます
 - 公式コンテンツを、演習形式で実際に手を動かしながら習得できます
 - 受講後のテキスト閲覧・講師への質問も可能です

費用対効果の高い [講師付きトレーニング](#)

144%

トレーニング受講後の MATLAB
使用能力の平均向上率

95%

コースでの学習を実際の仕事に応
用できたと回答した受講者の割合

96%

コースをお勧めできると回答した
受講者の割合



MATLAB 基礎トレーニングコースのご案内

MathWorks製品は、各種専門ツールボックスと連携し、さまざまな産業における開発、革新を促進ツールとして活用されています。MATLABは、その基幹製品であり、応用もしくは専門コースの第一歩になります。

3日間の本トレーニングを受講すると、データのインポート、前処理、解析、可視化の一連のワークフローを学び、複雑なタスクを自動化することができます。MathWorks製品を購入したばかりの方・これから使う方・もっと効果的に使いこなしたい方には必須のコースです。

受講対象者

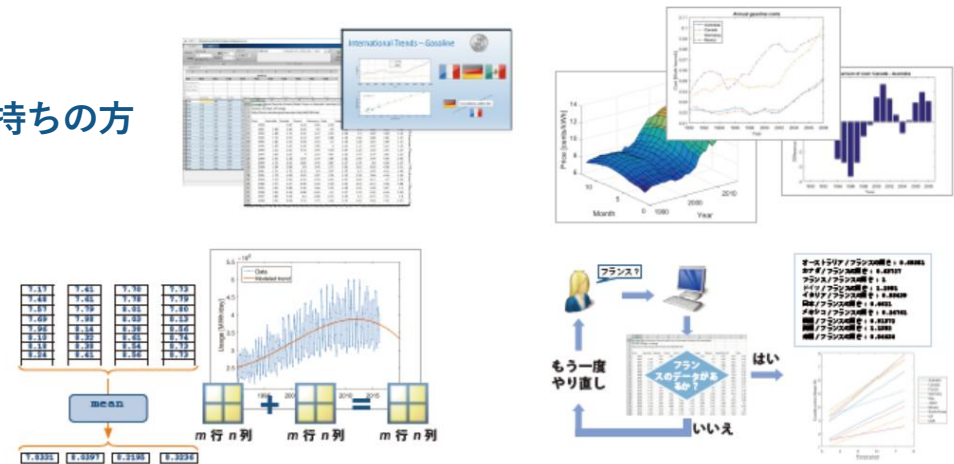
- これからMATLABを使われる方。MathWorks製品を購入したばかりの方
- 我流で学んできたがより効果的にMATLABを使いこなしたい方
- データ解析、機械学習やディープラーニングなどアルゴリズム開発に興味をお持ちの方

受講目標・コース概要

- このコースは例題や演習を実際に行いながら、少しずつ学べるように構成されています。これにより、実用的なテーマに重点を置きながらMATLABのスキルを高めることができます。主にデータ解析、可視化、モデリング、プログラミングなどのテーマがコース内に含まれています

お客様の声

- 今まで利用していなかった機能についても理解を深めることができました
- わかりやすく非常に良かった。もっと若い時に受講したかった



自己学習e-Learning コース

- 入門コース (2時間～3時間/コース)
中断・再開可能、進捗報告や修了証の発行が可能、MATLABアカウントがあればライセンスがなくても実行可能



MATLAB 入門

15 個のモジュール | 2 時間 | 言語
最短でMATLAB の基礎を学びましょう。



Simulink 入門

14 個のモジュール | 2 時間 | 言語
最短でSimulinkの基礎を学びましょう。本コースはSimulinkをインストールすると受講できます。



電気回路シミュレーション入門

7 個のモジュール | 2 時間 | 言語
Simscape で電気回路をシミュレーションするための基礎を学びます。



機械学習入門

6 個のモジュール | 2 時間 | 言語
分類問題のための実用的な機械学習手法の基礎を学びます。



ディープラーニング入門

5 個のモジュール | 2 時間 | 言語
ディープラーニング手法を使用した画像認識を行う方法を学びましょう



強化学習入門

5 個のモジュール | 3 時間 | 言語
強化学習ベースのコントローラを設計するための基礎を学びます。



画像処理入門

6 個のモジュール | 2 時間 | 言語
MATLAB で実用的な画像処理の基本を学びます。



信号処理入門

7 個のモジュール | 1 時間 | 言語
スペクトル解析のための実践に即した信号処理方法を対話形式で説明します。



無線通信入門

6 個のモジュール | 1 時間 | 言語
MATLAB で無線通信リンクをシミュレーションするための基礎を学ぶことができます。



Simscape 入門

9 個のモジュール | 1.5 時間 | 言語
Simscape で物理システムをシミュレーションするための基礎を学びます。



Stateflow 入門

12 個のモジュール | 2 時間 | 言語
Stateflow でステートマシンを作成、編集、およびシミュレーションするための基礎を学びます。



Simulink による制御設計入門

7 個のモジュール | 1 時間 | 言語
Simulink で基礎的なフィードバック制御系の設計方法を学びます。



最適化入門

5 個のモジュール | 1 時間 | 言語
MATLAB で最適化問題を解くための基礎を、問題解決型のアプローチで学びます。

» matlabacademy.mathworks.com/jp

AI Chat Playground で自己学習 (無料)

The screenshot shows the MATLAB AI Chat Playground web interface. At the top, there's a navigation bar with links to MATLAB Help Center, Community, and Learning. Below this is a blue header with tabs for MATLAB Answers, File Exchange, Cody, AI Chat Playground (selected), Discussions, Contests, Blogs, and Others. The main content area is divided into two panels. The left panel, titled 'AI', contains a welcome message from the AI assistant, a disclaimer about the accuracy of the AI's responses, and three example prompts: 'Create a table of 10 landmarks in France and use geoscatter to visualize the table', 'Plot the sine function over a linearly spaced vector of values from 0 to 2n', and 'Determine whether a matrix is sparse'. A 'Shuffle' button is located below these prompts. At the bottom of the left panel is a text input field with the placeholder 'Enter prompt here' and a submit button. The right panel is a code editor with tabs for CODE, ANALYZE, and RUN. The CODE tab is active, showing a single line of code '1'. Above the code editor are 'Copy' and 'Download' buttons. At the bottom of the interface, there's a footer with links to the Trust Center, Privacy Policy, Terms of Use, Application Status, and Contact Us, along with a copyright notice for 1994-2024 The MathWorks, Inc. and a language selector set to '日本'.

MATLAB AI Chat Playground - 1 x

https://jp.mathworks.com/matlabcentral/playground/

MATLAB ヘルプ センター コミュニティ 学習

MATLAB

MATLAB Answers File Exchange Cody AI Chat Playground ディスカッション コンテスト ブログ その他

+ New

Copy Download

AI

Hello! Whether you're checking out MATLAB for the first time or have been using it for years, I'm here to answer your questions and provide coding tips.

Please keep in mind that AI sometimes writes code and text that seems accurate, but isn't. AI does not yet have knowledge of features delivered after October 2023 and only limited knowledge of Simulink and specialized toolboxes. This is a space for experimentation. Try it, verify any resulting code, and kindly give or on the results to help improve the responses.

✦ Create a table of 10 landmarks in France and use geoscatter to visualize the table

✦ Plot the sine function over a linearly spaced vector of values from 0 to 2n

✦ Determine whether a matrix is sparse

Shuffle

Enter prompt here

Generated code and text might be inaccurate. Validate before use. [About](#) [Discuss](#)

CODE ANALYZE RUN

1

トラストセンター | 商標 | プライバシー ポリシー | 違法コピー防止 | アプリケーション ステータス | ご利用条件 | Contact Us

© 1994-2024 The MathWorks, Inc.

日本

<https://jp.mathworks.com/matlabcentral/playground/>

YouTube チャンネル

[illegible]

<https://www.youtube.com/c/MATLABJapan>



© 2026 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.