

Graphing in 3D, Multidimensional Arrays & Structures

LAB 7

ENGG100 - Spring 2024

Lab 7 Objectives

- To be able to graph in 3D using meshgrid and surf functions in MATLAB
- Understand, create and manipulate multidimensional arrays
- Understand, create and manipulate structures

Using Meshgrid & Surf - Plotting in 3D

- meshgrid() function is used to create 3D grid coordinates, e.g.

```
x = 4:1:9; % size of x is 1x6
```

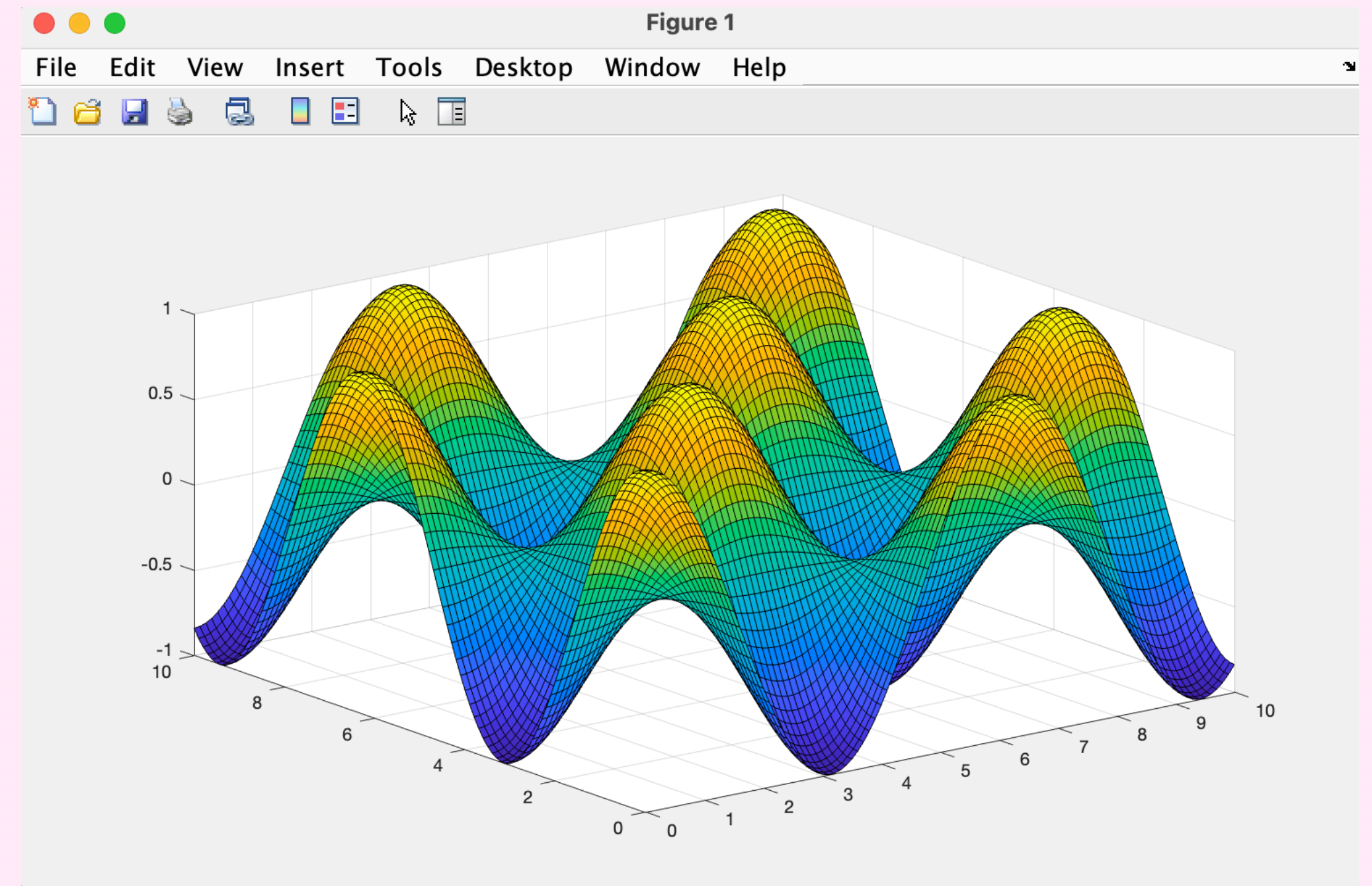
```
y = 7:1:10; % size of y is 1x4
```

```
[X,Y] = meshgrid(x,y) % resulting meshgrid  
matrix would be 6x4
```

- The size of the resulting meshgrid is x number of columns from x and y number of rows from y
- Once a meshgrid is computed, the surf() function is used to create a three-dimensional surface plot that has a solid edge and solid face colors, e.g. to plot the above, we would first compute the z coordinates and then plot:

```
Z = cos(X) .* cos(Y) % Make sure to use the  
element-wise operator when multiplying (.*)
```

```
surf(X,Y,Z) % This will plot a 3D graph
```

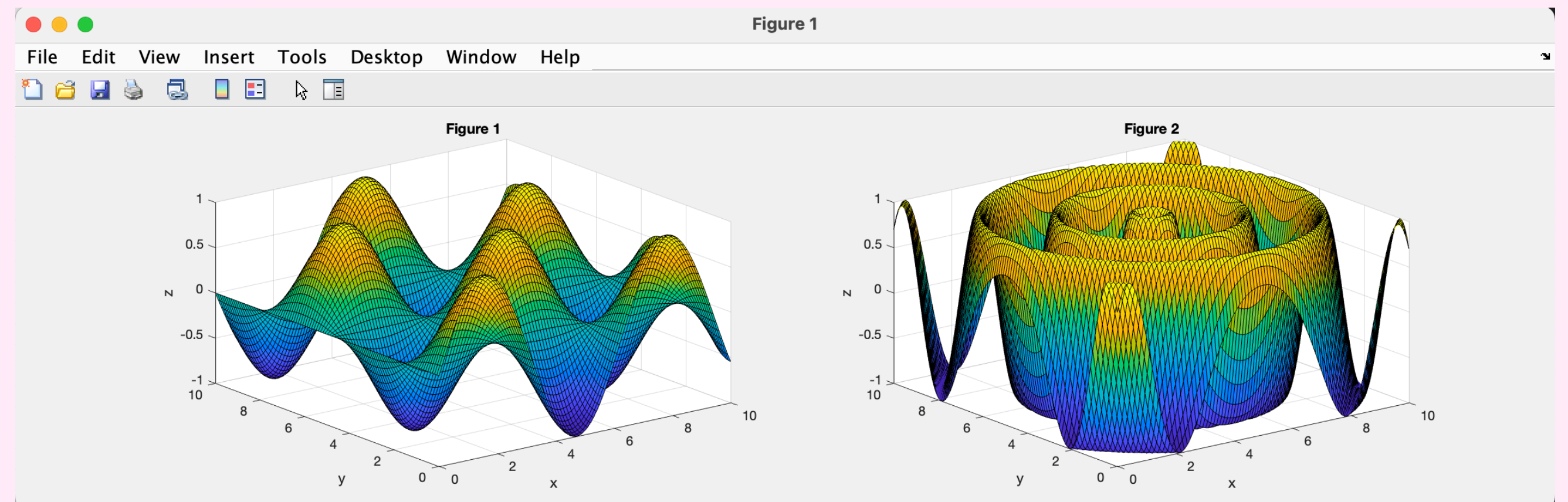


TASK 1 - SCRIPT

- Using the surface plot approach for plotting functions in two variables, plot the below two functions:

$$z = \sin(x) \cdot \cos(y)$$
$$z = \sin(3\sqrt{(x-5)^2 + (y-5)^2})$$

- Use a resolution of at least 0.1 between points, from 0 to 10 for x and y coordinates
- Plot both functions side by side using subplots on the same figure window
- Ensure all your axes (x, y & z) are labelled and each figure has a title
- Use element-wise operators (.*) or .^ when multiplying and squaring vectors
- Make sure your code is **commented** properly
- Results should look like the figure on the right —>

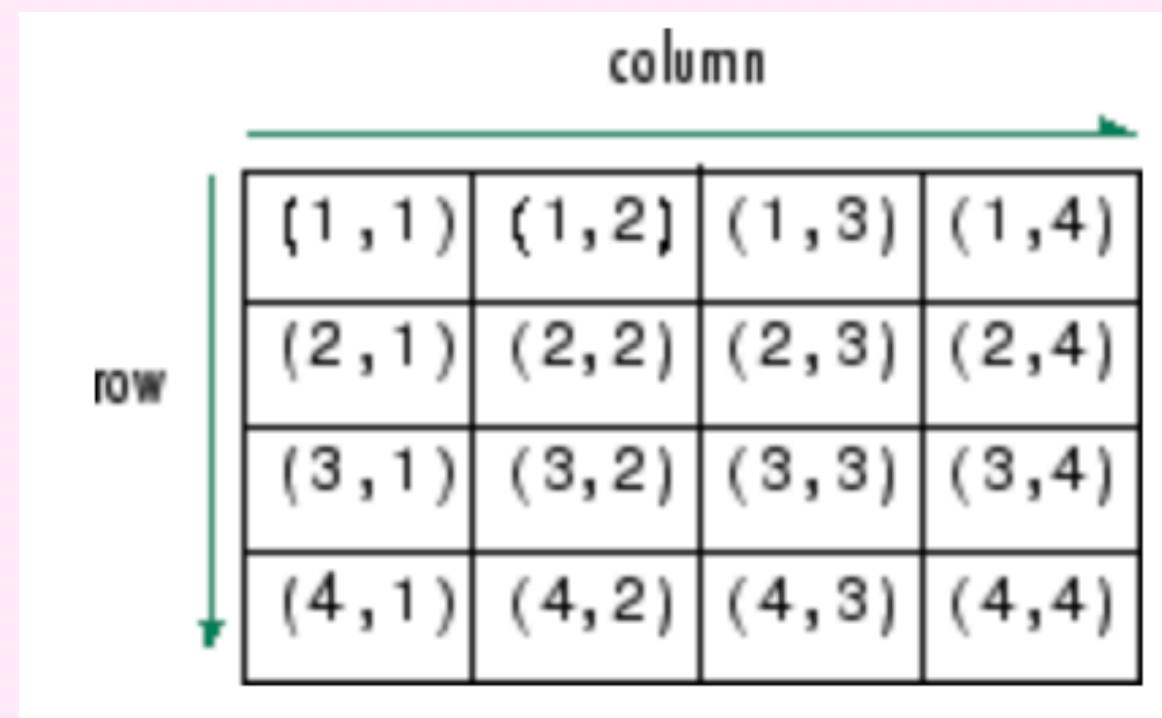


Multidimensional Arrays

- Multidimensional array in MATLAB is an array with more than 2 dimensions. In a matrix, the two dimensions are represented by rows and columns, where each element can be identified by the row index and columns index
- Multidimensional arrays are an extension of the 2D matrices and use additional subscripts for indexing, e.g. a 3D array would use 3 subscripts
- A multidimensional array can be created by first creating a matrix and then extending it:

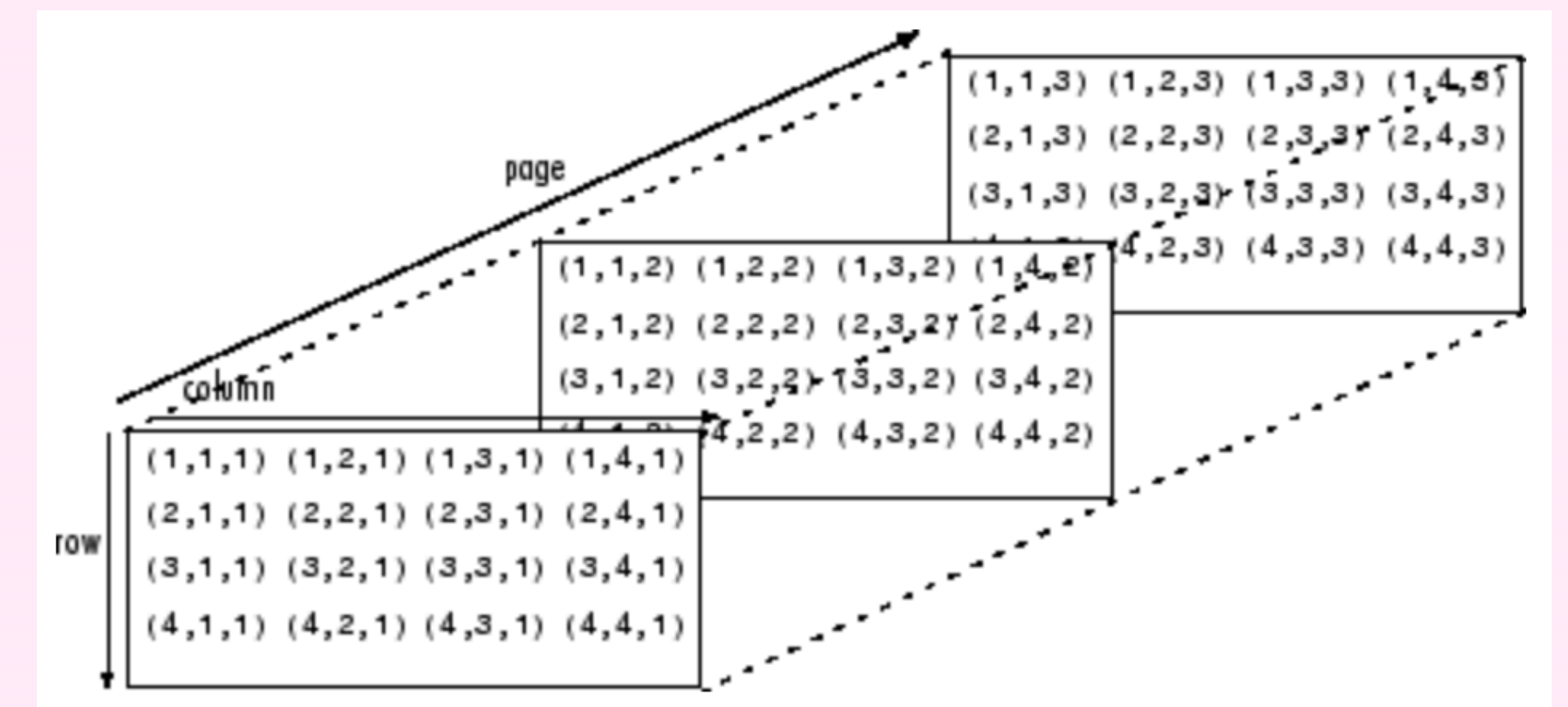
```
A = [1 2 3; 4 5 6; 7 8 9];
```

```
A[:, :, 2] = [10 11 12; 13 14 15; 16 17 18];
```



A 4x4 matrix diagram illustrating row and column indexing. The vertical axis is labeled 'row' with a downward arrow, and the horizontal axis is labeled 'column' with a rightward arrow. The elements are arranged in a grid with coordinates (row, column) in parentheses.

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)



TASK 2 - SCRIPT

- Create 2 multidimensional arrays:
 - array_A = 10x10x10 numerical array (3D)
 - array_B = 5x5x5x5 numerical array (4D)
- Each value inside each array should correspond to the multiplication of its indices (use nested for loops)
- E.g. for the array_A, the number at location (1,2,3) should be $1 \times 2 \times 3 = 6$
- Make sure your code is **commented** properly
- **For your report:** Show a screenshot of your code, along with the below location values:
 - Test values to show for array_A —> **array_A (1,4,5) = 20, array_A (4,7,9) = 252, array_A (9,10,3) = 270**
 - Test values to show for array_B —> **array_B(3,3,4,1) = 36, array_B(4,1,2,3) = 24, array_B(1,5,3,2) = 30**

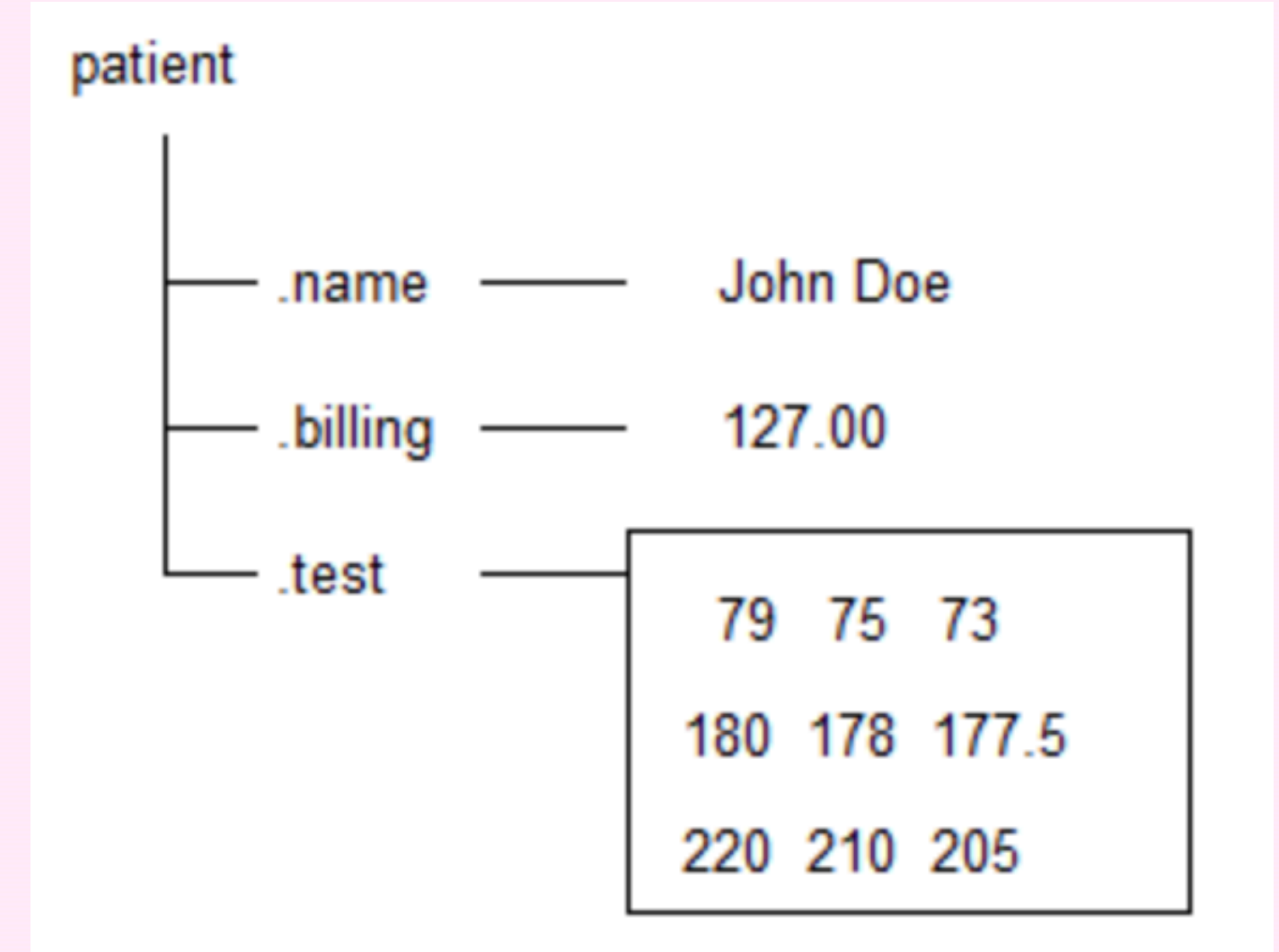
Creating Structures

- A structure array is a data type that groups related data containers using fields
- Each field can contain any type of data
- Information can be accessed inside structures using dot notation of the form `structName.fieldName`
- For example, to create the structure shown, we would code:

```
patient.name = 'John Doe';  
patient.billing = 127;  
patient.test = [79 75 73; 180 178 177.5; 220 210 205];
```

- To create multiple structures within a structure, we would code:

```
patient(1).name = 'John Doe';  
patient(1).billing = 127;  
  
patient(2).name = 'Mary James';  
patient(2).billing = 456;
```



TASK 3 - SCRIPT

- Create an array of structures corresponding to a list of students in a class. Each student structure must contain the below fields:
 - First Name (`first_name`)
 - Last Name (`last_name`)
 - Student Number (`student_no`)
 - Engineering Major (`eng_major`)
 - Test Mark (`marks`)
- Finally, show how you would output just the test marks for all the students (*hint: use a for loop*)

First Name	Last Name	Student Number	Engineering Major	Test Mark
Jane	Adams	1234567	Computer	89
Philip	Spencer	8910111	Mechatronics	87
Thomas	Evans	2131415	Chemical	94
James	Irwin	1617181	Civil	77
Rose	Baker	9202122	Mechanical	96

```
Command Window
Marks for Jane : 89
Marks for Philip : 87
Marks for Thomas : 94
Marks for James : 77
Marks for Rose : 796
```

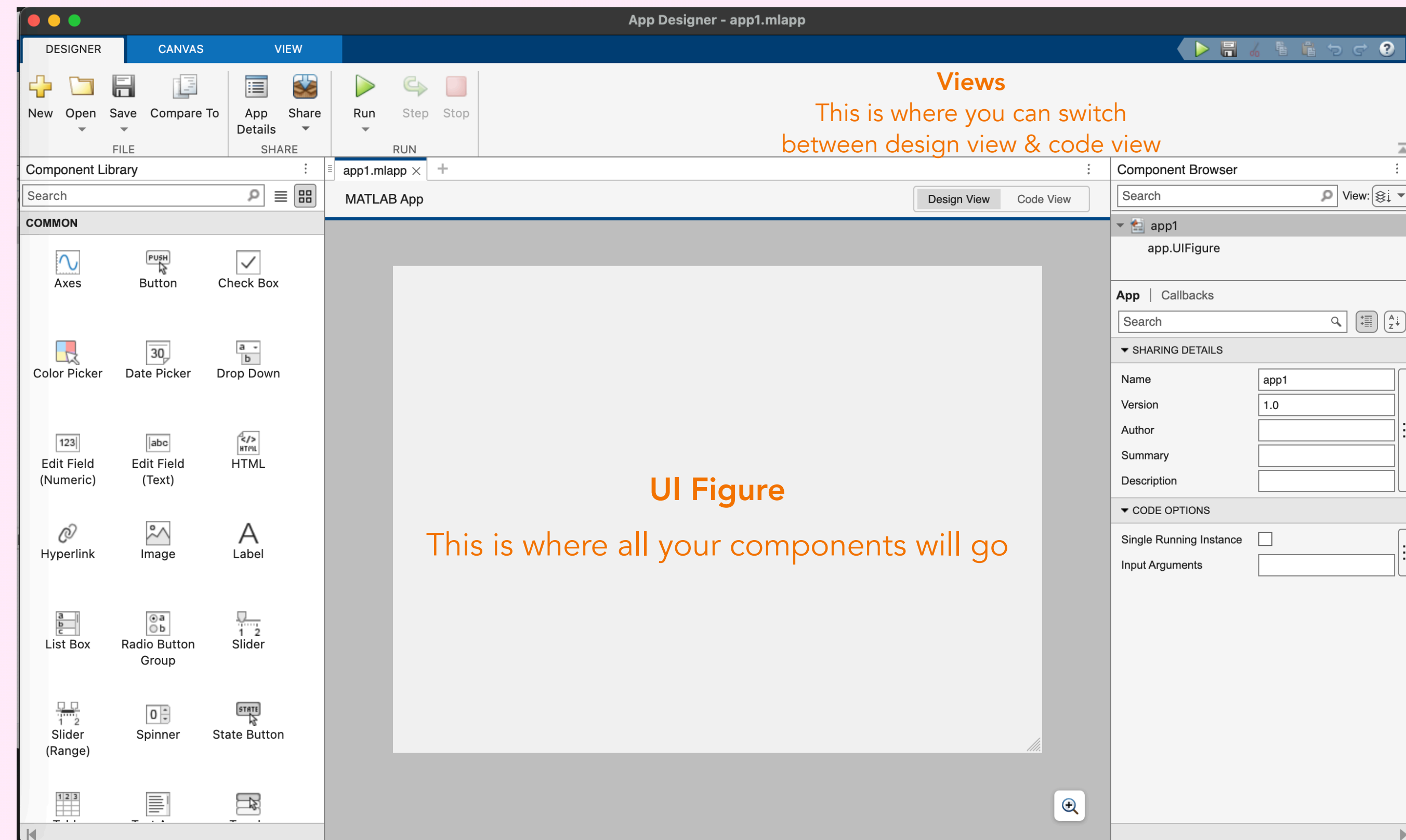

MATLAB App Designer

MATLAB's App Designer allows you to create applications by dropping visual components to lay out the design of your graphical user interface (GUI) and use the integrated editor to quickly program its behavior.

The App designer can be launched by going to the **Apps** tab > **Design App** or by typing **appdesigner** in the command window.

Component Library

Build apps with standard components such as buttons, check boxes, trees, and drop-down lists



Component Browser

Shows all the components inside your UI Figure. This is also where you can add "callbacks" where you can program a certain component to do an action when triggered

Additional Task - App Designer

Follow the steps below to design a basic app where the text "Hello World" will be displayed when a button is clicked.

1. Launch App Designer in MATLAB and start with a blank layout
2. From the component library, place 2 components in your UI window - "**Edit Field (Text)**" & "**Button**"
3. Delete the label from the "Edit Field (Text)" component, so you are just left with the input field
4. Resize your UI window appropriately to fit both these components and place them one above the other
5. Right click on the "**Button**" component > **Callbacks** > **Add ButtonPushedFcn callback**, and this will take you to the Code View automatically where you will add the code to do something when the button is pushed
6. In this function, you will type the name of the "**Edit Field (Text)**" component (you can see this in the component browser), and then add **.Value** at the end as we will be changing the value of the input field when the button is pressed. e.g. **ComponentName.Value**
7. This value should equal to the string "Hello World!" (Don't forget to add a semicolon at the end of your line)
8. When done, click on **Run**, save your application and test your GUI!

