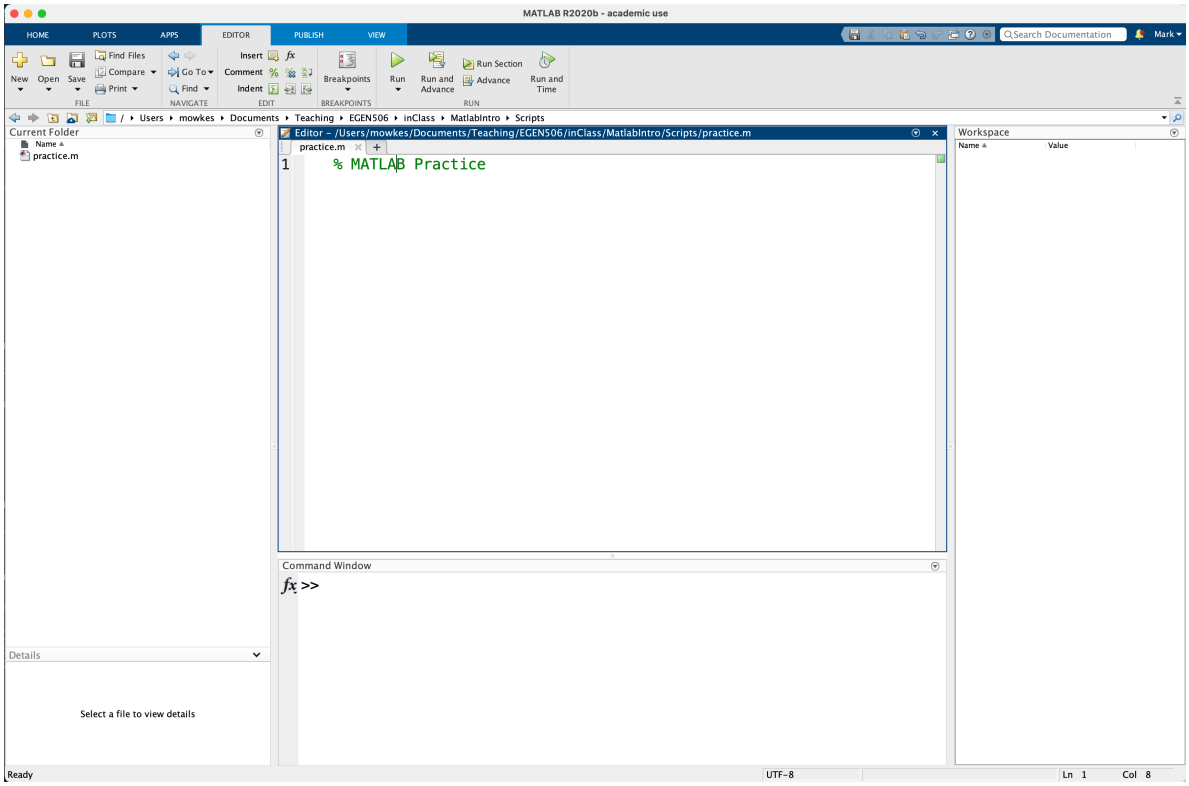
Step 1: Introduction to Matlab

- Open MATLAB and identify the different parts of the interface including
  - Current Folder - lists the files that are in the folder, any functions within these files can be run
  - Editor - (you may need to click on `New > Script` to see this). The Editor is used to write scripts or functions that can be run over and over again
  - Command Window - displays outputs of scripts and can be used to run code one time
  - Workspace - displays any variables that are currently defined



- Use Matlab as a calculator (use the `Command Window`)
  - Compute $\sqrt{5^2 + 3^2}$

```
>> sqrt(5^2 + 3^2)
```

  - Compute $\cos(\pi)$

```
>> cos(pi
```

  - Compute $\cos(180^o)$

```
>> cosd(180)
```

- Defining variables
  - Set the variable `a` to 5

```
>> a=5
```

  - *Note that in code you take the output of what is on the right of the equals sign and assign that to the variable on the left of the equals sign. So, the previous command* `a=5` *works, but* `5=a` *does not!*
  - Set a variable `b` to `5*a`

```
>> b=5*a
```

  - Redefine the variable `a` to be `b-3`

```
>> a=b-3
```

  - *Variable names can be almost anything, but must start with a letter and only contain letters, numbers, and underscores. Variables should be chosen to not be the same functions (e.g. do not use cos as a variable). Try to choose names that have meaning and make sense.*

- Working with arrays
  - Create an array of numbers $a = [0, 2, 4, 6, 8, 10]$

```
>> a=[0,2,4,6,8,10]
```

  - Add 5 to each entry in a and store the output in b

```
>> b=a+5
```

  - Add a to b and store in c

```
>> c = a + b
```

  - Check the length of a and b

```
>> length(a)
>> length(b)
```

  - Check the size of a and b

```
>> size(a)
>> size(b)
```

  - Compute the transpose of b, store in c, and check the size of c

```
>> c = b'
>> size(c)
```

  - Compute $a * c$

```
>> a*c
```

  - Compute $c * a$

```
>> c*a
```

  - Compute $a * c$

```
>> a*c
```

  - Try computing $a * b$

```
>> a*b
```

  - Multiply each element in a by each element in b

```
>> a.*b
```

    Note the "."!

  - Create an array of integers

```
>> i=1:20
```

  - Create an array of equally spaced numbers (try changing these numbers!)

```
>> x=linspace(0,5,10)
```

  - Access the fifth number in x

```
>> x(5)
```

    Note that the index must be a positive integer (1,2,3,...)

  - Access the first through the third numbers in x

```
>> x(1:3)
```

- Getting Help
  - You can quickly learn about any MATLAB function by typing `help FUNCTION_NAME`. For example,

```
>> help sort
```

  - You can also go to `Help > Documentation` to browse all the help documents.
  - Use to help to figure out how to create a 5×1 array of random numbers and then sort the numbers in assending order.
- Scripts and Functions
  - Scripts are codes that can easily be run over and over again. For example, you may have a script that loads some data you have collected, computes useful statistics, and makes plots.
  - Funtions are codes that take inputs, do something, and provide outputs. For example, `sort` is a function that takes in an array and sorts it in assending order, and then outputs the sorted array.
  - Comments - Add comments to your scripts and functions to make them easier to understand. Comments do not change how the code works, just how easily it is to understand. Add a comment by placing a `%` sign at the beginning of the line
  - Controlling outputs - Add a semicollin (`;`) at the end of a line of code to keep MATLAB from displaying the output. For example

```
a=5+5
b=3+3;
```

- Loops are used to perform a similar operation over and over again.
  - Write the loop below in a script. Click `Run` to run the script and see the output.

```
for i=1:20
  disp(i)
end
```

  - Loops can be used to compute sums. For example compute $a = \sum_{i=1}^{1} 0i^2$

```
a=0;
for i=1:10
  a = a + i^2;
end
disp(a)
```

  - Practice, compute π using the formulat $\pi = \sum_{k=0}^{\infty} \sqrt{12}\frac{(-3)^{-k}}{2k+1}$
  - While loops are used to do something over and over until a critera is met. For example,

```
i=0;
while i<10
  i = i+2;
  disp(i)
end
```

  - Loops can be used to do something to the entries in an array

```
N=5; % Number of elemnts in array
x=linspace(0,10,N); % Create array
for i=1:N
  y(i)=x(i).^2;
end
disp(y)
```

    Preallocating can make your code faster. Add `y=zeros(size(x)` *before the for loop to tell MATLAB how big the y array will be.*

- Debugging a code is the process of finding errors.
  - Think about simplifying the problem to something that you know the answer to and working up to a more complicated problem.
  - Print off the values of variables
  - Add `Breakpoints` to stop the code and then step through the code check each line does what you expect it should
- Making plots
  - A simple plot requires two arrays that contain the x and y values.

```
x=0:0.1:4
y=cos(x)
plot(x,y)
```

  - Add axis labels using `xlabel` and `ylabel`, for example

```
xlabel('x')
ylabel('cos(x)')
```

  - Add another curve to the figure using `hold on`

```
y2=sin(x);
hold on
plot(x,y2)
```

  - Add a legend

```
legend('cos(x)','sin(x)')
```

  - Make the font size of the axis bigger

```
set(gca,'Fontsize',20)
```

- Cleaning up
- `clear` - clears all variables from memory (very useful at the start of scripts)
- `clc` - clears the command window
- `close all` - close all open figures

Step 2: Additional Background

Work through the tutorial at
`https://www.mathworks.com/help/matlab/getting-started-with-matlab.html`

Step 3: Practice

(a) Write a script that computes the first 20 terms of the Fibonacci sequence $[0, 1, 1, 2, 3, 5, \dots]$. Note that after the first two terms $F_1 = 0$ and $F_2 = 1$ the n^{th} term can be computed using the recursion relation $F_n = F_{n-1} + F_{n-2}$. *Hint: Start with* `F=[0,1]`. *Then use a loop* `for n=3:20` *to compute the remainder of the terms.*

(b) Create a plot of the first 20 terms of the Fibonacci sequence. *Hint: the x-axis should be the numbers 1-20 and an array with these numbers can be created with* `x=1:20`.

(c) Create the array

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

  (a) Explain the output of `A*A`

  (b) Explain the output of `A*A'` *Note that* `'` *means transpose.*

  (c) Explain the output of `A.^2`

(d) Use a loop to compute 5! (factorial)

(e) Compute $1/x$ for $x = 0.25$ using the Taylor series

$$\frac{1}{x} = \sum_{i=0}^{\infty} (-1)^i (x-1)^i$$