

# Introduction to MATLAB

Mai Nguyen

# Class overview

- **Meeting time:** Wed Fri, 2-3:30 pm on Zoom
  - June 17 & 19: 3-4:30 pm for RCR
  - No class July 3
- **Instructor:** Mai Nguyen (mnguyen@princeton.edu), office hours by appointment
- **Structure:** Weeks 1-4 on programming fundamentals, Weeks 5-8 on data analysis pipeline & advanced topics
- **Weekly assignments:** 2-3 hours, assigned Friday, due Tuesday 6pm; ungraded but will receive feedback; weekly code review

# About me

- Undergrad at Stanford, majored in psych & bio, research human fMRI
- PhD in Psychology at Princeton, just graduated in June
  - Dissertation on shared and idiosyncratic neural representation of information during human communication
  - fMRI, matlab, python
  - First time teaching programming and on Zoom

# Who is this course for?

- Beginners with minimal experience in programming
  - If you've used Matlab before and comfortable with manipulating ND arrays, using control flow, writing fxs, feel free to skip Weeks 1-4 and rejoin us later
- Your lab or field primarily uses MATLAB
- You want an easy, all-in-one programming environment

**Seriously consider taking Python instead or in addition to this class**

# Matlab versus Python

## Matlab

- Used in academia and engineering
  - Lots of libraries for specialized tasks
  - Very good documentation and testing
  - Does a lot to “help” you
- 
- Very expensive \$\$\$
  - Updated once a year -> slower development
  - Not widely used in industry
  - Bad at more sophisticated stats

## Python

- Widely used in industry for data science, machine learning, starting to be more widespread in academia
  - Free to use, many open source libraries
  - Rapid development of new libraries
- 
- Have to cobble together environment from libraries
  - Libraries sometimes badly documented, hard to install
  - Also bad at more sophisticated stats

# Today

- Startup MATLAB
- Write your first program
- Functions
- Matlab help
- Matlab as a calculator
- Variables, functions
- Scripts

# MATLAB Online

← → ↻ 🏠

matlab.mathworks.com

☆ 🗨️ 📄 | 👤 ⬆️

HOME PLOTS APPS

📄 + New Script

📄 + New Live Script

✚️ + New

📄 ⬆️ Download

📄 ➡️ Upload

📄 ➡️ Go to File

🔍 Find Files

📄 ⬆️ Import Data

🗑️ Clear Workspace

⭐ Favorites

📄 ⬆️ Clear Commands

📄 Layout

⚙️ Preferences

📄 Parallel

📄 Add-Ons

❓ Help

👤 Community

🗨️ Feedback

📺 Learn MATLAB

Search Documentation 🔍 Mai ▾

FILE VARIABLE CODE ENVIRONMENT RESOURCES

🔍 / > MATLAB Drive > Location in file system

▼ CURRENT FOLDER

📄 Published (my site) ...

▼ WORKSPACE

Name	Value	Size
------	-------	------

>>

Command window - execute code here

# Write your first program

**In the command window write:**

```
>> fprintf('Hello World!')
```

**This prints out to the command window:**

```
Hello World!
```



# Let's break this down

Built-in MATLAB **function** for print **strings**

Use parentheses to **pass** arguments

Indicates a command

```
>> disp('Hello World!');
```

Output of the command

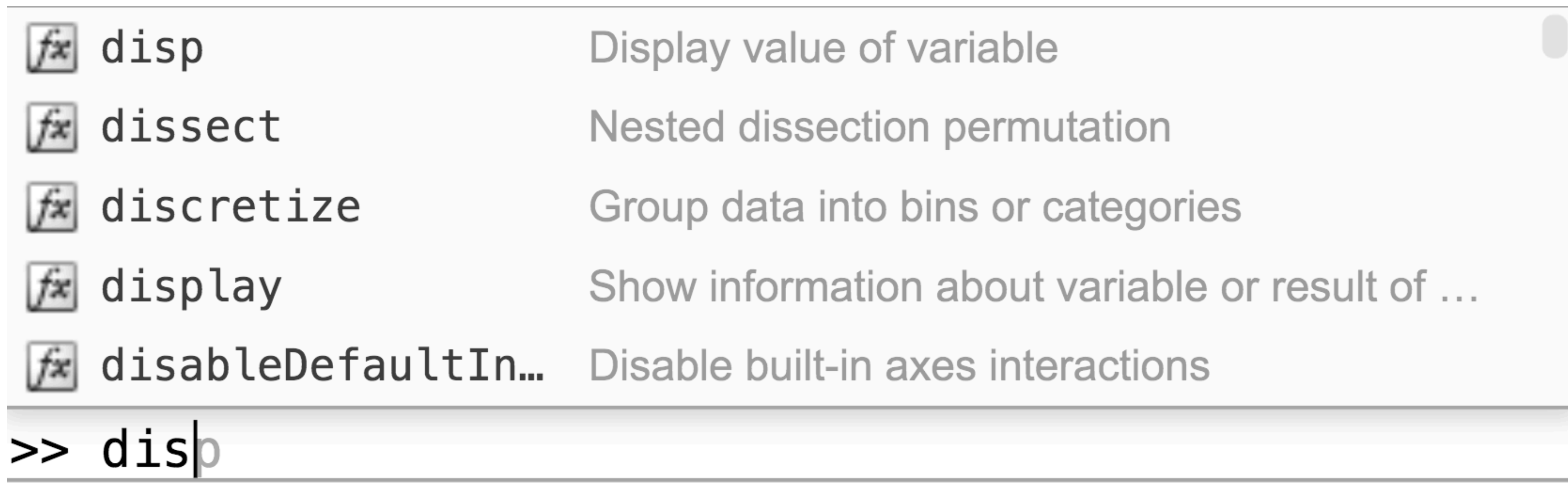
```
Hello World!
```

**argument** that is **passed**  
to the function; thing that  
is printed






Optional semicolon  
to end line and  
surpress output

# MATLAB helps you a lot

- Tab-complete and list of options



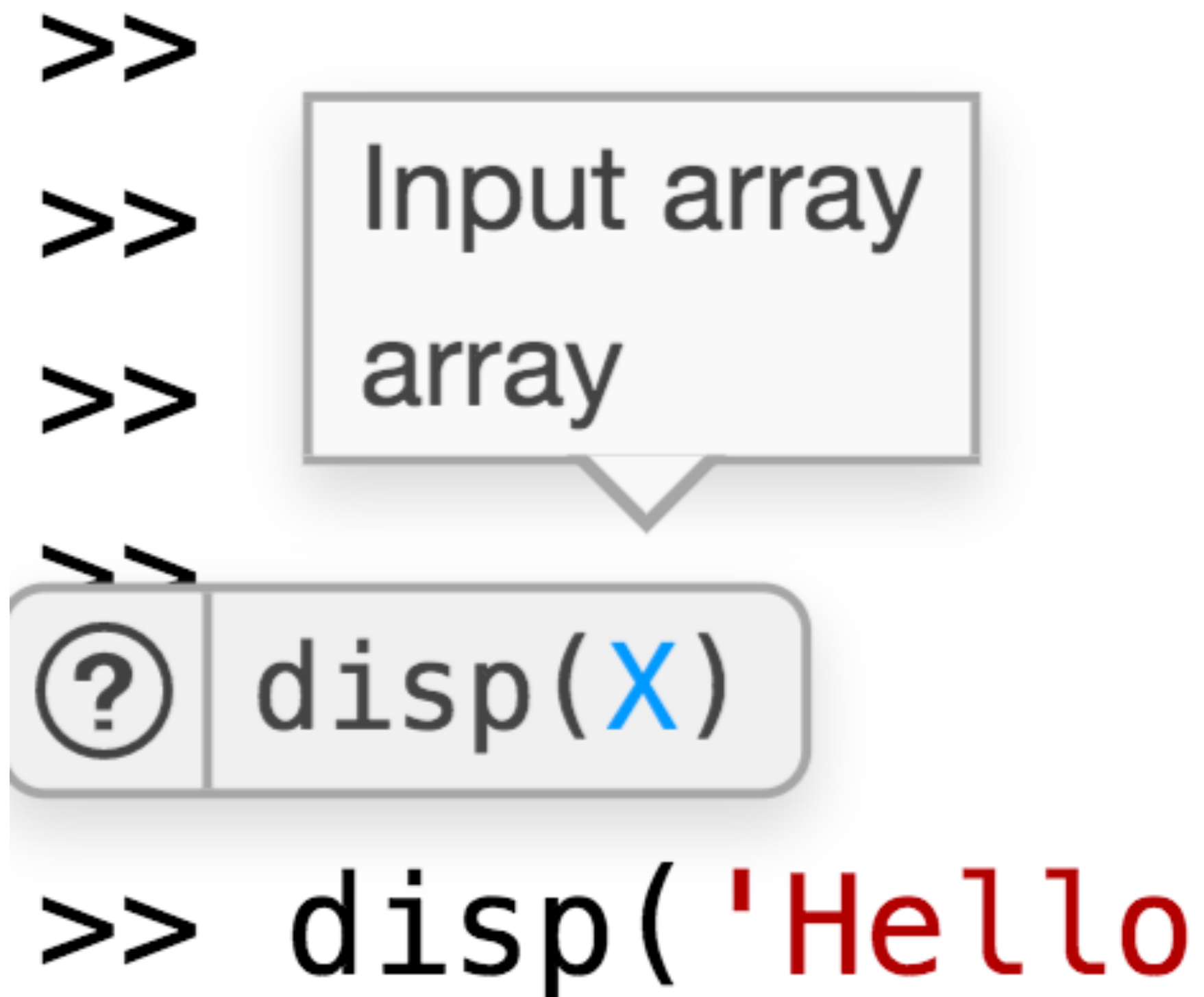
The image shows a MATLAB command window with a list of suggestions for the command 'disp'. Each suggestion is preceded by a small icon with the letters 'fx' inside a square. The suggestions are:

 disp	Display value of variable
 dissect	Nested dissection permutation
 discretize	Group data into bins or categories
 display	Show information about variable or result of ...
 disableDefaultIn...	Disable built-in axes interactions

Below the list, the command prompt shows the user has typed '>> disp' and the cursor is positioned at the end of the text.

# MATLAB helps you a lot

- Info on what arguments need to be passed



The image shows a MATLAB command window interface. At the top, there are three lines of the prompt '>>'. Below the third prompt, a tooltip box is displayed with the text 'Input array' and 'array'. Below the tooltip, there is a button with a question mark icon and the text 'disp(X)'. At the bottom, the command '>> disp('Hello' is entered, with 'Hello' in red text.

```
>>  
>>  
>>  
Input array  
array  
disp(X)  
>> disp('Hello
```

# MATLAB helps you a lot

- Built in documentation

```
>> help disp
```

```
disp Display array.
```

```
disp(X) displays array X without printing the array name or  
additional description information such as the size and class name.  
In all other ways it's the same as leaving the semicolon off an  
expression except that nothing is shown for empty arrays.
```

```
If X is a string or character array, the text is displayed.
```

```
See also fprintf, sprintf, int2str, num2str, rats, format, details.
```

```
Documentation for disp
```

```
Other functions named disp
```

# MATLAB helps you a lot

- Simple typo error catching

```
>> dispp('Hello World')
```

```
Unrecognized function or variable 'dispp'.
```

```
Did you mean:
```

```
>> disp('Hello World')
```

---

# MATLAB as a calculator

## Addition

```
>> 1+1
```

```
ans =
```

2

## Subtraction

```
>> 2-1
```

```
ans =
```

1

## Division

```
>> 1/2
```

```
ans =
```

0.5000

## Multiplication

```
>> 1*2
```

```
ans =
```

2

## Exponential

```
>> 2^2
```

```
ans =
```

4

# MATLAB as a calculator

- Follows order of operations

```
>> 2*3 + 1/2
```

```
ans =
```

```
6.5000
```

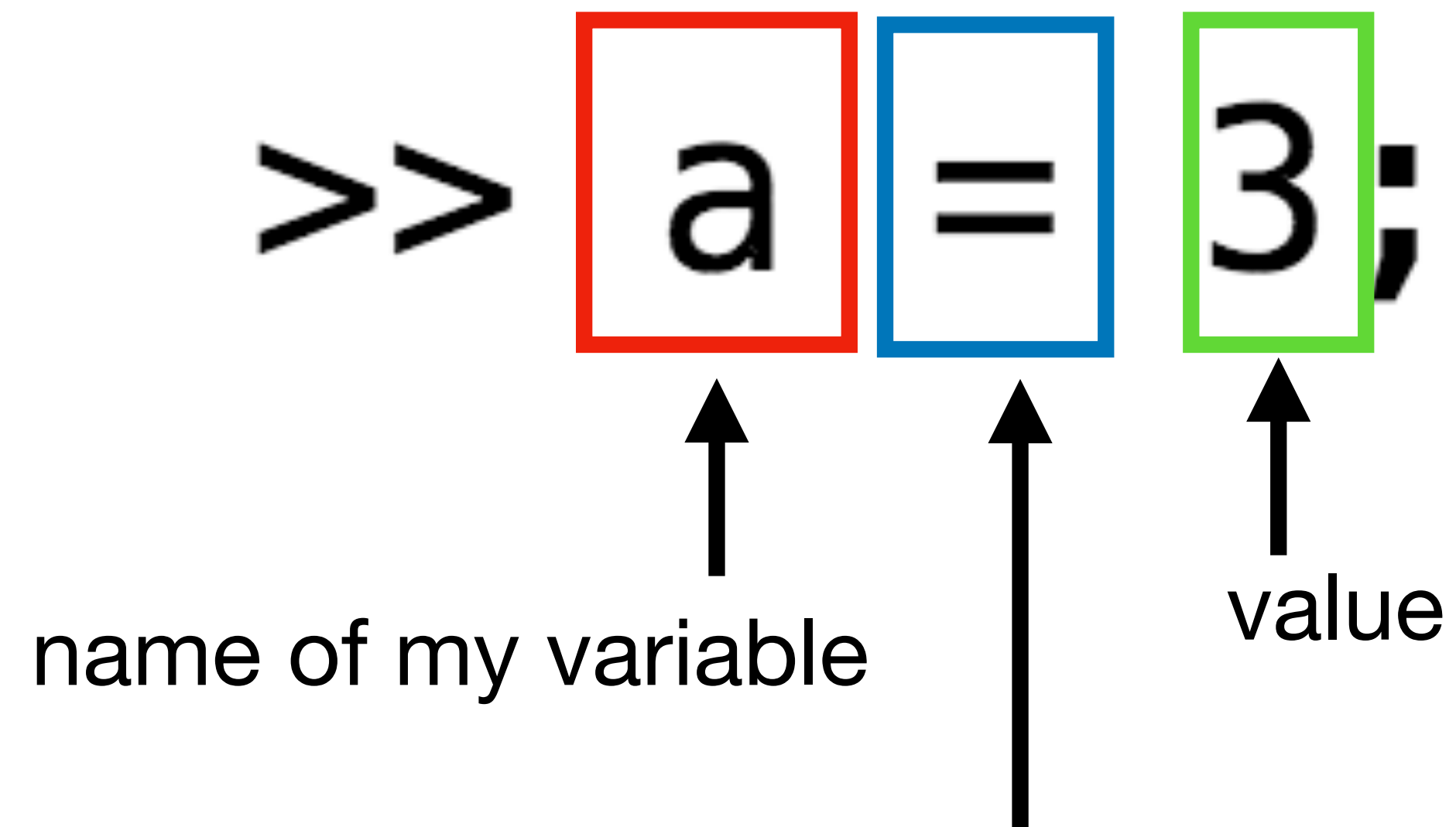
```
>> (2*3) + (1/2)
```

```
ans =
```

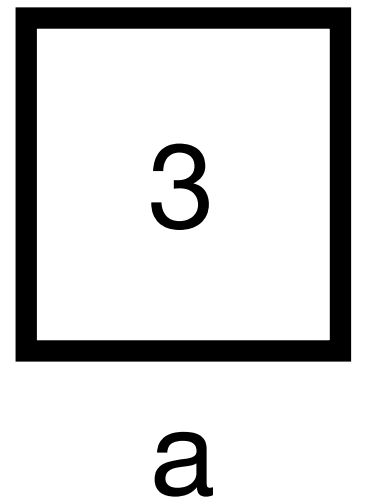
```
6.5000
```

# MATLAB variables

- Can use variables to hold values



assigns the value (3) to the variable (a)





# MATLAB variables

- Can use variables to hold values
- Use variables in calculations

```
>> a = 3;
```

```
>> a + 2
```

```
ans =
```

5

3

a

# MATLAB variables

- Can use variables to hold values
- Use variables in calculations
- Doesn't change value of a

```
>> a = 3;
```

```
>> a + 2
```

```
ans =
```

5

3

a

# MATLAB variables

- Can use variables to hold values
- Use variables in calculations
- Doesn't change value of a
- Need to reassign to a to change

```
>> a = 3;
```

```
>> a + 2
```

```
ans =
```

5
---

a

5

```
>> a = a+2;
```

# Putting it together: vars, math, & printing

- Write code to calculate the area of a rectangle with length 3 and width 5

```
>>  
>> length = 3;  
>> width = 5;  
>>  
>>  
>> area = 5*3;  
>>  
>>  
>> disp(area)  
15
```

# Putting it together: vars, math, & printing

- Now do this again for a rectangle with width 10 and height 2. Pretty annoying to type everything out again, right?

# Keeping track of our code

- Up arrow to see command history

```
help disp
— dispp('Hello World')
disp('Hello World')
a = 3
a = 3;
a + 2
2x clc
a = 3
>> a = 3
```

# Keeping track of our code

- Type few letters and then up arrow to see lines that start in same way

```
help disp
— dispp('Hello World')
disp('Hello World')
a = 3
a = 3;
a + 2
2x clc
a = 3
>> disp('Hello World')
```

# Keeping track of our code

- “whos” command to see list of variables you've defined

```
>> whos
```

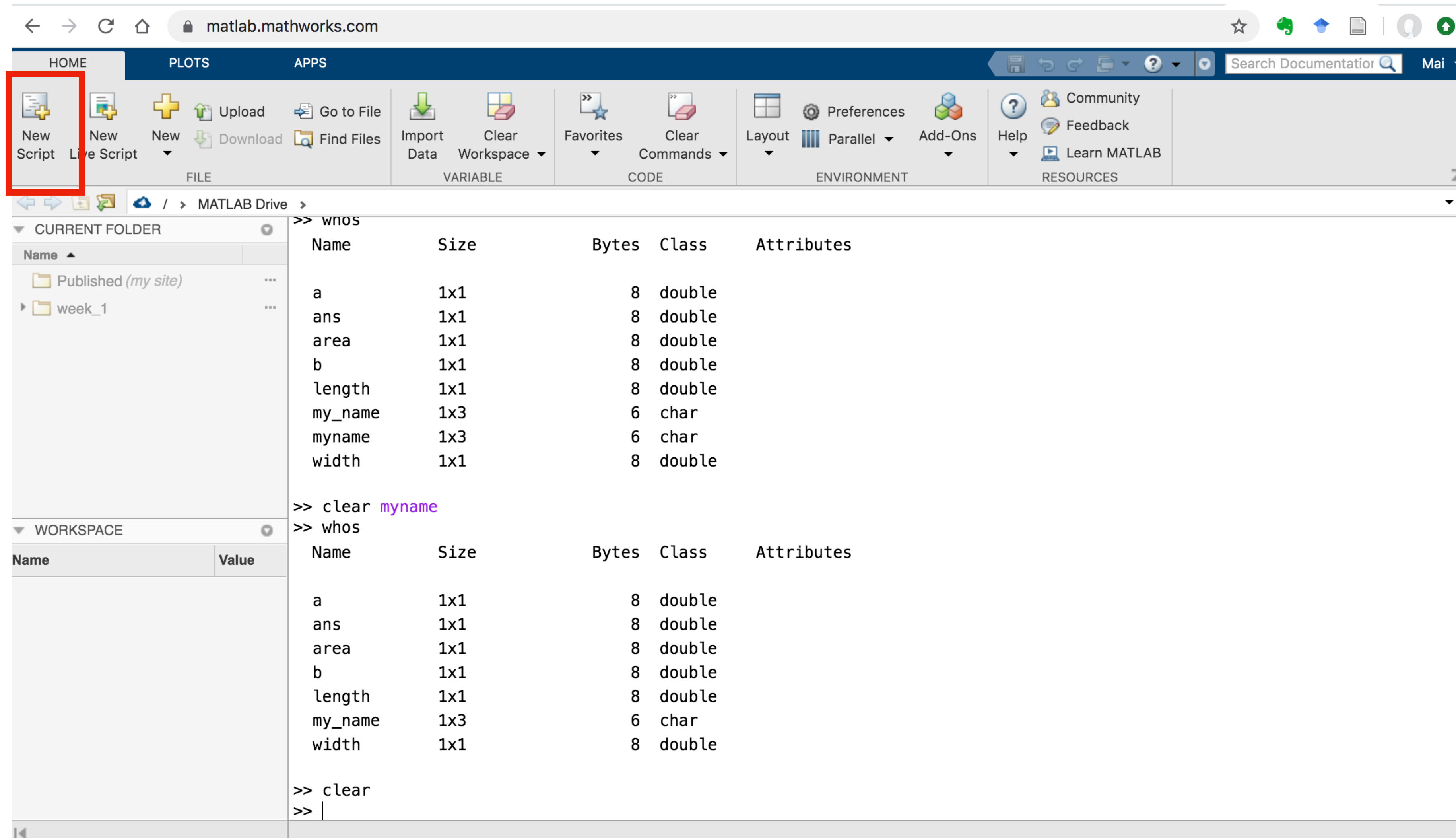
Name	Size	Bytes	Class	Attributes
a	1x1	8	double	
ans	1x1	8	double	
area	1x1	8	double	
b	1x1	8	double	
length	1x1	8	double	
my_name	1x3	6	char	
width	1x1	8	double	



# Keeping track of our code: scripts

- So far, writing all our code directly in command line
- Instead, write **scripts** to execute multiple lines of code
  - Text file with commands
  - MATLAB executes line-by-line

# Keeping track of our code: scripts

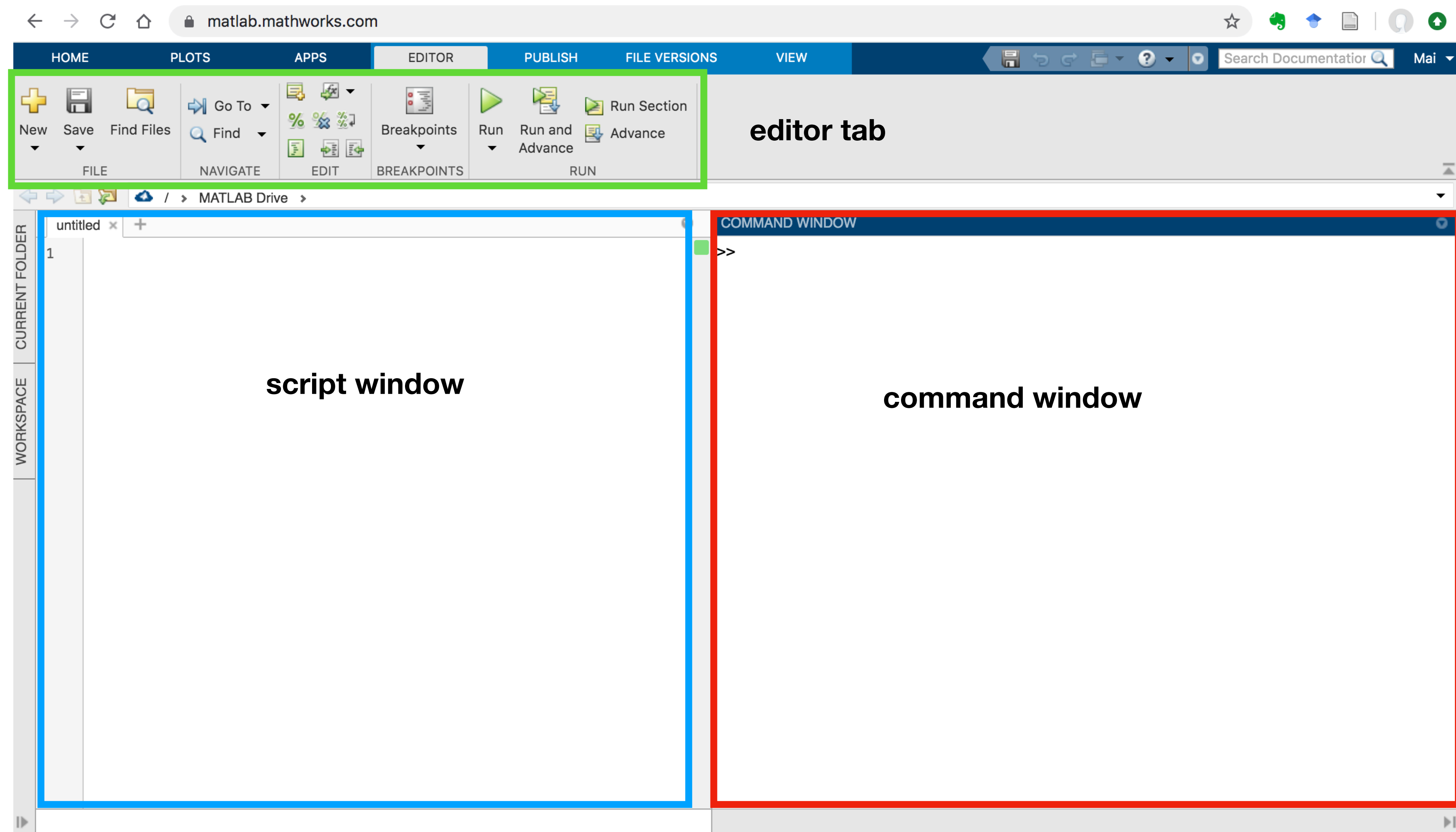


The screenshot displays the MATLAB web interface. The top navigation bar includes 'HOME', 'PLOTS', and 'APPS' tabs. A toolbar below the navigation bar contains icons for 'New Script', 'New Live Script', 'Upload', 'Download', 'Go to File', 'Find Files', 'Import Data', 'Clear Workspace', 'Favorites', 'Clear Commands', 'Layout', 'Parallel', 'Add-Ons', 'Help', 'Community', 'Feedback', and 'Learn MATLAB'. The 'New Script' button is highlighted with a red box. The main area is divided into three panes: 'CURRENT FOLDER' on the left, 'WORKSPACE' on the right, and a central command window. The 'CURRENT FOLDER' pane shows a tree view with 'Published (my site)' and 'week\_1' folders. The 'WORKSPACE' pane shows a table of variables in the workspace. The command window shows the output of the 'whos' command, followed by the 'clear myname' command, and then another 'whos' command showing the updated workspace.

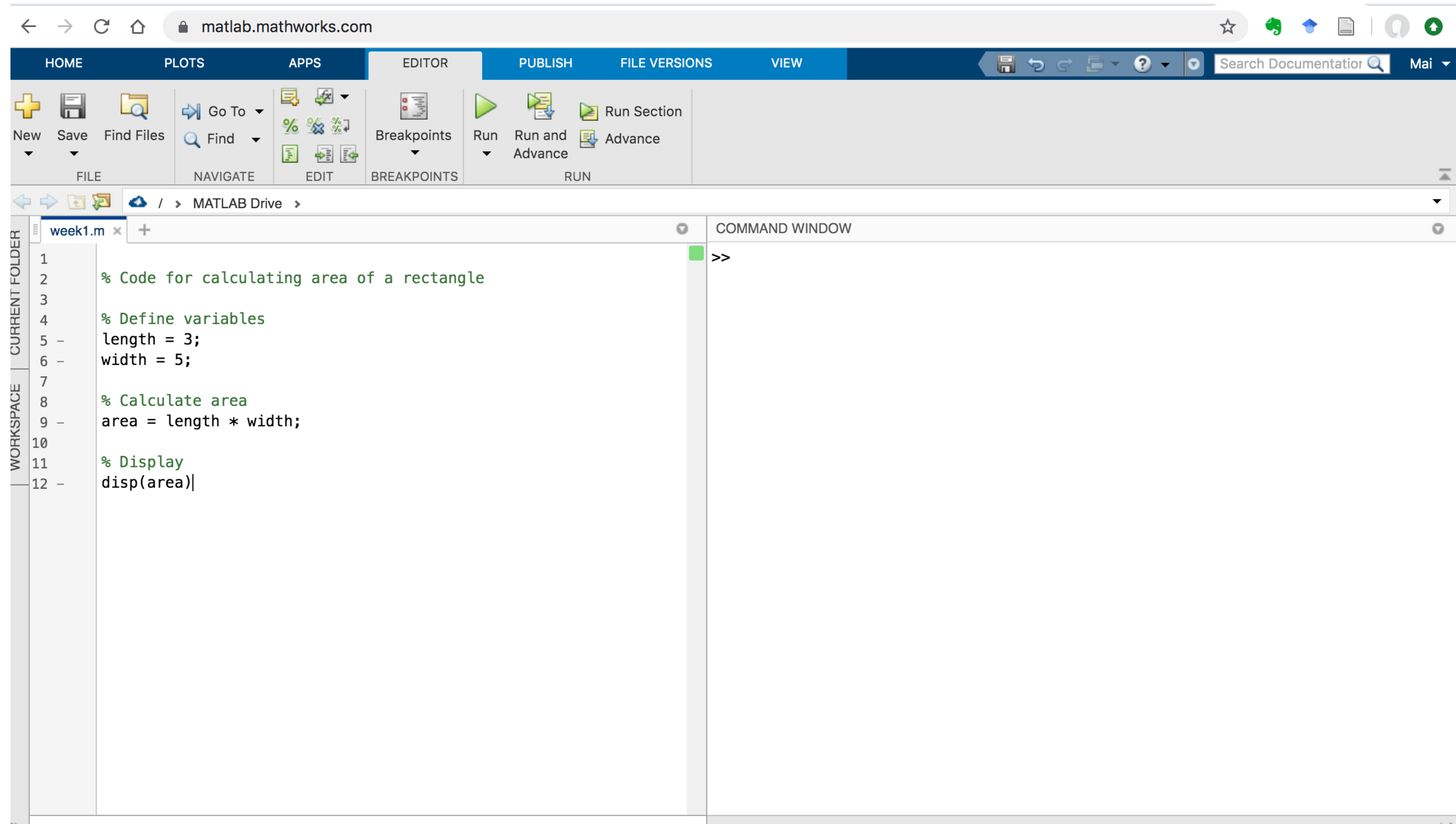
Name	Size	Bytes	Class	Attributes
a	1x1	8	double	
ans	1x1	8	double	
area	1x1	8	double	
b	1x1	8	double	
length	1x1	8	double	
my_name	1x3	6	char	
myname	1x3	6	char	
width	1x1	8	double	

```
>> whos
>> clear myname
>> whos
>> clear
>> |
```

# Rearrange your MATLAB window



# Copy your code for calculating area to the script



# Comments and code in a script

```
% Code for calculating area of a rectangle
```

```
% Define variables
```

```
length = 3;  
width = 5;
```

```
% Calculate area
```

```
area = length * width;
```

```
% Display
```

```
disp(area)
```

Lines starting with “%” are **comments**  
and won't be run by MATLAB

**VERY VERY VERY  
VERY VERY VERY  
IMPORTANT.  
SERIOUSLY.**

# Comments and code in a script

```
% Code for calculating area of a rectangle
```

```
% Define variables
```

```
length = 3;  
width = 5;
```

```
% Calculate area
```

```
area = length * width;
```

```
% Display
```

```
disp(area)
```

Lines starting with “%” are **comments** and won't be run by MATLAB

- Explains what the code does
- Makes it easier to read and understand (for you and others)
- Helps avoid mistakes
- **It is impossible to comment too much, and you will learn this to your own dismay**



# Comments and code in a script

```
% Code for calculating area of a rectangle
```

```
% Define variables
```

```
length = 3;  
width = 5;
```

```
% Calculate area
```

```
area = length * width;
```

```
% Display
```

```
disp(area)
```

Lines starting with “%” are **comments** and won't be run by MATLAB

Everything is code and will be run line-by-line by MATLAB

# Saving your script

←

→

↺

🏠

drive.matlab.com/files/

☆


🔗

📁

📄

👤

📶

 | MATLAB® Drive

Help

Mai ▾

Files

Files

Shared Content

Deleted Files

📁 Upload ▾

📁 New Folder

🔗 Share ▾

⬇️ Download

✏️ Rename

📁 Move to

📄 Copy to

🗑️ Delete

📁 MATLAB Drive

Name ▲	Size	Date Modified	Owned By
📁 Published		5/27/2020 02:53 PM	Me

126.7 KB / 5 GB used

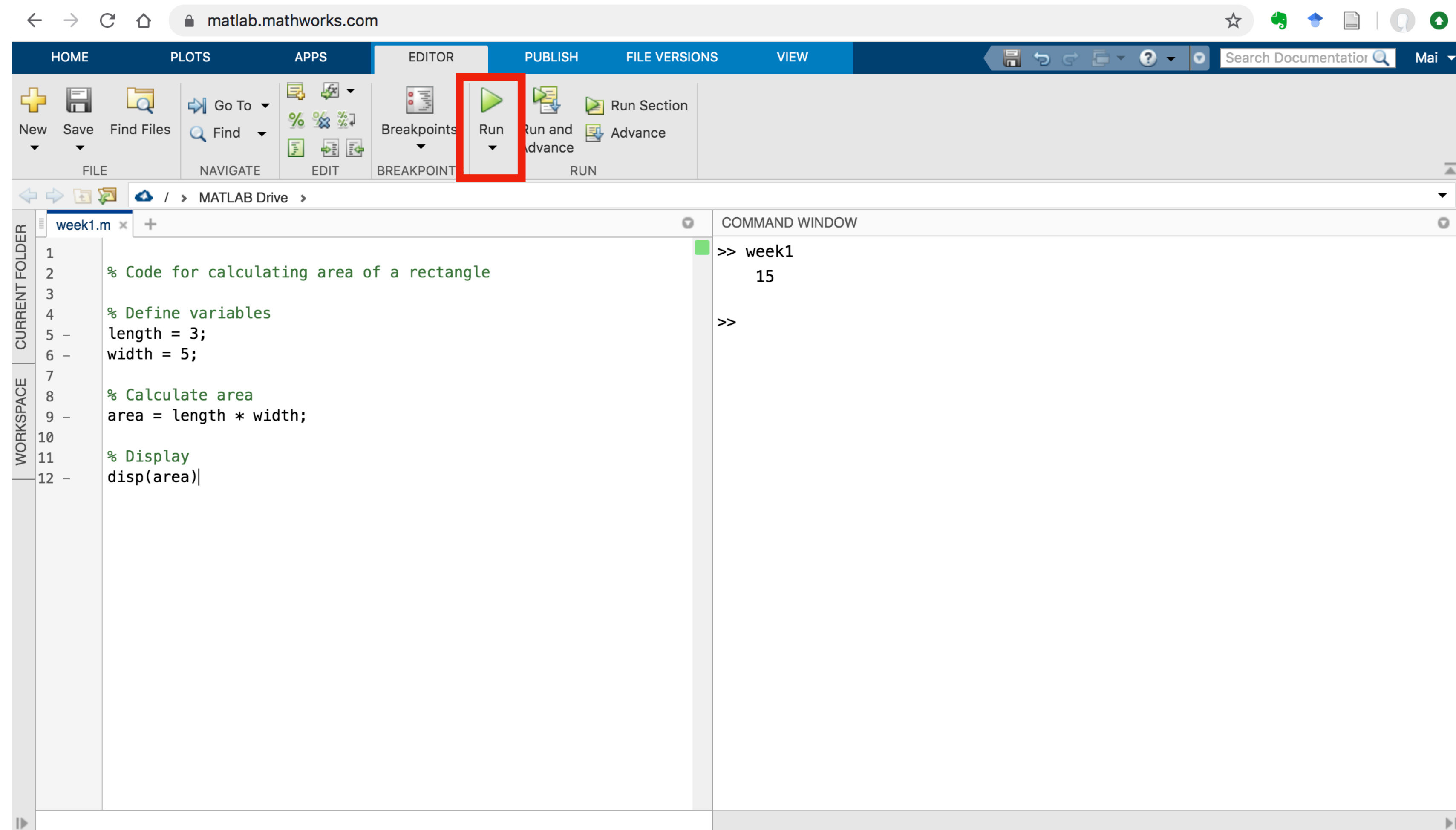
© 1994-2020 The MathWorks, Inc.

Patents | Trademarks | Privacy Policy | Preventing Piracy



# Running your script

- Option 1: Run the entire script using the run button



# Running your script

- Option 2: Run the entire script from the command line by entering its name:

COMMAND WINDOW

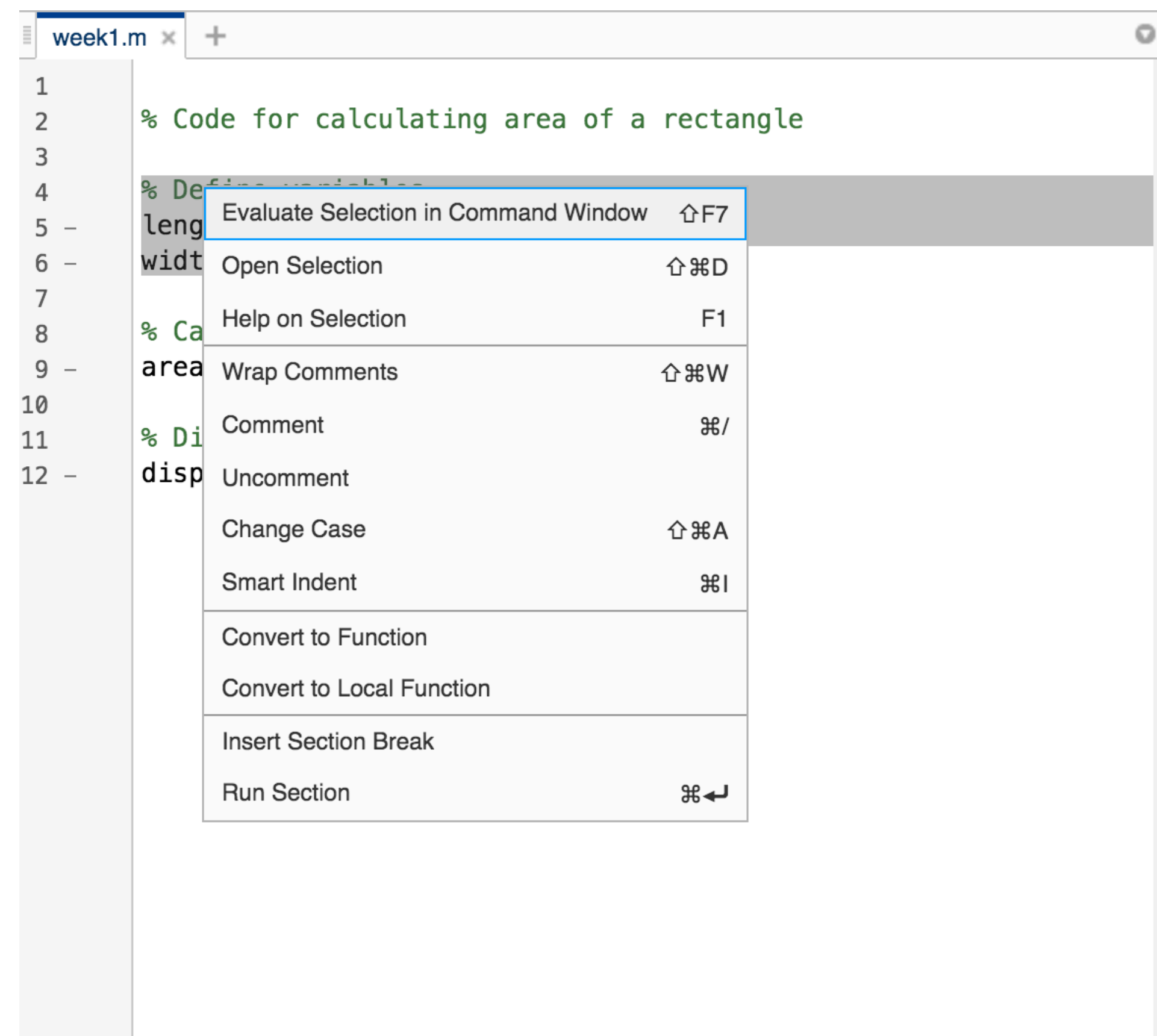
>> week1

15

>>

# Running your script

- Option 3: Run individual lines/sections by right clicking



# Exercise

- Write a new script to calculate the circumference and area of a circle with a radius of 3.
- Start by defining a variable, *radius*
- Using the built-in constant, *pi*, calculate the circumference of the circle and store a variable called *circumference*
- Calculate the area of the circle and store to a variable called *area*
- Display *circumference* and *area* using the disp function

# Review

- Running commands
- Assigning variables
- Using built-in functions
- Using MATLAB as a calculator
- Scripts and running scripts
- **EXCELLENT AND SEEMINGLY EXCESSIVE COMMENTING**

# Review

- **Built-in functions**

- disp(val): display a value in the command line
- whos: display variables in workspace
- clear: remove variables from workspace
- clc: clear command window
- help fxName: get documentation for fx

- **Assign a variable**

- syntax: var = value
- Ex 1: a = 3;
- Ex 2: my\_name = 'mai';

- **Mathematical operations**

- +, -, \*, /
- functions: log, cos, sin, tan, abs, max, min, mean, median, round
- constants: pi, i

- **Write a comment**

- Starts with %
- Isn't executed, helps explain what code is doing

- **Run a script**

- [Entire] Run button
- [Entire] Enter script name in command window
- [Line] Highlight and right click

-