Week 1: Introduction to MATLAB



Objectives:

- Get familiar with MATLAB's interface and basic commands.
- Learn how to think like a programmer (algorithmic thinking).
 - Start creating your own mini-programs to solve cool problems!

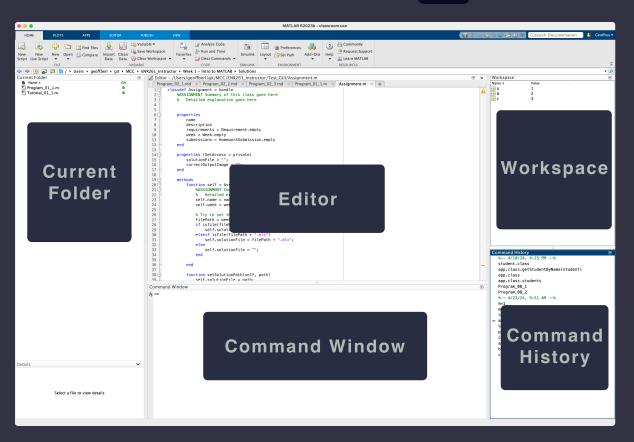
Introduction to the MATLAB Interface

What is MATLAB?

- MATLAB was originally developed for matrix calculations.
- It is useful to engineers and scientists for data analysis and solving complex problems.

The MATLAB Desktop Interface

- Command Window: Where you type commands and see immediate results.
- Workspace: Displays variables you've created during your session.
- **Command History**: Keeps a record of the commands you've entered.
- Editor: Used for writing scripts (M-files) and functions.
- Current Folder: The current working directory (cwd).



Interactive Demonstration:

- Open MATLAB and take a tour of the interface.
- Feel free to open MATLAB and follow along.

The Command Window

Used for Real-Time Interaction:

- MATLAB allows you to interact directly with your data. For example, typing 2 +
 - 3 and pressing enter will immediately return ans = 5.



clear: Clears all variables from the workspace.

clear: Clears all variables from the workspace.

whos (who): Displays detailed information about variables in the workspace.

clear: Clears all variables from the workspace.

whos (who): Displays detailed information about variables in the workspace.

VTip:

clc → Clear Command Window

clear → Clear Environment

Interactive Exercise:

- 1. Open MATLAB.
- 2. Try the following commands and observe:

```
2 + 3
clc
whos
```

Interactive Exercise:

- 1. Open MATLAB.
- 2. Try the following commands and observe:

```
2 + 3
clc
whos
```

3. What happened after each command?

Tip: More on Commands

- Use 1 to cycle through previously executed commands. (works better without the history window)
- Double-click or drag and drop from the command history window (may need to be enabled).
- Start typing a previously used command and use

 ↑ to filter
- Start typing a known command and use tab to see possible solutions
- ctrl+c: Stops the execution of a command or script if it runs too long or seems to be stuck.

- MATLAB uses standard mathematical operators: `+`, `-`, `*`, `/, `^`.
- Try to guess the output for the following:

Computation Notation:

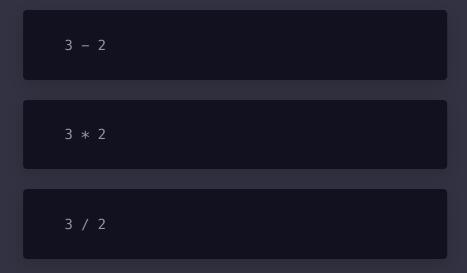
- MATLAB uses standard mathematical operators: `+`, `-`, `*`, `/`, `^`.
- Try to guess the output for the following:

3 - 2

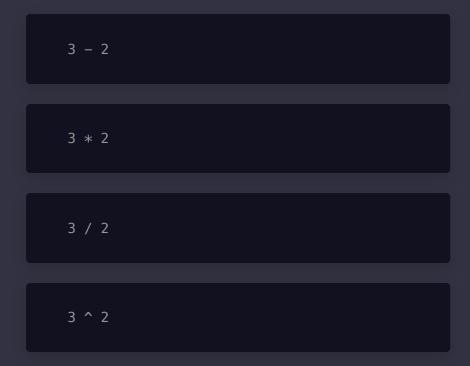
- MATLAB uses standard mathematical operators: `+`, `-`, `*`, `/`, `^`.
- Try to guess the output for the following:

```
3 + 2
```

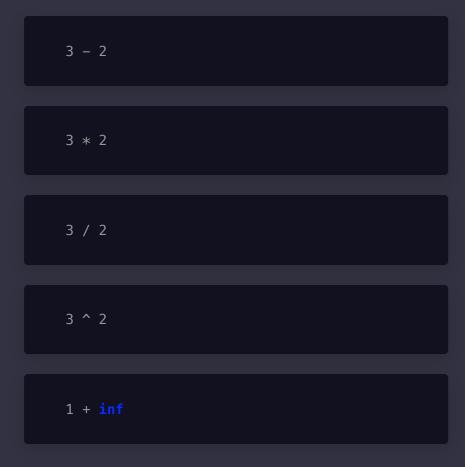
- MATLAB uses standard mathematical operators: `+`, `-`, `*`, `/`, `^`.
- Try to guess the output for the following:



- MATLAB uses standard mathematical operators: `+`, `-`, `*`, `/`, `^`.
- Try to guess the output for the following:



- MATLAB uses standard mathematical operators: `+`, `-`, `*`, `/`, `^`.
- Try to guess the output for the following:





Think of variables as boxes that hold things and can be labeled. Just like how a
box of cereal is labeled and has a specific cereal within it.



Think of variables as boxes that hold things and can be labeled. Just like how a
box of cereal is labeled and has a specific cereal within it.

Type:

- A box of cereal has a type, what's the type of stuff in this box? (cereal)
- Dynamic typing means you don't need to specify the type



Think of variables as boxes that hold things and can be labeled. Just like how a
box of cereal is labeled and has a specific cereal within it.

Type:

- A box of cereal has a type, what's the type of stuff in this box? (cereal)
- Dynamic typing means you don't need to specify the type

Value:

- A box of cereal has a value, it's written on the box? (Cheerios)
- You can change what's inside the box anytime you want!



Think of variables as boxes that hold things and can be labeled. Just like how a
box of cereal is labeled and has a specific cereal within it.

Type:

- A box of cereal has a type, what's the type of stuff in this box? (cereal)
- Dynamic typing means you don't need to specify the type

Value:

- A box of cereal has a value, it's written on the box? (Cheerios)
- You can change what's inside the box anytime you want!

"This box is a box of **cereal**, and it contains **Cheerios**"

Standard Naming Conventions

- camelCase
- snake_case
- ALL_CAPS`, `SCREAMING_SNAKE_CASE

Standard Naming Conventions

- camelCase
- snake_case
- ALL_CAPS`, `SCREAMING_SNAKE_CASE

Example:

```
MAX_FRUIT_ALLOWED = 15
numApples = 5
numOranges = 10
fruitCount = a * b
```

Important Concepts:

- Case Sensitivity: Variable and variable are different.
- Naming Conventions: Stick to clear, descriptive names to make your code easier to understand.

Reserved Words/Variables:

- MATLAB has predefined variables, known as constants, like pi and functions like inf.
- MATLAB also has predefined functions, such as sin, cos, clear, clc, sqrt, etc.

Warning! Avoid using reserved names for your variables when possible.

Interactive Exercise:

• Try the following:

```
radius = 7
height = 10
volume = (pi * radius^2 * height) / 3
```

Algorithm Development Process March 1988

- Analyze the problem Just like planning a trip, figure out where you want to go.
- 2. **Define inputs** Pack your bags! What do you need to go on your trip?
- 3. **Perform Manipulations** Map out your route, how are you going to get there?
- 4. **Produce Output(s)** Arrive at your destination!

Algorithm Development Process March 1988

- Analyze the problem Just like planning a trip, figure out where you want to go.
- 2. **Define inputs** Pack your bags! What do you need to go on your trip?
- 3. **Perform Manipulations** Map out your route, how are you going to get there?
- 4. **Produce Output(s)** Arrive at your destination!

Best Practices:

- **Commenting**: Use comments (%) to explain your code. This helps others (and future you) understand what your code does.
- Incremental Development: Write and test small pieces of code before integrating them into a larger program.

Applying What We've Learned:

1. Analyze the problem: Explain the formula $oldsymbol{v}$

$$v=rac{\pi r^2 h}{3}$$

2. **Define inputs** (in this case variables):

Applying What We've Learned:

1. **Analyze the problem**: Explain the formula $oldsymbol{v}$



2. **Define inputs** (in this case variables):

```
radius = 6
height = 12
```

Note: We don't need pi as that's provided for us

Applying What We've Learned:

1. **Analyze the problem**: Explain the formula $oldsymbol{v}$



2. **Define inputs** (in this case variables):

```
radius = 6
height = 12
```

Note: We don't need pi as that's provided for us

3. **Perform the manipulation**:

```
volume = (pi * radius^2 * height) / 3
```

Applying What We've Learned:

1. Analyze the problem: Explain the formula v



2. **Define inputs** (in this case variables):

```
radius = 6
height = 12
```

Note: We don't need pi as that's provided for us

3. **Perform the manipulation**:

```
volume = (pi * radius^2 * height) / 3
```

4. Produce Output/Result:

In this case, we are simply outputting to **volume** and the console (done by MATLAB)

Introduction to M-Files

What are M-Files?

 M-files are scripts or functions you can save and reuse. They help organize code and are essential for more complex projects.

Used for:

- Saving your programs (scripts).
- Executing a list of commands.
- Saving your work while you work through a problem.
- Re-opening and modifying your program at any time.

Creating a Basic M-File:

- 1. Open the MATLAB Editor.
- 2. Type the following:

```
% Compute and display the volume of a cone
radius = 6
height = 12
volume = (pi * radius^2 * height) / 3
```

Creating a Basic M-File:

- 1. Open the MATLAB Editor.
- 2. Type the following:

```
% Compute and display the volume of a cone
radius = 6
height = 12
volume = (pi * radius^2 * height) / 3
```

3. Save as volume_of_cone.m.

Creating a Basic M-File:

- 1. Open the MATLAB Editor.
- 2. Type the following:

```
% Compute and display the volume of a cone
radius = 6
height = 12
volume = (pi * radius^2 * height) / 3
```

- 3. Save as volume_of_cone.m.
- 4. Run the script by typing volume_of_cone in the Command Window.

More on Scripts:

- To run a MATLAB script call the filename without
- The script must be in the current working directory cwd
- Alternatively, you can provide a path (C:\my\script\location.m)

More on Scripts:

- To run a MATLAB script call the filename without
- The script must be in the current working directory cwd
- Alternatively, you can provide a path (C:\my\script\location.m)

Output Suppression:

- MATLAB, by default, prints command results to the command window.
- When someone asks for the volume of a cone, they probably don't want to see the radius, and height displayed.
- We can use the semicolon; operator to *suppress* output.

Updating our script

We'll add some semicolons to suppress the output of redundant values.

```
% Compute and display the volume of a cone
radius = 6;
height = 12;
volume = (pi * radius^2 * height) / 3;
```

Creating Quality Work:

- Write clear and understandable code.
- Add comments to document your code.
- Include a set of header comments with the program's objective, your name, anyone who assisted you.

Creating Quality Work:

- Write clear and understandable code.
- Add comments to document your code.
- Include a set of header comments with the program's objective, your name, anyone who assisted you.

TIP! Write your comments first, then write code to honor your comments.

Example Program with Expected format

```
% Compute and display the volume of a cone
%
% Filename: Volume_of_cone
% Developer: Your Name
% Assisted By:
% Date: 08/04/2024

% Always start with a clean output and workspace
clc
clear

% Declare variables
radius = 6; % Radius of the cone (inches)
height = 12; % Height of the cone (inches)
% Compute the volume of a cone
volume = (pi * radius^2 * height) / 3;
```

Error Handling and Debugging 😹

Be the Detective!

- Syntax Errors: Oops, you mistyped something! Let's fix it.
- Runtime Errors: Something went wrong during the run let's figure out why!

Interactive Exercise:

Run **clear** and introduce an error by changing **radius** to **radiuss** in the M-file and let's observe the error message.

Interactive Exercise:

Run clear and introduce an error by changing radius to radiuss in the M-file and let's observe the error message.

TIP! Start fresh to avoid confusion.

- Whenever you run into a problem like this it is a good idea to clear your workspace of any possible bad values.
- You should include clear, and sometimes clc in your script to ensure you are starting fresh.

Using the Help System

Accessing MATLAB Help:

- In Command Window: Type help enter to get information on a specific command.
- Right-click on a function and select "Help on functionName".
- Help Browser: Access via the ? icon.

Example:

• Type help sin to explore how MATLAB handles trigonometric functions.

General Shortcut Tips

Keyboard Navigation:

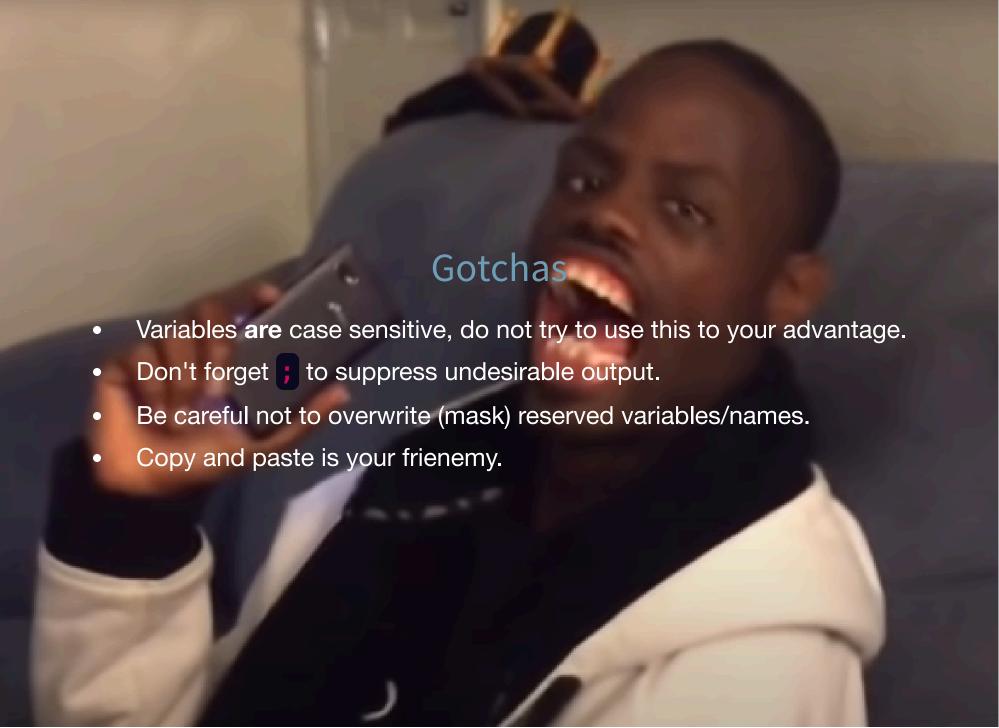
- home and end keys jump to the beginning and end of the line (cmd + ← / → on Mac).
- Arrow keys move a single character (← / →) or row (↑ / ↓)
- ctrl + ← / → jumps by word (option + ← / → on Mac).
- Holding shift while moving the cursor (using any of the aforementioned shortcuts), selects text.
- ctrl+c to copy, or ctrl+x to cut, ctrl+v to paste. (cmd+c)
 cmd+x, cmd+v on Mac)
- Double-click highlights a word, triple-click highlights a row.

Lecture Recap

- Variables are used for storing data for re-use.
- Use clear, and informative names.
 - Others and future-you will thank you
- M-files allow you to save and organize your work.
- The command window is immediate and can allow you to troubleshoot, or incrementally develop your code.
- Use Help whenever you're unsure of a command's syntax or usage.

Always:

- Write clear, well-documented code.
- Include comments. (This is your documentation)
- Suppress what doesn't need to be displayed.
- Start with a clear workspace and clcd command window.
- Learn keyboard shortcuts for efficient coding.
- Save often, commit often.



Analogies:

- MATLAB Interface: Think of MATLAB as a sophisticated calculator with a built-in notebook where you can save your calculations and notes.
- Command Window: Imagine MATLAB M-files (scripts) as a solution to a math problem. The Command Window is like your calculator where you check your values and help you develop your scripts.
- Variables: Think of variables as labeled containers. You store information in them,
 which you can use later.
- M-Files: M-files are like recipes; they list the steps needed to perform a task. You can save them, modify them, and share them with others.

Software Engineering Example