Topic: PYTHON PROGRAMMING

Presented by: RITAM MONDAL

python intern at Vault of Codes

(Assignment 1, task 2)

Date: 7th June 2025

```
$(function(){cards();});
$(window).on('resize', function(){cards();});
function cards(){
    var width = $(window).width();
    if(width < 750){
        cardssmallscreen();
    }else{
        cardsbigscreen();
    }
}
function cardssmallscreen(){
    var cards = $('.card').length;
    var height = 0;
    var card2 = 2;
```

Introduction to Python

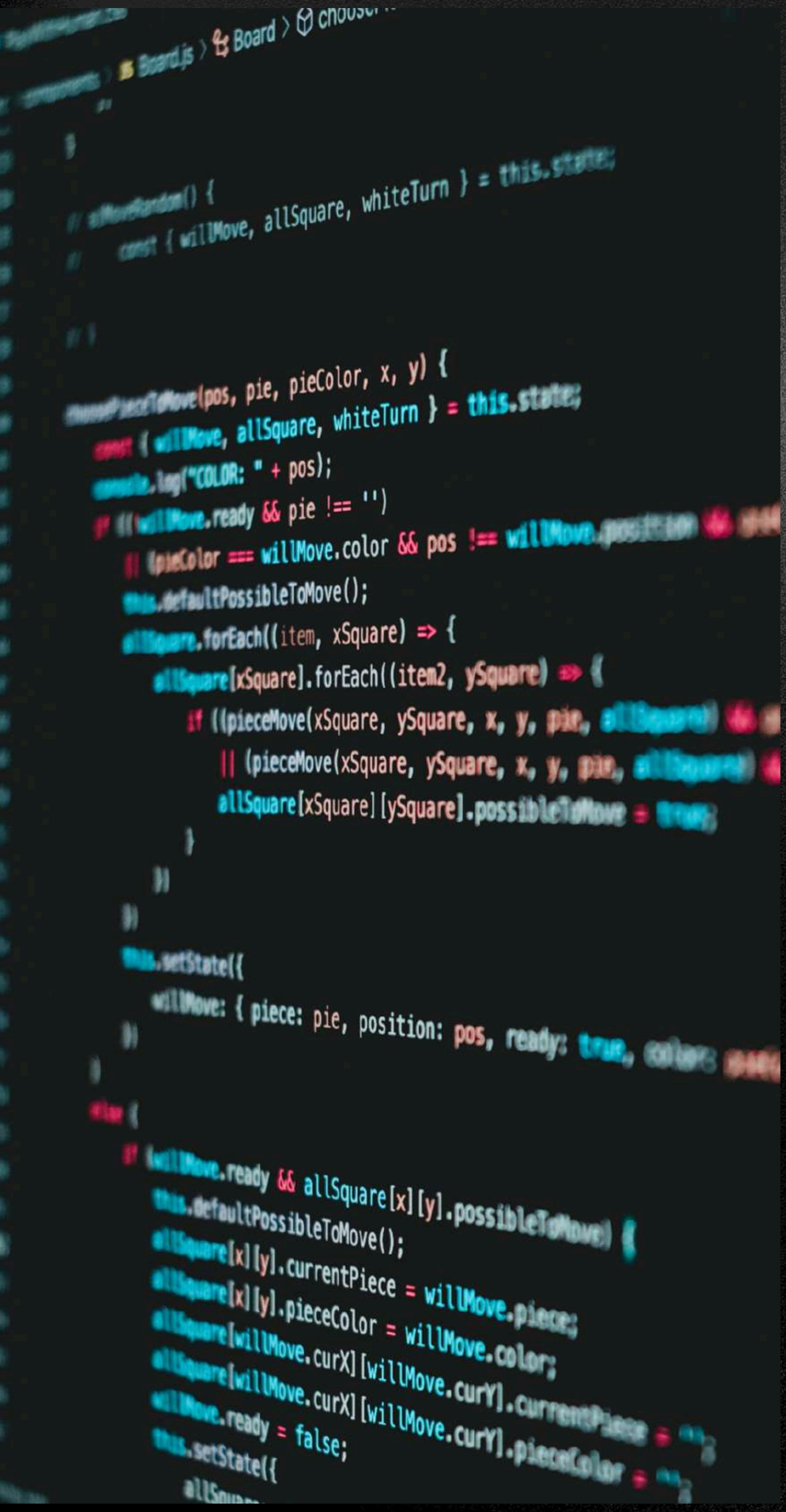
- Python is a high-level, interpreted programming language.
- Known for its readability and simplicity.
- Widely used in web development, data science, AI, machine learning, automation, etc.
- Supports object-oriented programming (OOP) principles.
- Open-source and community-supported.

History of Python

- Python was created by Guido van Rossum in the late 1980s.
- Officially released in 1991.
- Named after the British comedy group "Monty Python".
- Major versions:
    1. Python 2.0 (2000): in 2000 introduced major feature
    2. Python 3.0 (2008): (Py3k) released in 2008 with modern syntax
- Constantly evolving with regular updates.

Key Features of Python

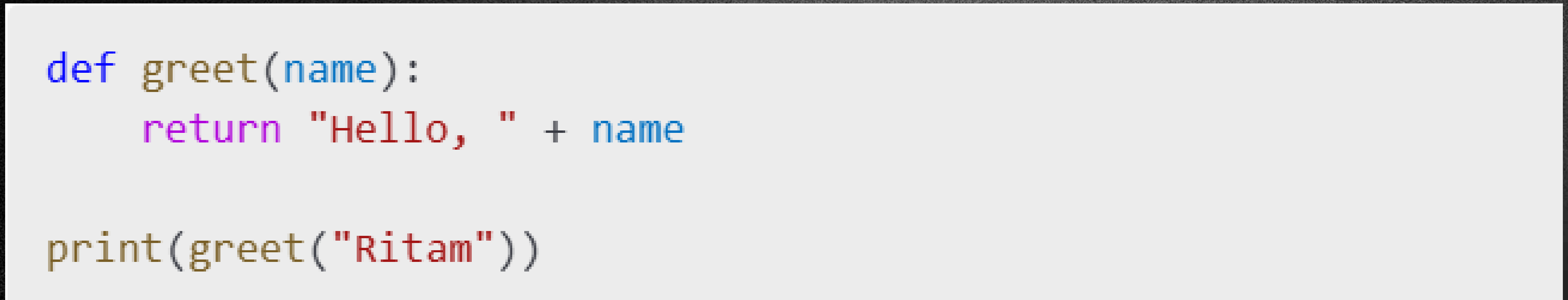
- Easy-to-learn syntax
- Interpreted language (no compilation needed)
- Dynamically typed
- Large standard library
- Extensible with other languages like C/C++



Functions in Python

- Definition: Created with def keyword followed by name and parameters.
- Parameters: Placeholders for values passed to functions.
- Arguments: Actual values passed to functions.
- A block of reusable code that performs a specific task.
- Helps in breaking code into smaller, manageable parts.

Example



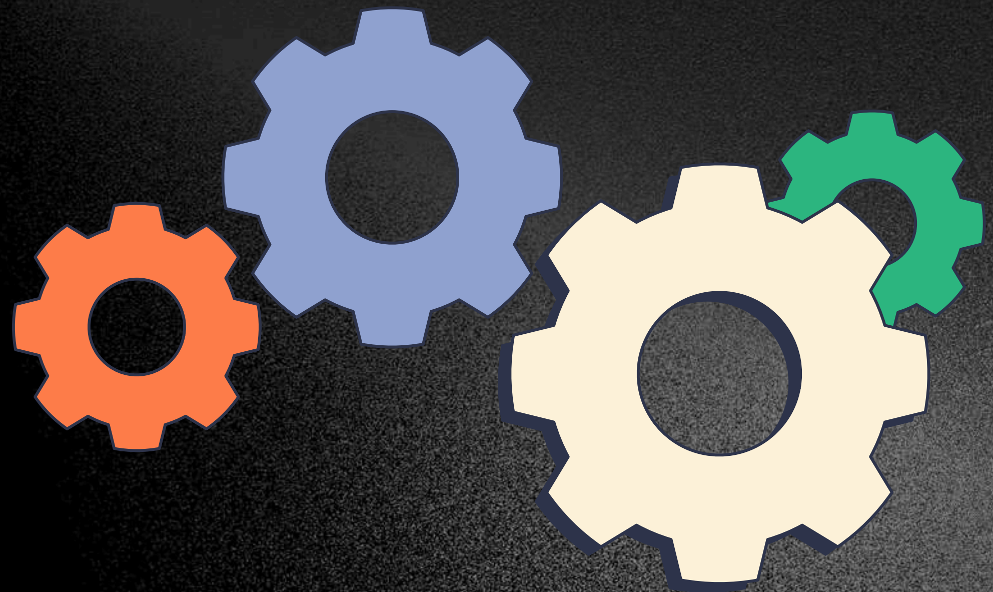
```python
def greet(name):
    return "Hello, " + name

print(greet("Ritam"))
```

Output: “Hello, Ritam”

Types of Functions

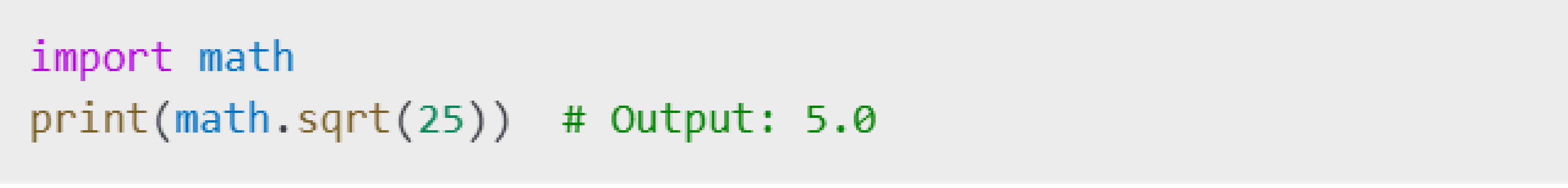
- Built-in Functions: print(), len(), range() etc.
- User-defined Functions: Created by the user using def keyword.
- Lambda Functions: Anonymous, one-liner function
- Function Arguments: Positional, keyword, default, variable-length.
- Function Overloading: Achieved through default and variable-length arguments.



Modules in Python

- A module is a file containing Python code (functions, variables, classes).
- Purpose: Organize code into reusable and manageable units
- Promotes code reuse and modular programming.
- Third-Party Modules: External modules available via package managers (pip).
- Can import built-in or custom modules.
- Creating Modules: Write your own modules for custom functionality.
- Usage: Access module contents using dot notation (module.function()).

Importing Modules

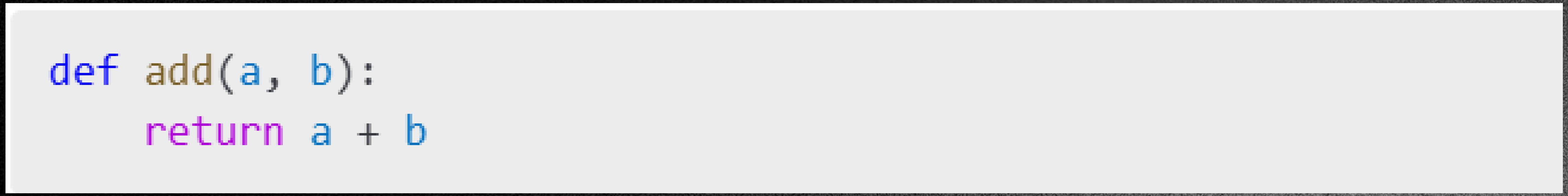


```
import math
print(math.sqrt(25))  # Output: 5.0
```

- Use import keyword to access functions/objects from another file/module.
- Can also use from module import function.

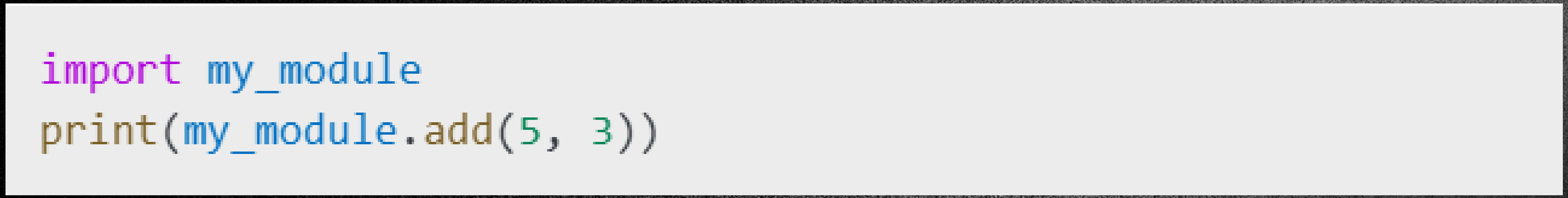
Creating Your Own Module

my_module.py



```
def add(a, b):
    return a + b
```

main.py



```
import my_module
print(my_module.add(5, 3))
```

Conclusion

- Python is a beginner-friendly language with powerful features.
- Functions help break down code and reuse logic.
- Modules organize and structure large codebases.
- Mastering these concepts forms the foundation of Python programming.

Reference

Python Programming

history of python

Geeks of geeks, history of python

Thank you