

# Python Immersion, Day 1

A Project-Based Approach to Learning the World's Most Popular  
Programming Language



**Shaun Wassell**

Software Developer, Educator

# What We'll Be Covering Today

- **Basics of Writing and Running Python Code**
- **Python Control Structures**
- Python Data Structures
- Working with Files
- Networking Basics
- Generative AI for Python

# Q & A

Any questions? Put them in the Q&A widget

## Section 1: Python Basics



- Writing and Running Python Code
- Getting User Input
- Basic Data Types and Operations

# Writing and Running Python - It's Easy!

1. Create a ".py" file
  2. Add Python code
  3. Run the file
- If you want to follow along, go to <https://replit.com/new/python3>
  - You can also download Python locally at <https://www.python.org/downloads/>

# Getting User Input

- The "input" function:

```
name = input("What's your name?")
```

- By default, this value will be a "string", even if you enter a number

# Python's Data Types

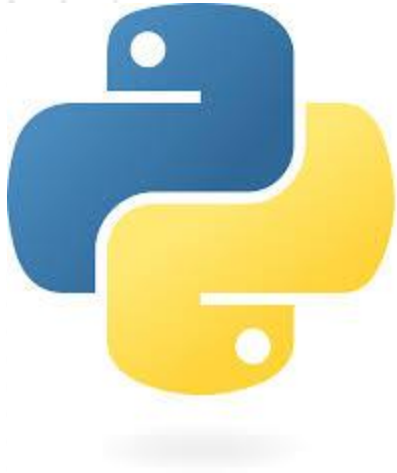
- Data types represent different kinds of data
- Python has several standard data types
- The fundamental data types include:
  - Strings - basically just text, i.e. "Hello"
  - Integers - counting numbers, i.e. 1, 2, 3
  - Floats - numbers with a decimal portion, i.e. 3.14159
  - Booleans - yes/no values, i.e. True or False

# Q & A

Any questions? Put them in the Q&A widget



## Section 2: Python Control Structures



- If-Statements
- Loops
- Functions
- Exception Handling

# Python's Control Structures

- If-Statements - let us "skip" certain parts of our programs
- Loops - let us repeat code an arbitrary number of times
- Functions - let us give a name to certain pieces of code
- Exceptions - let us name and anticipate errors

# Python If-Statements

- Let us "skip" certain parts of our programs
- Based on boolean values
- Provide "else" and "elif" statements that let us further refine behavior
- Boolean values can be combined and modified with "and", "or", "not"
- Other types of values can sometimes be used as the boolean expression

# Python Loops

- Let us repeat code an arbitrary number of times
- Can be used with "range" to repeat code a specific number of times
- Can be used with "collection types" to run code once for each "thing" in a collection
- Can be used with a boolean expression to repeat code until some condition is no longer True

# Python Functions

- Let us give a name to certain pieces of code
- Great for increasing code reusability
- Arguments can be used to leave "blank spaces" that can be filled with different values
- Can "return" values that can be used outside the function

# Python Exceptions

- Let us name and anticipate errors
- There are many errors in Python programs that it's not possible to prevent completely
- Examples include: user input, file not found, failed network requests, etc.
- Giving errors a name makes them easier to handle "gracefully" using "try" and "except" blocks

# Take-Home Project #1 - Build A Number-Guessing Game



- Randomly generate a number between 1 and 10
- The user must try to guess the number
- The program must check to see if the user guessed correctly
- If not, the program should loop until the user guesses correctly
- Extra credit: Implement a "maximum" number of guesses
- Extra credit #2: Add exception handling for user input

# Q & A

Any questions? Put them in the Q&A widget