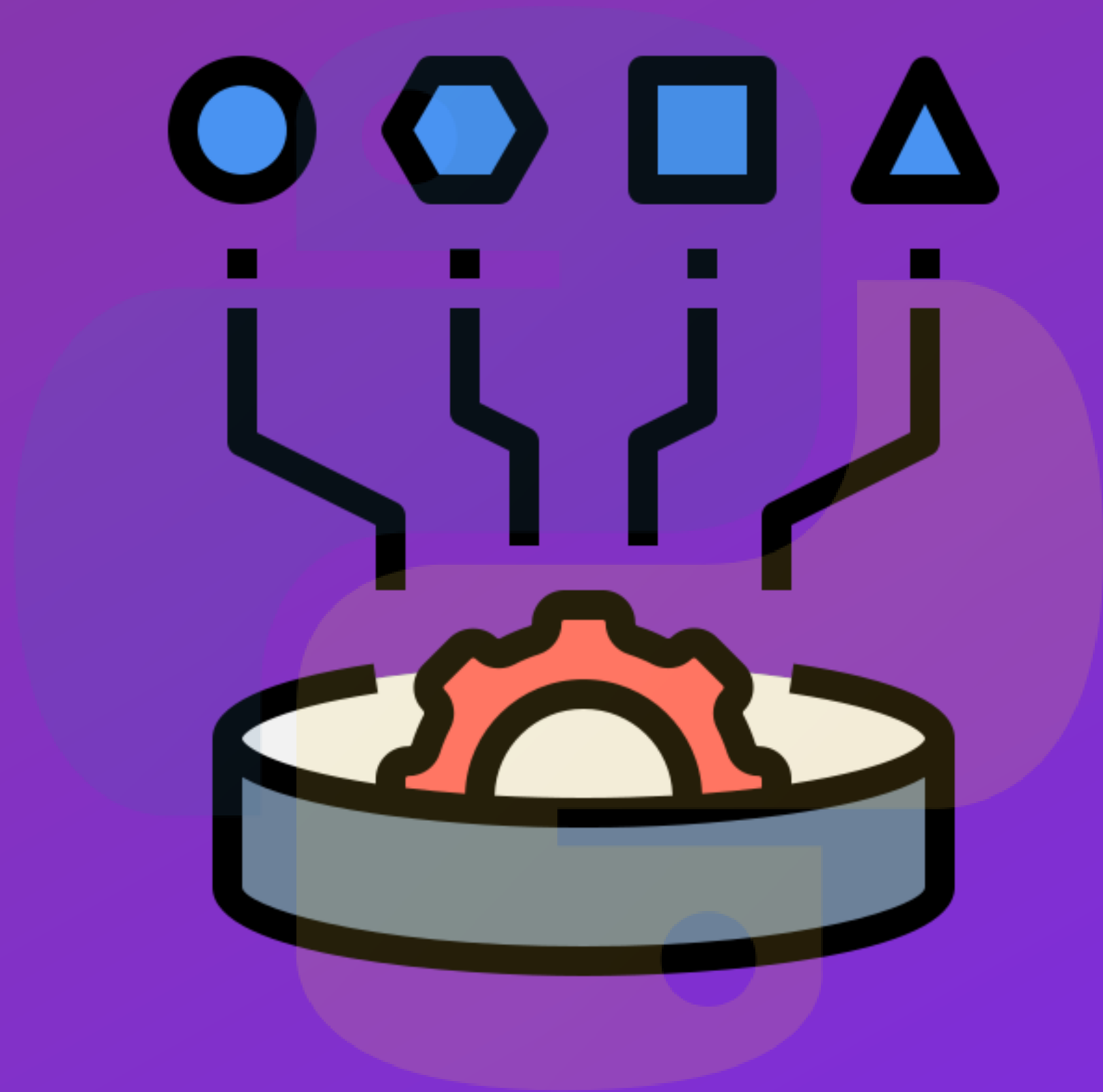


PYTHON BASICS



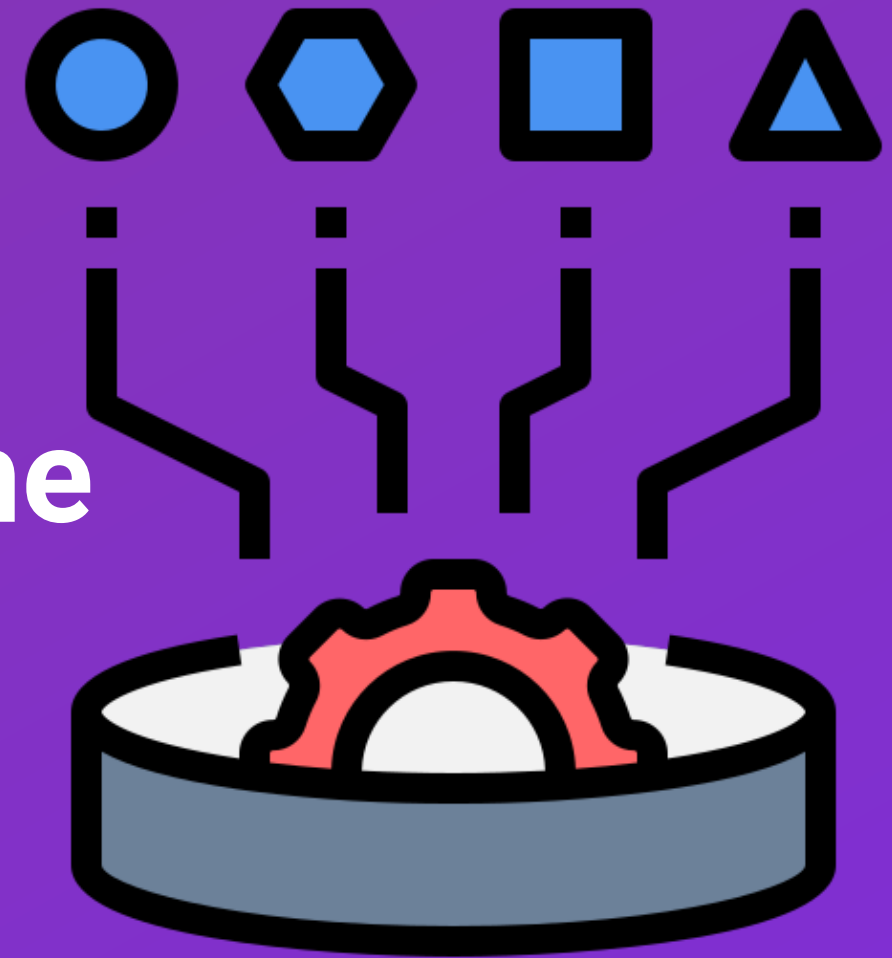
PYTHON VARIABLES



PYTHON VARIABLES

- We use variables to store the data the application is working with;

- A variable is **label** or a **name** for a memory location where a value, which can be manipulated, is stored;



NAMING CONVENTIONS



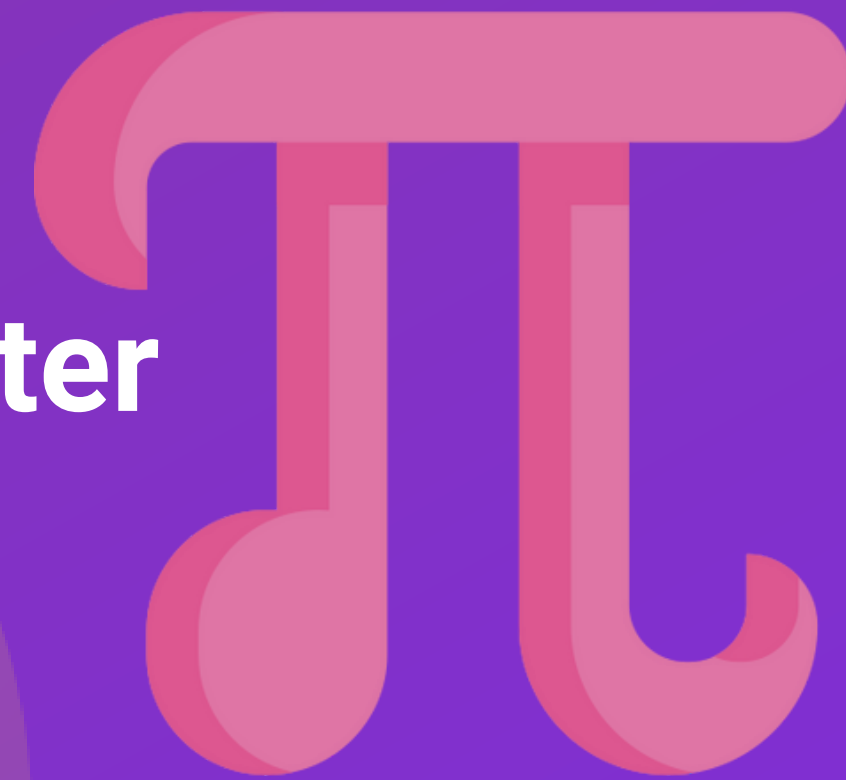
- Variable names can contain letters (a-zA-Z), digits (0-9) and underscore;
- A variable name can't start with a number;
- Don't use special characters in variable names;
- Variable names can't be reserved words;
- Use snake_case for variable names and PascalCase for class names;
- Don't use words that have special meanings or are already defined;

CONSTANTS IN PYTHON



CONSTANTS IN PYTHON

- Values that will never change (pi, water boiling temperature, etc);
- Python does not have anything in the language to declare a constant. It's declared like a variable;
- "Constant" values are often written with UPPER-CASE letters;



STATIC VS. DYNAMIC TYPING



A language is statically typed if the type of a variable is known before running the program. Examples: C, C++, Java, Go;

Python is dynamically typed. The type is associated with run-time values and not with named variables;

In Python the values stored in variables have types and not the name or the variable itself;

STATIC VS. DYNAMIC TYPING

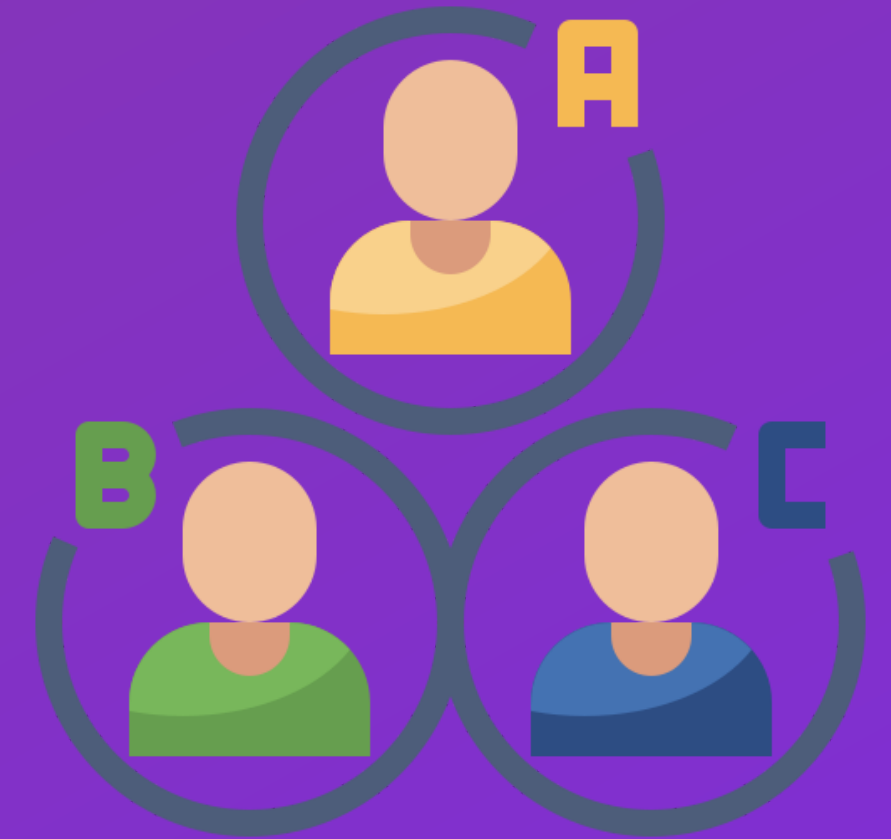
Statically-typed languages have better performance at run-time and are faster.

Dynamically-typed languages are comparatively faster during development time and are more flexible.

Python is a dynamically-typed language.



BUILT-IN DATA TYPES



- Numbers: `int`, `float`, `complex`;
- Booleans: `True`, `False`;
- Strings. `language = "Python rulz!"`
- Lists and Tuples. `cities = ["NYC", "London", "Berlin"]`
- Sets. `states = {"CA", "NY", "DC"}`
- Dictionaries. `friend = {"name": "Dan", "age": 30, "location": "Bucharest"}`

NUMBERS IN PYTHON

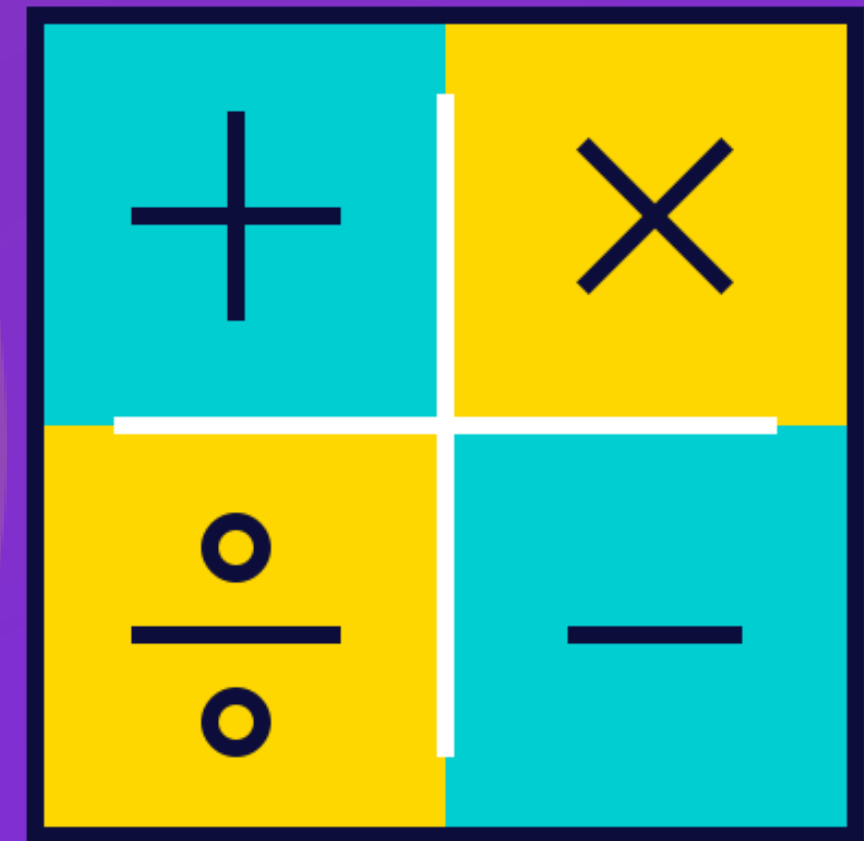
- Integers
- Floats
- Complex numbers



PYTHON OPERATORS

An operator is a symbol of the programming language able to operate on values.

- Arithmetic Operators: + - * / // ** %
- Assignment Operators: = += -= *= /=
- Comparison Operators: == != > >= < <=
- Identity Operators: "is" "is not"
- Logical Operators: "and" "or" "not"



ORDER OF OPERATIONS (OPERATOR PRECEDENCE)

- 1 Exponentiation (******)
- 2 Multiplication (*****) and division (**/**)
- 3 Addition (**+**) and subtraction (**-**)



ASSIGNMENT OPERATORS

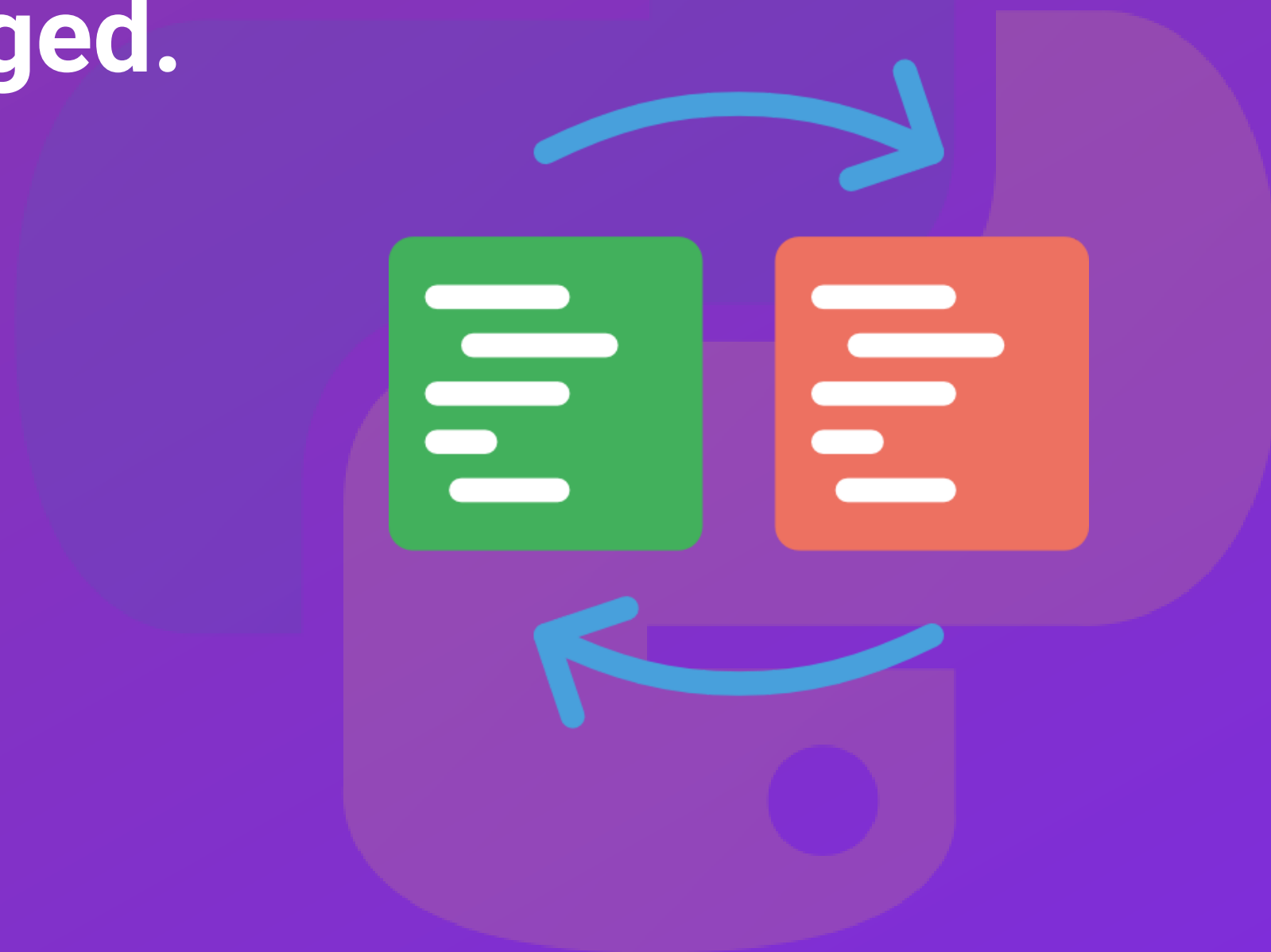
- Equals (=)
- Plus equals (+=)
- Minus equals (-=)
- Star equals (*=)
- Slash equals (/=)
- Double stars equals (**=)
- Percent equals (%=)

MUTABILITY vs. IMMUTABILITY



MUTABILITY vs. IMMUTABILITY

The value of a mutable variable can be changed after it has been created, but the value of an immutable variable can't be changed.



FLOATING POINT ARITHMETIC

Numbers in a computer are represented using bits (base 2), not decimal digits (base 10).



FLOATING POINT ARITHMETIC

Many decimal fractions cannot be represented exactly in binary and are only approximated by the binary floating-point numbers stored in the machine.



FLOATING POINT ARITHMETIC

Some rational numbers cannot be represented using a finite number of digits.

