
# PROGRAMMING PYTHON

Lecture # 2 - Basics of Python

# Data Types in Python


## Data type

Kind of values, a variable can store

### int

Integer data type represents whole numbers, such as -2, -1, 0, 1, 2, etc.

### float

Float data type represents floating-point numbers with decimal places, such as 3.14, -0.5, $6.67 \times 10^{-11}$ etc.

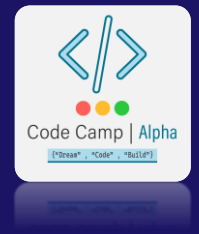
### str

String data type represents text, enclosed in single or double quotes, like "Hello, World!" or 'Python'.

### Bool

Boolean data type represents either True or False.

# Data Types in Python




## Data type

Kind of values, a variable can store

### list

List is an ordered collection of items, which can be of any data type. Lists are defined using square brackets, like [1, 2, 3].

### tuple

Tuple is an ordered, immutable collection of items. They are defined using parentheses, like (1, 2, 3).

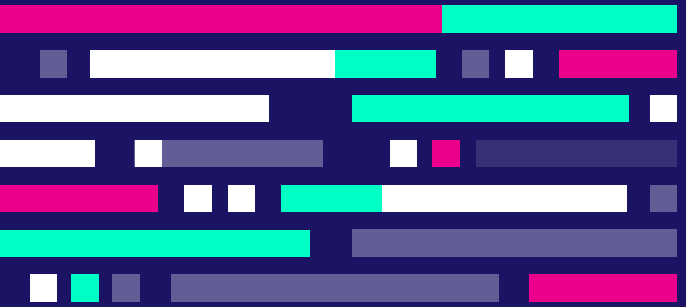
### Dict (dictionary)

Dictionary is an unordered collection of key-value pairs. They are defined using curly braces, like {'name': 'John', 'age': 30}.

### set

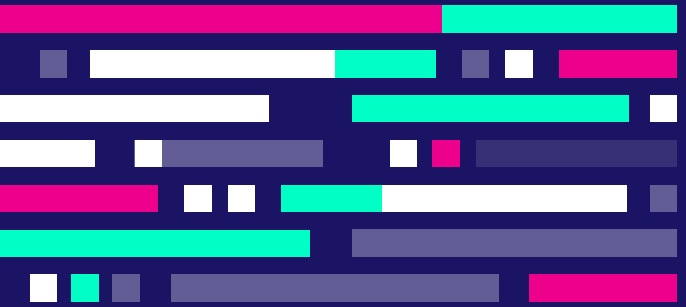
Set is an unordered collection of unique elements. They are defined using curly braces, like {1, 2, 3}.

Code Camp | Alpha



# Variables

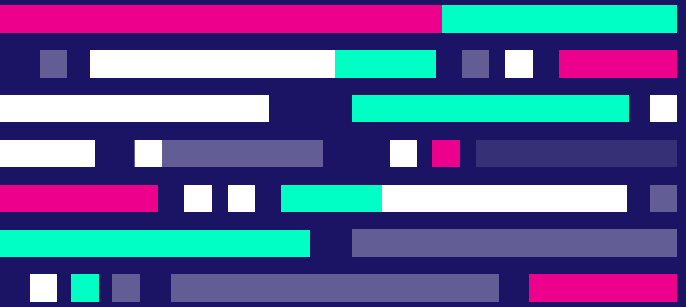
- Variables are containers for storing data values.
- Python has no command for declaring a variable.
- A variable is created the moment you first assign a value to it.
- Example:
  - x = 20
  - name = “Muhammad”
  - condition = true



# Naming Rules

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _ )
- Variable names are case-sensitive (age, Age and AGE are three different variables)
- A variable name cannot be any of the Python keywords.

Code Camp | Alpha



# More …

- You can change variable values
  - Name = “Wasiq”
  - Name = “Khan”
- Types are automatically assigned but you can type-cast as well
  - x = str(3)  # 3 is string
  - y = int(3)  # y is integer
  - z = float(3)  # z is 3.0
- You can always get the type of variables by type command.
  - pi = 3.14
  - Print(type(x))

# Right and Wrong?

2name

my.var

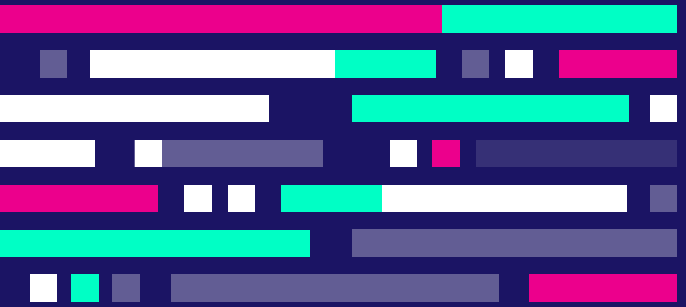
my variable

Name2

name_2

first_name

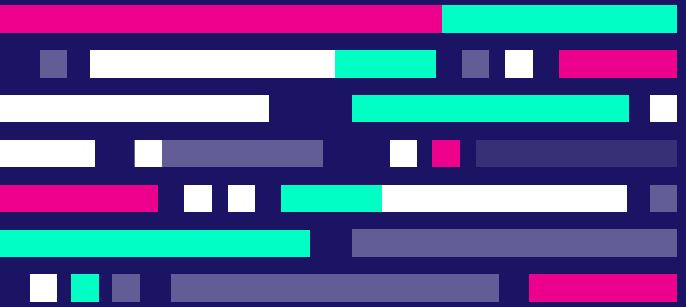
Code Camp | Alpha



# Input & Output

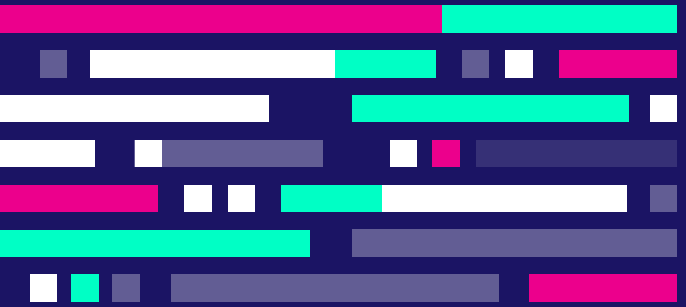
- Values of variables can be stored via user input.
  - name = input(“Enter your name”)
- The output is produced via print command
  - print(name)
- You can also use use “+” or “,” to combine outputs
  - f_name = “Wasiq”
  - l_name = “Khan”
  - print(f_name + l_name)
  - print(“Name is: “ + f_name + “ “ + l_name

Code Camp | Alpha



# Input & Output

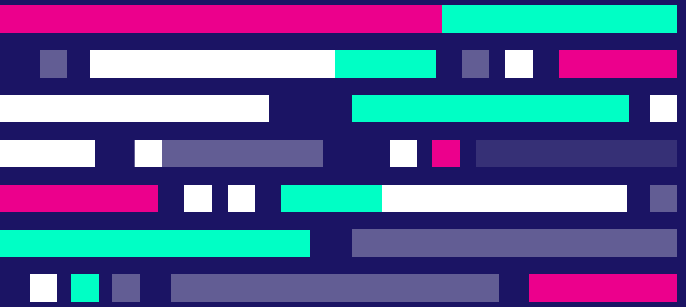
- “+” is only used with similar data types
- “,” comma can be used with different data types as well
- “+” is addition for int and float but **concatenation** in strings
  - Concatenation is like combining strings
- Write a python program that takes two inputs from user, name and age, and print them in one line using concatenation.



# Numbers

- Types
  - Integers
    - x = 12
  - Floats
    - const_pi = 3.14
  - Complex Numbers
    - c = 4+2j
      - “j” represents complex part

Code Camp | Alpha



# Booleans

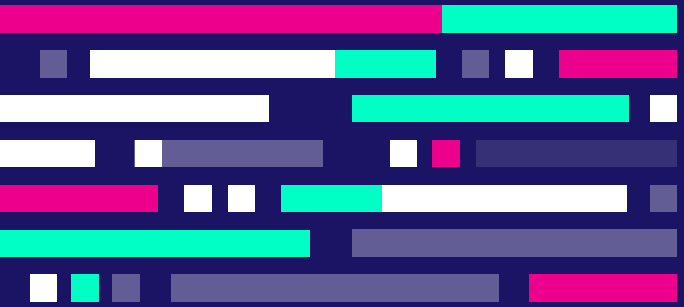
- Can be assigned as variables
  - Is_married = true
- Or may appear as condition as well
  - a = 20
  - b = 30
  - if a>b:

    print(A)

    else:

    print(B)
- Booleans can be:
  - True / False
  - 1 / 0
  - Any number can be true except 0
  - Same is for values
- Very useful

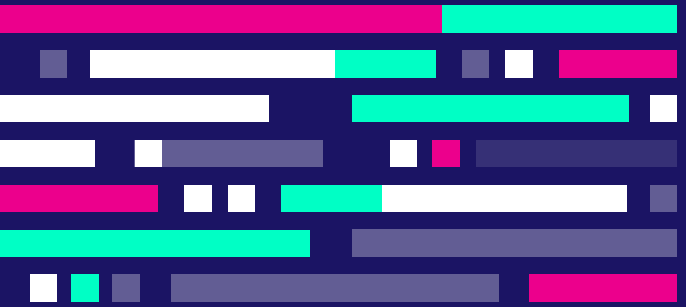
Code Camp | Alpha



# Strings

- Strings in python are surrounded by either single quotation marks, or double quotation marks.
  - “hello” or ‘hello’ will do the same
- Strings can be assigned normally
  - Name = “Wasiq”
- Multiline Strings use “””
  - a = “”” this is a
    multiline string”””
- Strings are arrays! (We will discuss)

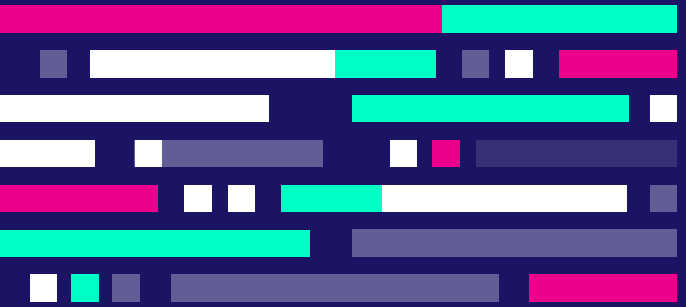
Code Camp | Alpha



# Operators

- Arithmetic operators
  - + addition
  - - subtraction
  - * multiplication
  - / division
  - % modulus
  - ** exponent
  - // floor division
- Assignment operators
  - = mostly used
  - += or -= are incremental, used in loops
- Comparison operators
  - == equal to
  - != not equal to
  - > greater than
  - < less than
  - >= greater than equal to
  - <= less than equal to

Code Camp | Alpha



# Operators

- Logical Operators
  - AND - true only if both conditions are true
  - OR - true if any condition is true
  - NOT - reverses the condition
- Identity Operators operators
  - is
  - Is not
- Membership Operators
  - in
  - not in
- ASSIGNMENT 2
  - Assignments will be separately given