Python: Data Types

Connect with me: Youtube | LinkedIn | WhatsApp Channel | Web | Facebook | Twitter

- Download PDF
- To access the updated handouts, please click on the following link: https://yasirbhutta.github.io/python/docs/data-types.html

Data Types in Python

In Python, data types define the kind of value a variable can hold and the operations that can be performed on it. They act as blueprints, specifying how data is stored and manipulated in your programs.

Video: Variables in Python

Here are some of the fundamental built-in data types in Python:

1. **Numeric Types:**
    - `int`: Stores whole (non-decimal) numbers, like `10`, `-5`, or `9999`.
    - `float`: Represents floating-point numbers with decimals, like `3.14`, `-2.5e2` (scientific notation), or `1.2345678901234567` (limited precision).
    - `complex`: Holds complex numbers with a real and imaginary part, like `3+2j` or `1.5-4.7j`.

- Example #1: How to use int variable
- Example #2: int variable
- Example #3: float variable

```
# Integer (int) to store age
age = 25

# Float (float) to store price with decimals
price = 14.99

# Complex number (complex) - not as common in everyday use
complex_num = 3 + 2j  # Imaginary unit represented by j
```

2. **String Type:**
    - `str`: Represents textual data enclosed in single or double quotes, such as `"Hello, world!"`, `'This is a string'`, or multi-line strings using triple quotes (''' or """).

- Example #1: How to Convert a Python String to int
- Example #2: How to Convert a Python integer to string
- Example #3: Convert integer to octal and hexadecimal

```
# String (str) to store a name
name = "Alice"
```

```
# String with a sentence
greeting = "Hello, how are you?"

# Multi-line string using triple quotes
message = """This is a message
that spans multiple lines."""
```

3. **Boolean Type:**
    - `bool`: Represents logical values: `True` or `False`. Used for conditional statements and boolean expressions.

- Example #1: Exploring Boolean Values and Type Checking with isinstance() and bool() functions

```
# Boolean (bool) for a true/false condition
is_raining = True

# Using booleans in an if statement
if is_raining:
    print("Bring an umbrella!")
```

Why Use Data Types?

Data types are essential in Python for several reasons:

- **Memory Management:** Different data types use memory in different ways. Knowing the type helps Python allocate the right amount of memory. For example, an integer requires less space than a string or a list.
    - video: How to Get the Size of an Object in Bytes | Python Tutorial for Beginners
- **Type Safety:** Data types help prevent errors by ensuring operations are compatible with the data being used. You can't add a string to an integer, for instance.
- **Readability:** Using appropriate data types makes code easier to understand. It's clear what kind of data a variable holds and how it can be used.
- **Performance:** Python can optimize certain operations based on the data type. For example, mathematical calculations on integers are faster than on floats.

Dynamic Typing

Python is a dynamically typed language. This means that the Python interpreter does type checking only as code runs, and the type of a variable is allowed to change over its lifetime.[1]

- Python Quiz -String
- Python Quiz - Scalar Types

Key Terms

True/False (Mark T for True and F for False)

1. In Python, the type of a variable is determined at runtime.

Answer Key (True/False):

1. True

Multiple Choice (Select the best answer)

1. **Which function would you use to determine the type of a variable in Python?**
    - A) id()
    - B) type()
    - C) str()
    - D) isinstance()

2. **Which of the following is not a scalar data type in Python?**
    - A) bool
    - B) int
    - C) float
    - D) list

3. Which data type is used to represent decimal numbers in Python?
    - A) int
    - B) float
    - C) complex
    - D) str

4. Which of the following is an example of a boolean value in Python?
    - a. "True"
    - b. 1
    - c. 3.14
    - d. False

5. Which scalar data type is used to represent textual data in Python?

- a. str
- b. char
- c. text
- d. string

6. What is the default type of a numerical literal without a decimal point in Python?

- a. int
- b. float
- c. complex
- d. bool

7. **What is the result of the expression type("Hello, World!") in Python?**
    - A) <class 'str'>
    - B) <class 'bool'>

    - C) <class 'int'>
    - D) <class 'float'>

8. What is the output of type(42)?

    - A) <class 'str'>
    - B) <class 'bool'>
    - C) <class 'int'>
    - D) <class 'float'>

9. What is the result of 3 + 4.5?

- a. 7
- b. 7.5
- c. Error
- d. None of the above

10. How do you create a string in Python?

    - A) Using single quotes (')
    - B) Using double quotes (")
    - C) Both a and b
    - D) None of the above

11. Which of the following is a valid boolean value in Python?

- a. True
- b. False
- c. 0
- d. All of the above

12. What is the output of str(3.14)?

- a. 3.14
- b. '3.14'
- c. Error
- d. None of the above

13. What is the result of the following expression?

```
type(3 + 4.0)
```

```
- A) <class 'str'>
- B) <class 'bool'>
- C) <class 'int'>
- D) <class 'float'>
```

14. Which of the following is a correct way to declare a complex number in Python?

- A) a = 3 + 4j
- B) a = 3.4j
- C) a = 3 + 4i
- D) a = 3 + 4

15. Which function can be used to convert a float to an integer in Python?

- A) float()
- B) int()
- C) str()
- D) bool()

16. Which of the following statements is true regarding dynamic typing in Python?

- A) Variables can only be assigned values of the same type.
- B) The type of a variable is determined at runtime based on the value it holds.
- C) Variables must be declared with a specific type.
- D) Once a variable is assigned a type, it cannot be changed.

17. In Python, what happens if you assign a new value of a different type to a variable?

- A) Python will raise a type error.
- B) Python will change the variable's type to match the new value.
- C) Python will ignore the new value.
- D) Python will convert the value to the original type.

18. What is the output of the following code?

```
x = 10
x = "Hello"
print(type(x))
```

```
- A) <class 'int'>
- B) <class 'str'>
- C) <class 'bool'>
- D) <class 'floa   - t'>
```

19. What is the result of the following code?

```
x = 5
x = 5.0
x = True
x = "Python"
print(x)
```

```
- A) 5
- B) 5.0
- C) True
- D) Python
```

20. What is the main advantage of dynamic typing in Python?

    - A) Faster execution time.
    - B) More flexibility in code.
    - C) Improved error detection at compile-time.
    - D) Reduced memory usage.

21. Which of the following best describes a dynamically typed language?

    - A) Type checking is performed during code compilation.
    - B) Type checking is deferred until program execution.
    - C) Type checking is not performed at all.
    - D) Types are always explicitly declared by the programmer.

22. In Python, which of the following is true about variable assignment?

    - A) The type of the variable is determined when the variable is first assigned a value.
    - B) The type of the variable is determined at compile-time.
    - C) Variables must be explicitly typed before assignment.
    - D) Variables cannot change type once assigned.

23. What is the output of the following code?

```
x = "10"
x = int(x) + 2
print(x)
```

```
- A) "102"
- B) 102
- C) 12
- D) "12"
```

Fill in the Blanks

Answer Key (Fill in the Blanks):

Exercises

- Find Occurrences in Strings Effortlessly with Python's Count Method
- How to Replace the Second Occurrence of a Character in a String

- String swapcase() Method

Python str examples:

- How to Reverse a String in Python
- Generate Random 4-Digit PIN Code in Python

Review Questions

References and Bibliography

[1]R. Python, “Dynamic vs Static – Real Python,” realpython.com. https://realpython.com/lessons/dynamic-vs-static/ [2]Python Software Foundation, “Built-in Types — Python 3.12.1 documentation,” Python.org, 2019. https://docs.python.org/3/library/stdtypes.html [3]“PEP 526 – Syntax for Variable Annotations | peps.python.org,” peps.python.org. https://peps.python.org/pep-0526/