

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>

Video: [Variables in Python](#)

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.

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!")
```

See also:

- [Variables in Python](#) **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.
- [Python Quiz -String](#)
- [Python Quiz - Scalar Types](#)

Key Terms

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

Answer Key (True/False):

Multiple Choice (Select the best answer)

1. Which data type is used to represent decimal numbers in Python?

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

2. Which of the following is an example of a boolean value in Python?

- a. "True"
- b. 1
- c. 3.14
- d. False

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

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

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

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

5. Which of the following is NOT a scalar data type in Python?

- a. Integer
- b. Float
- c. String
- d. List

6. What is the output of `type(42)`?

- a. int
- b. float
- c. str
- d. None

7. What is the result of `3 + 4.5`?

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

8. 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
9. Which of the following is a valid boolean value in Python?
- a. True
 - b. False
 - c. 0
 - d. All of the above
10. What is the output of `str(3.14)`?
- a. 3.14
 - b. '3.14'
 - c. Error
 - d. None of the above

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

- [Built-in Types - Python 3.12.1 documentation](#)