



Practice Problems

Quiz 00 Practice Problems

The following questions will help solidify concepts learned in class, as well as prepare you for the quizzes and final!

If you feel overwhelmed by the number of questions here, start with just a couple from each section that you think would help fill in any knowledge gaps you have. Then, work through the remaining questions as needed for additional practice.

The quiz itself will be similar in difficulty to these practice questions. In addition to these questions, you should review all of your lesson responses and challenge questions on Gradescope.

If you find yourself feeling lost, please ask for help in office hours or tutoring.

Multiple Choice and True/False

1. What is a **bool** data type in Python?

- a. Data type for storing text
- b. Data type for storing whole numbers such as **-10**, **3**, or **100**
- c. Data type for storing True/False values
- d. Data type for storing any type of information
- e. Data type for storing numbers with a decimal point such as **1.0**, **3.14**, or **-0.1**

► SHOW SOLUTION

2. What is an **int** data type in Python?

- a. Data type for storing text
- b. Data type for storing whole numbers such as **-10**, **3**, or **100**
- c. Data type for storing True/False values
- d. Data type for storing any type of information
- e. Data type for storing numbers with a decimal point such as **1.0**, **3.14**, or **-0.1**

► SHOW SOLUTION

3. What is a **str** data type in Python?

- a. Data type for storing text
- b. Data type for storing whole numbers such as **-10**, **3**, or **100**
- c. Data type for storing True/False values
- d. Data type for storing any type of information
- e. Data type for storing numbers with a decimal point such as **1.0**, **3.14**, or **-0.1**

► SHOW SOLUTION

4. An **int** literal can begin with any number of zeroes.

- a. True
- b. False

► SHOW SOLUTION

Overview

- Quiz 00 Practice Problems
- Multiple Choice and True/False
- Expressions
- Functions
- Function Signatures



- b. `"4"`
- c. `4.0`
- d. `"4.0"`

► SHOW SOLUTION

6. What function can you use to determine the type of an object in Python?

- a. `print()`
- b. `str()`
- c. `len()`
- d. `type()`

► SHOW SOLUTION

7. What is the *type* of the following expression?

```
1.5 + 2
```

- a. `int`
- b. `float`
- c. `str`
- d. `bool`
- e. `TypeError`

► SHOW SOLUTION

8. What is the *type* of the following expression?

```
len("cottage")
```

- a. `int`
- b. `float`
- c. `str`
- d. `bool`
- e. `TypeError`

► SHOW SOLUTION

9. What is the result of the following expression?

```
"110" + "110"
```

- a. `220`
- b. `"110110"`
- c. `TypeError`
- d. `"220"`
- e. `"110""110"`

► SHOW SOLUTION



- a. 2
- b. 20.4"
- c. "20"
- d. `TypeError`
- e. 2.0

► SHOW SOLUTION

11. What is the *type* of this value in Python?

`"True"`

- a. `bool`
- b. `str`
- c. `TypeError`
- d. `int`
- e. `float`

► SHOW SOLUTION

12. What *value* will the following expression evaluate to?

`"map"[1]`

- a. `"ap"`
- b. `"m"`
- c. `str`
- d. `"a"`
- e. `TypeError`

► SHOW SOLUTION

13. What will the following expression evaluate to?

`int(3.8 + 1)`

- a. 4.8
- b. 5
- c. 3.81
- d. 4
- e. `TypeError`

► SHOW SOLUTION

Expressions

1. What are the *types* of the following expressions and what *values* do they evaluate to? If a `TypeError` occurs, just write `TypeError`.

1.1. `1.5 + 2`



- 1.4. `str(110) * 2.1`
- 1.5. `float("100.0") / 20`
- 1.6. `21 // 2 ** 2 + 3`
- 1.7. `float("220") >= float("100" + "100")`
- 1.8. `int("COMP 110"[5]) + 99.0`
- 1.9. `(42 % 4) == (79 % 11)`
- 1.10. `int(4.99)`

► SHOW SOLUTION

2. Which of the following expressions correctly *concatenates* two strings together?

- a. `"Michael" * "Jordan"`
- b. `"Michael" + "Jordan"`
- c. `"Michael" , "Jordan"`
- d. `"Michael" : "Jordan"`

► SHOW SOLUTION

3. When using subscription syntax, what *index* does Python start with?

- a. `-1`
- b. `0`
- c. `1`
- d. `""`
- e. `True`

► SHOW SOLUTION

4. What value would you substitute for `x` to make the following expression True? Note: There is only one correct value here.

`(3 + x) == ((55 // 11) ** 2)`

► SHOW SOLUTION

5. Use subscription notation, string concatenation, and the string `"nevermind"` to write an expression that evaluates to `"nvm"`. Hint: Break this problem down into smaller pieces by first thinking of how you would write an expression using `"nevermind"` and subscription notation that evaluates to `"n"`.

► SHOW SOLUTION

Functions

1. What is a function call, and how does it differ from a function definition?

► SHOW SOLUTION

2. What is the difference between parameters and arguments in the context of functions?

► SHOW SOLUTION



4. If you never encounter a return statement in your function, what is your return value?

► SHOW SOLUTION

5. What does the `len()` function do in Python?

- a. Converts a value to a string
- b. Rounds a number to the nearest whole number
- c. Returns the length of a sequence, such as a `str`
- d. Converts a string to a number
- e. Counts the digits in an int

► SHOW SOLUTION

6. Given the following function definition, answer the following questions.

```
1 def evaluate_length(name: str) -> int:
2     """This function returns the length of the name."""
3     return len(name)
```

6.1. What is the result of the following function call? `evaluate_length("airplane")`

6.2. What is the *type* of the parameter `name`? What is the *return type* of this function?

► SHOW SOLUTION

7. Given the following function definition, answer the following questions.

```
1 def tablespoons_to_teaspoons(tablespoons: int) -> str:
2     """This functions tells you how many teaspoons are in the given nu
3     return str(tablespoons * 3) + " teaspoons"
```

7.1. Write a function call to `tablespoons_to_teaspoons` that returns the string `"9 teaspoons"`.

7.2. What is the *type* of the parameter `tablespoons`? What is the *return type* of this function?

► SHOW SOLUTION

Function Signatures

1. What is a function signature, and why is it significant?

► SHOW SOLUTION

2. Write the function signature for a function called `pos_or_neg` that takes as input an integer and returns `"Positive"` if the integer is positive and returns `"Negative"` if the integer is negative. Use `number` as the name of the parameter. Note: You are not writing the body of this function so do not worry about how it would actually work on the inside for this question.

► SHOW SOLUTION

3. Write the function signature for a function called `gcd` that takes two integers as input and returns the integer that is their greatest common divisor. Use `num_one` and `num_two` as your parameter names. Note: You are not writing the body of this function so do not worry about how it would actually work on the inside for this question.

► SHOW SOLUTION



Contributor(s): Alyssa Lyke, team110, IZZI Links, Benjamin Edwards