#### **Part 1:**

#### If you leave out one of the quotation marks or both while trying to print a string, it will result in a syntax error because the string needs to be enclosed in matching quotation marks in order for Python to recognize it as a string. For example:

>>> print("Hello)

File "<stdin>", line 1

print("Hello)

^

SyntaxError: EOL while scanning string literal

If you use the ++ operator, Python will not recognize it as a valid operator and will result in a syntax error because in Python, it's not a valid operator and the + operator is used for addition.

>>> 2++2

File "<stdin>", line 1

2++2

^

SyntaxError: invalid syntax

>>> 2--2

File "<stdin>", line 1

2--2

^

SyntaxError: invalid syntax

>>> 2+-2

File "<stdin>", line 1

2+-2

^

SyntaxError: invalid syntax

>>> 2-+2

File "<stdin>", line 1

2-+2

^

SyntaxError: invalid syntax

If you try to use a leading zero in a number like 02, Python will interpret it as an octal number and will result in a syntax error because 8 and 9 are not valid octal digits.

>>> 02

File "<stdin>", line 1

02

^

SyntaxError: invalid token

If you have two values with no operator and a space between them, Python will interpret it as two separate statements and will result in a syntax error because the statements need to be separated by a newline or a semicolon.

>>> 2 2

File "<stdin>", line 1

2 2

^

SyntaxError: invalid syntax

#### **Part 2:**

***Experiment with mathematical operations:***

Python

# Experiment with mathematical operations

a = 5

b = 3

c = a + b

print("The value of c after adding a and b is:", c)

d = a - b

print("The value of d after subtracting b from a is:", d)

e = a \* b

print("The value of e after multiplying a and b is:", e)

f = a / b

print("The value of f after dividing a by b is:", f)

**Output:**

The value of c after adding a and b is: 8

The value of d after subtracting b from a is: 2

The value of e after multiplying a and b is: 15

The value of f after dividing a by b is: 1.6666666666666667

This experiment demonstrates how to use Python for basic mathematical operations such as addition, subtraction, multiplication and division. It also shows how to assign values to variables and how to print the results of the operations.

***Experiment with string operations:***

python

# Experiment with string operations

name = "Dove"

age = 22

greeting = "My name is " + name + " and I am " + str(age) + " years old."

print(greeting)

**Output:**

My name is Dove and I am 22 years old.

This experiment demonstrates how to use Python to work with strings. It shows how to concatenate strings using the + operator and how to convert a number to a string using the str() function. It also shows how to assign values to variables and how to print the results.

***Experiment with conditional statements:***

python

# Experiment with conditional statements

x = 10

if x > 0:

print("x is positive")

elif x < 0:

print("x is negative")

else:

print("x is zero")

**Output:**

x is positive

This returns x is positive and returns vice versa when a negative integer is applied.

This experiment demonstrates how to use Python's conditional statements. It shows how to use the if, elif and else statements to check for different conditions and how to execute different code based on the results of those conditions. It also shows how to use comparison operators such as >, < and ==.

#### **References**

Downey, A. (2015). *Think Python: How to think like a computer scientist*. Green Tree Press.