Exercises (Solution)

Combining strings using concatenation

1. Create two string variables, 'greeting' and 'name', and concatenate them to print a personalized greeting.

```
greeting = "Hello, "
name = "world!"
print(greeting + name)
```

2. Declare two more string variables, 'first_name' and 'last_name', and concatenate them to print the full name.

```
first_name = "John"
last_name = "Doe"
print("Full Name: " + first_name + " " + last_name)
```

Arithmetic expressions in Python

1. Calculate and print the result of the expression: x = 10 + 5 * 2

```
x = 10 + 5 * 2
print("Result of x:", x)
```

2. Create a new variable 'y' and assign the result of the expression: y = 3 ** 2

```
y = 3 ** 2
print("Result of y:", y)
```

Exponentiation in Python

1. Calculate and print the result of 2 raised to the power of 4 using the exponentiation operator.

```
result = 2 ** 4
print("2 to the power of 4:", result)
```

Modulo operator in Python

1. Calculate and print the remainder when 15 is divided by 4 using the modulo operator.

```
x = 15 % 4
print("Remainder:", x)
```

Augmented assignment in Python

1. Initialize variables x and y with some values.

```
x = 5
y = 3

# 2. Use augmented assignment to add y to x and print the result.
x += y
print("Result of x += y:", x)
```

Comments in Python

1. Add comments to the following code explaining the purpose of each line.

```
# Calculate the sum of two numbers
num1 = 10
num2 = 20
sum_result = num1 + num2

# Print the result
print("Sum:", sum_result)
```

Understanding data types - int vs float

1. Declare an integer variable 'integer_var' and a float variable 'float_var'.

```
integer_var = 5
float_var = 3.14
```

2. Print the data type of each variable.

```
print("Type of integer_var:", type(integer_var))
print("Type of float_var:", type(float_var))
```

Multi-line string in Python

1. Create a multi-line string containing your address.

```
address = """123 Main Street
Cityville, USA
Zip: 12345"""
```

2. Print the multi-line string.

```
print(address)
```

Booleans in Python

1. Declare boolean variables indicating weather conditions.

```
is_sunny = True
is_raining = False
```

2. Print the values of the boolean variables.

```
print("Is it sunny?", is_sunny)
print("Is it raining?", is_raining)
```

Type error in Python

1. Fix the type error in the following statement and print the corrected string.

```
f1 = 0.25
f2 = 40.0
p = f1 * f2
bs = "The price is " + p
print(bs)
```

Fixed:

```
f1 = 0.25
f2 = 40.0
p = f1 * f2
bs = "The price is " + str(p)
print(bs)
```

For loop with countries

1. Create a list of countries: Spain, France and Germany.

```
countries = ["Spain", "France", "Germany"]
```

2. Use a for loop to print a message for each country.

```
for country in countries:
    print("The country is " + country)
```

For loop with numbers

1. Create a list of numbers: [1, 2, 3].

```
numbers = [1, 2, 3]
```

2. Use a for loop to calculate the product of all numbers and print the result.

```
total = 1
for num in numbers:
    total *= num

print("Product of numbers:", total)
```

Return statements in Python

1. Define a function 'calculate_sum' that takes two parameters and returns their sum.

```
def calculate_sum(a, b):
    return a + b
```

2. Use the function to calculate and print the sum of 7 and 3.

```
result = calculate_sum(7, 3)
print("The sum is:", result)
```

3. Create a function 'is_positive' that takes a number as a parameter and returns True if it's positive, False otherwise.

```
def is_positive(number):
    return number > 0
```

4. Test the 'is_positive' function with both positive and negative numbers and print the results.

```
print(is_positive(5))  # Should print True
print(is_positive(-2))  # Should print False
```

Defining a square function and calling it

1. Define a function 'square' that takes a number as a parameter and returns its square.

```
def square(num):
    return num * num
```

2. Use the 'square' function to calculate and print the square of 8.

```
result = square(8)
print("The square is:", result)
```

3. Create a function 'calculate_area' that calculates the area of a square given its side length. Use the 'square' function to find the area of a square with side length 5.

```
def calculate_area(side_length):
    return square(side_length)

area = calculate_area(5)
print("The area of the square is:", area)
```

Using a for loop

1. Write a program that uses a for loop to print the numbers from 1 to 5.

```
for i in range(1, 6):
    print(i)
```

Comparison operators in Python

1. Create a function 'compare_numbers' that takes two numbers as parameters and prints a message indicating which number is greater or if they are equal.

```python
def compare_numbers(x, y):
    if x > y:
        print(f"{x} is greater than {y}.")
    elif x < y:
        print(f"{x} is less than {y}.")
    else:
        print(f"{x} is equal to {y}.")
```

2. Test the 'compare_numbers' function with different pairs of numbers.

```python
compare_numbers(7, 3)
compare_numbers(5, 8)
compare_numbers(4, 4)
```