Exercises_wk1 (including solutions)

Q1: Define two variables ('word1' and 'word2') with respectively the following values ('Data' and 'Analytics'). Next, first use concatenation to print the following statement: "Data Analytics", and then print the same statement without using concatenation.

In [1]:
```
word1 = 'Data'
word2 = 'Analytics'
print(word1 + ' ' + word2)  # Note that a space needs to be added.
print(word1, word2)  # Note that now no space needs to be added.
```

```
Data Analytics
Data Analytics
```

Q2: Print the following three statements: "Phew, that's a small number!", "Phew, that is a "small" number!" and "Phew, that's a "small" number!".

In [2]:
```
print("Phew, that's a small number!")
print('Phew, that is a "small" number!')
print("Phew, that's a " + '"small"' + " number!")
```

```
Phew, that's a small number!
Phew, that is a "small" number!
Phew, that's a "small" number!
```

NOTE: An alternative way of printing the third statement of this exercise is by using a so-called escape character. Escape characters are read in by the defining backslash "\" character. Whatever comes after that backslash is what the computer will interpret. Some popular escape characters are "\t" for a tab indentation and "\n" which means newline and acts as if you hit the enter key. Try it! If we would have used an escape character, the third statement would read either:

- print('Phew, that\'s a "small" number!')

Or:

- print("Phew, that's a \"small\" number!")

Q3: Define three variables ('num1', 'num2' and 'num3') with respectively the following values (100, 0.25, '25'). Next, print the data type of each of the three variables, and then try to convert 'num1' into a string, and 'num2' and 'num3' into integers, and print the converted values. Carefully examine what happens!

In [3]:
```
num1 = 100
num2 = 0.25
num3 = '25'

print(type(num1))
print(type(num2))
print(type(num3))

print(str(num1))
print(int(num2))  # Note that converting a floating-point number to an integer rounds down.
print(int(num3))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
100
0
25
```

Q4: One by one, perform the following mathematical operations: add 5 to 10, subtract 5 from 10, multiply 10 by 5, divide 10 by 5, and raise 10 to the power of 5, and print them. Carefully examine what happens!

In [4]:
```
print(10 + 5)
print(10 - 5)
print(10 * 5)
print(10 / 5)  # Note that dividing two numbers (always) provides a floating-point number.
print(10 ** 5)
```

```
15
5
50
2.0
100000
```

Q5: Use floored (// integer) division and the modulus operator (%) to respectively calculate the quotient and remainder of the following division: 1235 divided by 125, and print the output. Next, use the modulus operator (%) to extract the right-most digit from the remainder, and print it. Then do the same for the last two digits.

In [5]:
```
quotient = 1235 //125
print(quotient)

remainder = 1235 % 125
print(remainder)

print(remainder % 10)
print(remainder % 100)
```

```
9
110
0
10
```

Q6: Define two variables ('num1' and 'num2') with respectively the following values ('1235' and '150'; i.e., as strings). Then define a variable ('a') by dividing 'num1' by 'num2', and print it. Next, use the round() function (see the note) to print variable 'a' with only two digits. Then define a variable ('b') by converting variable 'a' to an integer, and print it. Finally, define a variable ('c') by converting variable 'b' to a floating-point number, and print it. Carefully examine what happens! (Note: The syntax of the round() function is: round(number, number of digits).)

In [6]:
```
num1 = '1235'
num2 = '150'
a = int(num1) / int(num2)
print(a)  # Note that dividing two numbers (always) provides a floating-point number.
print(round(a, 2))
b = int(a)
print(b)  # Note that converting a floating-point number to an integer rounds down.
c = float(b)
print(c)  # Note that converting the rounded integer back to a floating-point number does not
provide the original \
          # floating-point number.
```

```
8.233333333333333
8.23
8
8.0
```

Q7: Define a variable 'name' by asking the user to input his/her name, and print the data type of this variable. Then do the same for the age of the user. Carefully examine what happens!

In [7]:
```
name = input('What is your name?: ')
print(type(name))
age = input('What is your age?: ')
print(type(age))  # Note that the input() function (always) provides a string.
```

```
What is your name?: Martijn
<class 'str'>
What is your age?: 50
<class 'str'>
```

Q8: Define two variables ('number' and 'exchange_rate') with respectively the following values (100 and 1.10522). Then use concatenation to print the following statement: "Given the exchange rate of 1.10522, 100 British Pounds is currently equal to 110.522 Euros."

In [8]:
```
number = 100
exchange_rate = 1.10522

print('Given the exchange rate of ' + str(exchange_rate) + ', ' + str(number) + ' British Poun
ds is currently equal to ' \
      + str(number * exchange_rate) + ' Euros.')
```

```
Given the exchange rate of 1.10522, 100 British Pounds is currently equal to 110.522 Euros.
```

Q9: Write a program that asks the user to input a (integer) number, and then based on whether this number is smaller than 0 or not, either prints "This is a negative number." or "This is a positive number.".

In [9]:
```
num = int(input('Please input a (integer) number: '))

if num < 0:
    print('This is a negative number.')
else:
    print('This is a positive number.')
```

```
Please input a (integer) number: 5
This is a positive number.
```

Q10: Write a program that first defines two variables ('num1' and 'num2') by asking the user to input two (integer) numbers, and then based on whether dividing the first number by the second provides a whole number or not, prints either "Dividing the first number by the second provides a whole number." or "Dividing the first number by the second does not provide a whole number.".

In [10]:
```
num1 = int(input('Please input a (integer) number: '))
num2 = int(input('Please input a second (integer) number: '))

if num1 % num2 == 0:
    print('Dividing the first number by the second provides a whole number.')
else:
    print('Dividing the first number by the second does not provide a whole number.')
```

```
Please input a (integer) number: 10
Please input a second (integer) number: 5
Dividing the first number by the second provides a whole number.
```