PythonNotebook1_solution_2023

November 21, 2023

Exercise 1.4.1

You know that your taxes is the product of your income and the tax rate. Complete the line below to assign the correct calculation for the my_taxes variable. Use the my_income and tax_rate variables for that.

The dotted lines (...) below mark the location where you should change the code yourself!

[ ]:
```
my_income = 100
tax_rate = 0.1
my_taxes = ...

###BEGIN SOLUTION TEMPLATE=
my_taxes = my_income * tax_rate
###END SOLUTION
```

Exercise 1.4.2

What is the type of the result of the expression (3 + 1.5 + 4?(Assign the type to the variable beneath (e.g. if you think it is an integer then expression_type = int)

[ ]:
```
expression_type = ...
###BEGIN SOLUTION TEMPLATE=
expression_type = float
###END SOLUTION
```

Exercise 1.4.3

Print the letter 'w' of the sentence 'Hello world!' using indexing of the variable sentence.

[ ]:
```
sentence = 'Hello world!'
letter_w = ...

###BEGIN SOLUTION TEMPLATE=
letter_w = sentence[6]
###END SOLUTION
print(letter_w)
```

(Fixing) Exercise 1.4.4

There are 2 types of numbers in Python: integer numbers and floating point numbers. Both ha
Beneath you can see a small dataset of the measured daily temperatures and a filter function, w

[ ]:
```
temp_data = [12.0, 13.5, 11.0, 12.7, 67.5, 11.8] # temperature measurements in␣
↪Delft, in Celsius

# A function to remove erroneous measurements
def filter_errors(data, error_id):
    print('Data before filtering', data)

    del temp_data[error_id - 1] # deleting the erroneous measurement

    print('Data after filtering', data)
```

In the code cell below a student wanted to indicate the position of the erroneous measurement, so it can be later filtered out during data analysis. However, it wasn't removed. The filter is known to work flawlessly (no errors there), meaning that the mistake happened in the indication step. It is known that the corrupted sample is the 5th in the dataset (5th because the first element has index 0! Python has a zero-based indexing). Your task is to fix it and remove the error!

[ ]:
```
error_id = 4 # index of erroneous measurement
###BEGIN SOLUTION TEMPLATE=
error_id = 5
###END SOLUTION
filter_errors(temp_data, error_id)
```

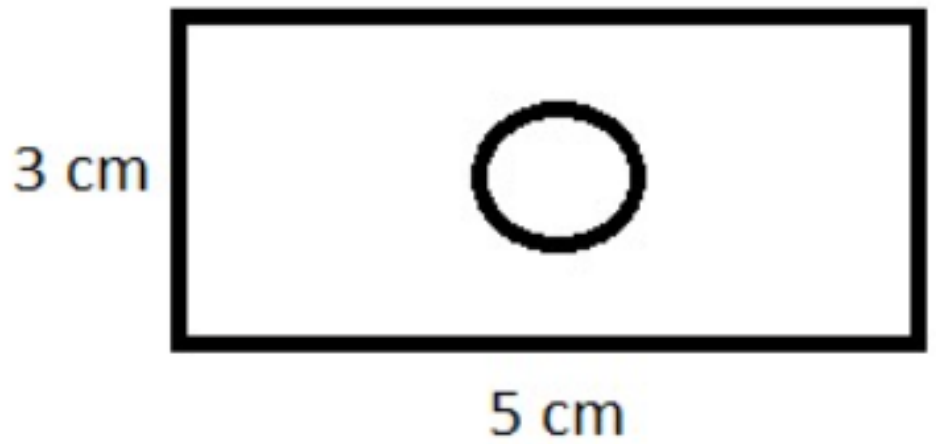
(Searching) Exercise 1.4.5 There are also other data types in Python, which were not mentioned in this part (in case they are not as useful or will be introduced later). For instance, Complex is one of them. The sad truth is that every software developer Googles a lot. It's not because she/he is lazy or incompetent. It is because programming languages are constantly updated and some tools are limited to a narrow field of applications. It is impractical to learn everything at once, therefore every coder has learned how to look up the functionality their task requires. Therefore, it is crucial for you to learn how to Google as well. Complex type is used in Python to represent complex numbers. If you haven't heard about them — a complex number is a number, which has a real and an imaginary part. For example, (x = 17 + 5i(. Here, (x(is a complex number with a real part of (17(and an imaginary part of (5(. Your task is to create a variable my_complex_num of Complex type and assign a (3 + 2i(value to it. For that you will have to Google a bit. Try to look for something like "Python complex variables". Python is very popular and you will be able to find everything you need. Make sure to filter the information you need — not everything you will find will be useful for this simple exercise.

[ ]:
```
my_complex_var = ...
###BEGIN SOLUTION TEMPLATE=
my_complex_var = 3 + 2j
###END SOLUTION
```

[ ]:
```
###BEGIN HIDDEN TESTS
assert type(my_complex_var) == complex, '1.4.5 Incorrect variable type'
assert my_complex_var == 3+2j, '1.4.5 Correct variable type but incorrect value'
###END HIDDEN TESTS
```

[ ]:
```
parenthesis_before = (2 + 4) * 3
print('With parenthesis before =',parenthesis_before)
parenthesis_after = 2 + (4 * 3)
print('With parenthesis after =',parenthesis_after)
```

Exercise 1.5.1 Given that we have a rectangle with a height of (3(cm, width of (5(cm and a cutout circle of radius (0.6(cm, as shown below.



What will the remainder area of the rectangle be? Hint: (A_{circle} = r^2(

[ ]:
```
# variables for the rectangle
rect_h = ...
rect_w = ...

# variables for the sphere
pi = 3.14159265359
r = ...

# calculating area
circle_area = ...
rect_area = ...
remaining_area = ...

###BEGIN SOLUTION TEMPLATE=
rect_h = 3
rect_w = 5
pi = 3.14159265359
r = 0.6

circle_area = pi * r ** 2
```

```
rect_area = rect_h * rect_w
remaining_area = rect_area - circle_area
###END SOLUTION
print(f'The remaining area is {remaining_area:.2f} cm^2')
```

Exercise 1.5.2Suppose a group of 5 engineers ordered 3 pizzas with 8 slices each. It is known that miners respect engineers and they want to share their meal equally. Your task is to calculate the amount of pizza slices each engineer will get and the amount of slices which will be left untouched. Use // and % operators.

[ ]:
```
pizza_slices = ...
grp_members = ...

slices_per_person = ...
left_over_slices = ...

###BEGIN SOLUTION TEMPLATE=
slices_per_person = pizza_slices // grp_members
left_over_slices = pizza_slices % grp_members
###END SOLUTION
print('Number of slices per person:', slices_per_person)
print('Number of slices left over:', left_over_slices)
```

[ ]:
```
print(num > 7)
```

Exercise 1.5.3 Assign any numbers to num_a and num_b variables in such a way that their comparison returns a True value.

[ ]:
```
num_a = ...
num_b = ...

###BEGIN SOLUTION TEMPLATE=
num_a = 2
num_b = 3
###END SOLUTION

comparison_result = num_a != num_b
print(comparison_result)
```

(Fixing) Exercise 1.5.4

Recall that Sentinel data titles are formatted as: S1A_IW_SLC___1SDV_20181205T015821_20181205T015851_0 where each part means something, e.g., S1A means Sentinel-1A.An ESA engineer wrote a piece of code to figure out if the data she has belongs to the Remote Sensing Satellite Sentinel-1; and, if it does, she would like to know the time of overpass… but for some reason she keeps getting an error. Could you maybe help her out?

[ ]:
```
import datetime
data = 'S1A_IW_SLC__1SDV_20220218T231336_20220218T231403_041974_04FFAC_A40F'
```

```
platform_name = data[0:3]#selecting platform name
print('Platform name:', platform_name)

is_platform_name_correct = platform_name = 'S1A' #checking platform

###BEGIN SOLUTION TEMPLATE=
is_platform_name_correct = platform_name == 'S1A' #checking platform
###END SOLUTION

if is_platform_name_correct == True:
    date = data[17:32]
    print('Sentinel-1 overpassed at {}'.format(datetime.datetime.strptime(date,␣
 ↪"%Y%m%dT%H%M%S")))
else:
    print('Data is not from Sentinel-1 and will not be processed')
```

(Searching) Exercise 1.5.5 In Python, there are a lot of shortcuts! There are multiple ways to solve tasks (e.g. reading data, filtering noise, solving systems of equations, etc) and there is, usually, a way to write a "one-liner". A "one-liner" is a code, which solves the task in one line. Sometimes "one-liners" are not preferred (it usually lowers the readability of your code) but it is still useful to think about them. Think about "How can I solve this problem differently?", or "How can I use Python most efficiently?".

For example, in the cell bellow you can see a small combination of comparison operators.

[ ]:
```
my_number = 5
result = 3 < my_number and my_number < 6
print(f'My number {my_number} is within (3, 6) range:', result)
```

This expression checks whether: first, <code>my_number</code> is bigger than \(3\(and then, s

[ ]:
```
# use this line to create a chained comparison
...
###BEGIN SOLUTION TEMPLATE=
result = 3 < my_number < 6
print(f'My number {my_number} is within (3, 6) range:', result)
###END SOLUTION TEMPLATE=
```