Homework_1_Notebook

March 23, 2025

1 DSE5002 Homework/Lab 01

2 Beija Richardson 3/21/25

These problems are all in the textbook, I just cut and pasted the questions into this Python Notebook

In future homeworks, you will have to constuct the notebook yourself, I'm trying to go easy on you this week.

Exercises from

https://allendowney.github.io/ThinkPython/chap01.html

and

https://allendowney.github.io/ThinkPython/chap02.html

Run these examples in an AI LLM model (chatbot, Claude, etc, take your pick)
Briefly comment in the cell below on how well this worked- only a few sentences are needed.

1.9.1. Ask a virtual assistant

As you work through this book, there are several ways you can use a virtual assistant or chatbot to help you learn.

If you want to learn more about a topic in the chapter, or anything is unclear, you can ask for an explanation.

If you are having a hard time with any of the exercises, you can ask for help.

In each chapter, I'll suggest exercises you can do with a virtual assistant, but I encourage you to try things on your own and see what works for you.

Here are some topics you could ask a virtual assistant about:

Earlier I mentioned bitwise operators but I didn't explain why the value of 7 ^ 2 is 5. Try asking "What are the bitwise operators in Python?" or "What is the value of 7 XOR 2?"

I also mentioned the order of operations. For more details, ask "What is the order of operations in Python?"

The round function, which we used to round a floating-point number to the nearest integer, can take a second argument. Try asking "What are the arguments of the round function?" or "How do I round pi off to three decimal places?"

There’s one more arithmetic operator I didn’t mention; try asking “What is the modulus operator in Python?”

Most virtual assistants know about Python, so they answer questions like this pretty reliably. But remember that these tools make mistakes. If you get code from a chatbot, test it!

2.1 I asked “What are the bitwise operators in Python?” it stated “&, |, ^, ~, «, »”

2.2 I asked “What is the order of operations in Python?” it listed the specific order starting with 1. Parentheses, Brackets, Braces 2. Exponeation ** 3. Unary operators +x, -x, ~x and so forth.

2.3 I asked “What are the arguments of the round function?” it stated the round function in Python takes two arguments ; number and ndigits.

2.4 I asked ” What is the modulus operator in Python?” it states ” The modulus operator in Python is the percent sigh %. It returns the remainder of dividing the left operand by the right operand.”

1.9.3. Exercise

When you learn about a new feature, you should try it out and make mistakes on purpose. That way, you learn the error messages, and when you see them again, you will know what they mean. It is better to make mistakes now and deliberately than later and accidentally.

You can use a minus sign to make a negative number like -2. What happens if you put a plus sign before a number? What about 2++2?

What happens if you have two values with no operator between them, like 4 2?

If you call a function like round(42.5), what happens if you leave out one or both paren

{Write the code that checks all of these }theses?

[3]:

```
# Enter your code here
```

[5]:

```
2++2
```

[5]:

```
4
```

[7]:

```
4 2
```

```
  Cell In[7], line 1
    4 2
      ^
SyntaxError: invalid syntax
```

[9]:

```
(42.5
  Cell In[9], line 1
    (42.5
         ^

SyntaxError: incomplete input
```

[ ]:

```
1.9.4. Exercise
Recall that every expression has a value, every value has a type, and we can use the type function to find the type of any value.

What is the type of the value of the following expressions? Make your best guess for each one, and then use type to find out.

765

2.718

'2 pi'

abs(-7)

abs(-7.0)

abs

int

type

{write the code below to check these}
```

[17]:

```
type(765)
```

[17]: int

[19]:

```
type(2.718)
```

[19]: float

[21]:

```
type(' 2 pi')
```

[21]: str

[23]:

```
type(abs(-7))
```

[23]: int

```
[25]: type(abs)
```

```
[25]: builtin_function_or_method
```

```
[27]: type(int)
```

```
[27]: type
```

```
[29]: type(type)
```

```
[29]: type
```

1.9.5. Exercise The following questions give you a chance to practice writing arithmetic expressions.

How many seconds are there in 42 minutes 42 seconds?

How many miles are there in 10 kilometers? Hint: there are 1.61 kilometers in a mile.

If you run a 10 kilometer race in 42 minutes 42 seconds, what is your average pace in seconds per mile?

What is your average pace in minutes and seconds per mile?

What is your average speed in mile

{Write your code below}s per hour?

```
[31]: (42*60) + 42
```

```
[31]: 2562
```

```
[33]: 10*0.621371
```

```
[33]: 6.21371
```

```
[43]: 2562/(10*0.621371)
```

```
[43]: 412.31406036007473
```

6 minutes and 52 seconds per miles

```
[45]: 6.21371/0.7117
```

```
[45]: 8.730799494168892
```

2.11.2. Exercise Repeating my advice from the previous chapter, whenever you learn a new feature, you should make errors on purpose to see what goes wrong.

We've seen that n = 17 is legal. What about 17 = n?

How about x = y = 1?

In some languages every statement ends with a semi-colon (;). What happens if you put a semi-colon at the end of a Python statement?

What if you put a period at the end of a statement?

What happens if you spell the name of a module wrong and try to import maath?

```
[47]: 17=n
```

```
  Cell In[47], line 1
    17=n
    ^
SyntaxError: cannot assign to literal here. Maybe you meant '==' instead of '='
```

```
[49]: x=y=1
```

```
[51]: 17;
```

```
[53]: 18.
```

```
[53]: 18.0
```

```
[55]: import maath
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[55], line 1
----> 1 import maath

ModuleNotFoundError: No module named 'maath'
```

2.11.3. Exercise Practice using the Python interpreter as a calculator:

Part 1. The volume of a sphere with radius is

. What is the volume of a sphere with radius 5? Start with a variable named radius and then assign the result to a variable named volume. Display the result. Add comments to indicate that radius is in centimeters and volume in cubic centimeters.

Part 2. A rule of trigonometry says that for any value of , . Let's see if it's true for a specific value of like 42.

Create a variable named x with this value. Then use math.cos and math.sin to compute the sine and cosine of , and the sum of their squared.

The result should be close to 1. It might not be exactly 1 because floating-point arithmetic is not exact—it is only approximately correct.

Part 3. In addition to pi, the other variable defined in the math module is e, which represents the base of the natural logarithm, written in math notation as . If you are not familiar with this value, ask a virtual assistant "What is math.e?" Now let's compute three ways:

Use math.e and the exponentiation operator (**).

Use math.pow to raise math.e to the power 2.

Use math.exp, which takes as an argument a value, , and computes .

You might notice that the last result is slightly different from the other two. See if you can find out which is correct.

3 V=(4/3)(3.1415926)(5)^3

```
[71]: x=42
```

```
[104]: import numpy as np
```

```
[108]: np.cos(x)
```

```
[108]: np.float64(-0.39998531498835127)
```

```
[110]: np.sin(x)
```

```
[110]: np.float64(-0.9165215479156338)
```

```
[114]: import math
```

```
[122]: print(math.e)
```

```
2.718281828459045
```

```
[128]: math.pow(math.e,2)
```

```
[128]: 7.3890560989306495
```

```
[138]: math.exp
```

```
[138]: <function math.exp(x, /)>
```

```
[ ]:
```