

ME 5194: Homework 1

Adrian Bakhtar

Exercise 1.1.

Whenever you are experimenting with a new feature, you should try to make mistakes. For example, in the "Hello, world!" program, what happens if you leave out one of the quotation marks? What if you leave out both? What if you spell print wrong?

This kind of experiment helps you remember what you read; it also helps when you are programming, because you get to know what the error messages mean. It is better to make mistakes now and on purpose than later and accidentally.

1. In a print statement, what happens if you leave out one of the parentheses, or both?

If one of the parentheses are left out of the print statement, a syntax error will occur with an indicator below where the parenthesis is expected to be.

If both parentheses are omitted, two things can happen. If there is no space between "print" and whatever was trying to be printed out, Python will treat it as its own variable and will output a NameError that it was not defined (i.e. printhello). If there is a space, Python will say that parentheses were missing in the print statement and suggest a fix

In [179...

```
print("hello"
```

```
File "<ipython-input-179-075bb1d15818>", line 1
  print("hello"
      ^
```

SyntaxError: unexpected EOF while parsing

In [180...

```
printhello
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-180-8abb30f7fec9> in <module>
----> 1 printhello
```

NameError: name 'printhello' is not defined

In [181...

```
print hello
```

```
File "<ipython-input-181-2a20a4ba3335>", line 1
  print hello
```

^

SyntaxError: Missing parentheses in call to 'print'. Did you mean print(hello)?

2. If you are trying to print a string, what happens if you leave out one of the quotation marks, or both?

If one of the quotations are missing from the print statement, a syntax error will occur with an indicator saying where Python thinks the second quotation should be.

If both quotations are missing, Python will take whatever was supposed to be between the quotations as the name of a variable and try to print the value of it. For example, `print(hello)` will attempt to locate the variable called `hello` and print whatever its value is. Since no variable named `hello` was ever created, a `NameError` is output that says `hello` was never defined.

In [182...

```
print(hello')
```

```
File "<ipython-input-182-02fc17ac4a3a>", line 1
    print(hello')
          ^
```

SyntaxError: EOL while scanning string literal

In [183...

```
print(hello)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-183-1cd80308eb4c> in <module>
----> 1 print(hello)
```

NameError: name 'hello' is not defined

3. 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`?

The number will either be positive or negative. For example, `x = +2` stores the value of positive 2 into the variable `x`. Also, `x = -2` will store negative 2 into the variable `x`. There can also be a combination: `x = -++++++2`. For this, a negative number will be stored for an odd amount of '-'s. For performing math operations, an even number of '-'s will default to addition.

In [198...

```
x = +2
y = ++++++++3
z = -+4
a = -+++++++5
b = 2-+++++++9

print(x)
print(y)
print(z)
```

```
print(a)
2
3
-4
-5
11
```

4. In math notation, leading zeros are ok, as in 09. What happens if you try this in Python? What about 011?

Leading zeros are not accepted for integers but are permissible for floats. A syntax error will be output if leading zeros are used on an integer variable.

In [185...

```
x = 011.0
print(x)

y = 011
print(y)
```

```
File "<ipython-input-185-b787c2b62749>", line 4
    y = 011
        ^
```

SyntaxError: leading zeros in decimal integer literals are not permitted; use an 0o prefix for octal integers

5. What happens if you have two values with no operator between them?

If there are two values next to each other with no operator between them, then it is read as one value. For example, 2 and 3 with no operator between them would be read as 23. When done with variables, a NameError will most likely occur saying that the variable is not defined unless there a variable with the name of the combined values was previously created.

In [199...

```
books = 4
pages = 1000

print(bookspages)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-199-a73ee6246df4> in <module>
      2 pages = 1000
      3
----> 4 print(bookspages)
```

NameError: name 'bookspages' is not defined

Exercise 1.2.

Start the Python interpreter and use it as a calculator

1. How many seconds are there in 42 minutes 42 seconds?

In [187...

```
minutes = 42
seconds = 42

print('There are {} seconds in 42 minutes and 42 seconds.'.format(minutes*60+seconds))
```

There are 2562 seconds in 42 minutes and 42 seconds.

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

In [188...

```
kilo= 10
kilo_in_mile = 1.61

print('There are {:.2f} miles in 10 kilometers.'.format(kilo/kilo_in_mile))
```

There are 6.21 miles in 10 kilometers.

3. If you run a 10 kilometer race in 42 minutes 42 seconds, what is your average pace (time per mile in minutes and seconds)? What is your average speed in miles per hour?

In [189...

```
distance_kilo = 10 #kilometers
distance_miles = distance_kilo/1.61 #miles

time_seconds = 42*60+42 #seconds

pace_seconds = time_seconds/distance_miles
pace_minutes = time_seconds/60/distance_miles
speed_mph = distance_miles/(time_seconds/3600)

print('The average pace was {:.2f} seconds per mile or {:.2f} minutes per mile.'.format(pace_seconds, pace_minutes))
print('The average speed was {:.2f} miles per hour.'.format(speed_mph))
```

The average pace was 412.48 seconds per mile or 6.87 minutes per mile.
The average speed was 8.73 miles per hour.

Exercise 2.1

Repeating my advice from the previous chapter, whenever you learn a new feature, you should try it out in interactive mode and make errors on purpose to see what goes wrong.

1. We've seen that `n = 42` is legal. What about `42 = n`?

`42 = n` is not a legal statement. It does not follow the correct naming structure, so a `SyntaxError` will occur.

In [190...

```
42=n
```

```
File "<ipython-input-190-94a448c80b86>", line 1
```

```
42=n
```

```
^
```

```
SyntaxError: cannot assign to literal
```

2. How about `x = y = 1`?

This is a valid way to declare and initialize `x` and `y` to have the value of 1.

This can be done with however many variables.

In [191...

```
x = y = z = a = 23
print(x)
print(y)
print(z)
print(a)
```

```
23
23
23
23
```

3. 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?

Putting a semi-colon, `;`, at the end of a line in Python allows the users to write multiple statements on a single line.

This can be useful when declaring and initializing variables.

In [192...

```
x = 123; y = 145; z = 1; a = -123
print(x); print(y); print(z); print(a)
```

```
123
145
1
-123
```

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

It depends what the statement is. For a simple numerical variable declaration such as `x = 2`, putting a period at the end of this statement will change the variable type of `x`. Initially an integer, `x = 2`. will make `x` a float variable. Periods are also used to access methods or properties of classes or functions. For example, a period is used to access the plotting functions of `matplotlib.pyplot`. A syntax error could occur here if nothing is typed after the period.

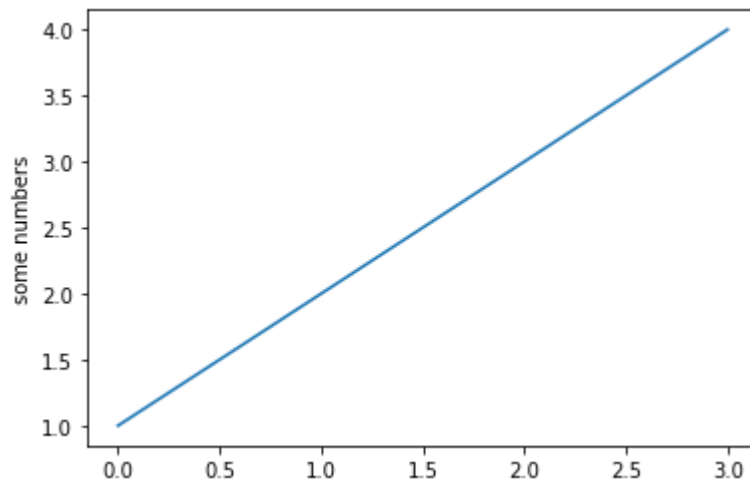
In [201...

```
x = 2
y = 2.

print(type(x))
print(type(y))

import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.show()
```

```
<class 'int'>
<class 'float'>
```



5. In math notation you can multiply x and y like this: xy . What happens if you try that in Python?

This will most likely cause a `NameError`. Python will search for a variable called `xy` and will return that it has not been defined if this is the case.

In [194...

```
x = 25
y = 3

print(x*y)
print(xy)
```

75

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-194-cd18c7d06cca> in <module>
      3
      4 print(x*y)
----> 5 print(xy)
```

NameError: name 'xy' is not defined

Exercise 2.2.

Practice using the Python interpreter as a calculator:

1. The volume of a sphere with radius r is $\frac{4}{3}\pi r^3$. What is the volume of a sphere with radius 5?

In [195...

```
import math

radius = 5
volume = 4/3*math.pi*radius**3

print('The volume of a sphere with a radius of 5 is {:.2f} units cubed.'.format(volume))
```

The volume of a sphere with a radius of 5 is 523.60 units cubed.

2. Suppose the cover price of a book is \$24.95, but bookstores get a 40% discount. Shipping costs \$3 for the first copy and 75 cents for each additional copy. What is the total wholesale cost for 60 copies?

```
In [219... book_price = 24.95 # in $
discount = 0.4
shipping_first = 3 # in $ for the first copy
shipping_additional = 0.75 # in $ for each additional copy
num_copies = 60

total = num_copies*book_price*(1-discount) + shipping_first + shipping_additional*(num_
print('The total wholesale cost for 60 copies is ${:.2f}'.format(total))
```

The total wholesale cost for 60 copies is \$945.45.

3. If I leave my house at 6:52 am and run 1 mile at an easy pace (8:15 per mile), then 3 miles at tempo (7:12 per mile) and 1 mile at easy pace again, what time do I get home for breakfast?

```
In [217... start_time = [6, 52] #units of hour, minute

first_pace = [8,15,1] #units of [minutes, seconds, miles]
second_pace = [7,12,3] #units of [minutes, seconds, miles]
third_pace = [8,15,1] #units of [minutes, seconds, miles]

running_time_min = first_pace[0]*first_pace[2]+second_pace[0]*second_pace[2]+third_pace
running_time_sec = first_pace[1]*first_pace[2]+second_pace[1]*second_pace[2]+third_pace

if running_time_sec > 60:
    running_time_sec = running_time_sec -60
    running_time_min +=1

print('Total running_time is {} minutes and {} seconds'.format(running_time_min, runnin

final_time = [start_time[0], start_time[1]+running_time_min, running_time_sec] #units o

if final_time[1]>60:
    final_time[1] = final_time[1]-60
    final_time[0]+=1

print('The runner returns at {}:{} AM.'.format(final_time[0], final_time[1]))
```

Total running_time is 38 minutes and 6 seconds

[7, 30, 6]

The runner returns at 7:30 AM