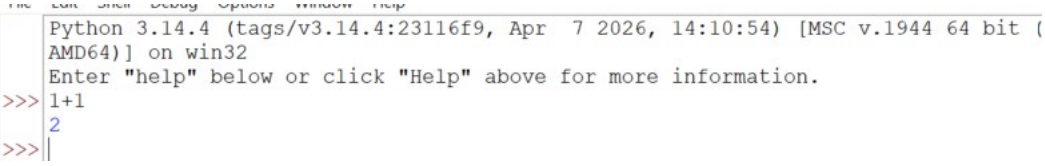
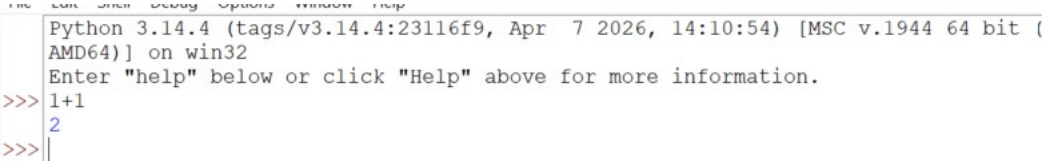
Exercises for Deck 1

Getting Started with Python
Dr. Hazel “twitch.tv/hazeldotzone” Campbell



1) In the above screenshot, circle the REPL input.



2) In the above screenshot, circle the REPL’s evaluation.

3) What does REPL stand for?

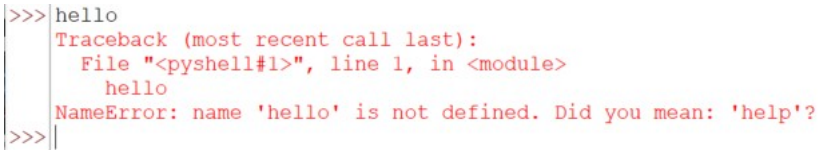
   a) R

   b) E

   c) P

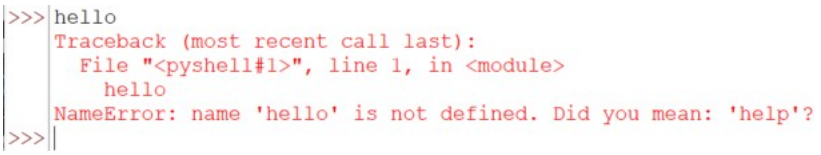
   d) L

4) In the screenshot to the right, what is the red text?



5) In the above screenshot, circle the traceback.

6) In the screenshot to the right, circle the error.



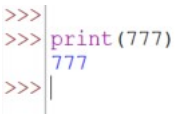
7) In the screenshot to the right, circle the REPL’s output.

8) Why does the REPL not show output when the input was “None”?

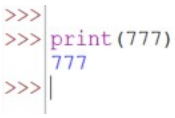
9) Other than input... what are the three kinds of things shown in the REPL?

10) In what two situations does the REPL not show the evaluation?

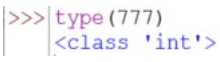
11) In the screenshots to the right, circle the identifiers.



12) In the screenshot to the right, circle the evaluation. Draw a box around the written output.



13) In the screenshot to the right, circle the evaluation. Draw a box around the written output.



14) In the screenshot to the right, label the following:

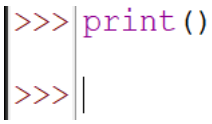
a) identifier

b) delimiters

c) REPL's evaluation

d) written output

e) arguments



15) In the screenshot to the right, label the following:

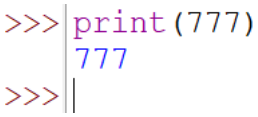
a) identifier

b) delimiters

c) REPL's evaluation

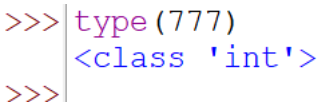
d) written output

e) arguments

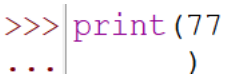


16) In the previous screenshot, the argument is also a(n) _____?

17) In the screenshot to the right, find the identifier and the literal.



18) In the screenshot to the right, determine where the arguments begin and where they end.



19) In the previous screenshot, without running the code... what do you expect the REPL's evaluation to be? What do you expect the written output to be?

20) In the screenshot to the right, what is evaluated first? What is evaluated second?
Is the shown "<class 'int'>" REPL's evaluation or written output?

```
>>> type(777)
<class 'int'>
```

21) In the screenshot to the right, what is evaluated first? What is evaluated second?
On the second line is"777" REPL's evaluation or written output?

```
>>> print(777)
777
```

22) Is your answer to the previous two questions different? Why?

23) In the screenshot to the right, what is evaluated first? Second? Third?
On the second line is "<class 'int'>" REPL's evaluation or written output?

```
>>> print(type(123))
<class 'int'>
```

24) Was your answer to (20) and (23) different? Why?

25) In the screenshot to the right, label the:

a) delimiters

b) literals

c) arguments

d) identifiers

e) written outputs

f) REPL's evaluations

```
>>> print(print(123))
123
None
```

26) In the screenshot to the right, label what is evaluated:

a) first

b) second

c) third

```
>>> print(print(123))
123
None
```

27) In the previous screenshot, what is the evaluation of the inner print?

28) In the previous screenshot, what is the argument of the outer print? What does it evaluate to?

29) In the screenshot to the right, circle the literals.

```
>>> 1+2*3
7
>>>
```

30) In the screenshot to the right, circle the operators.

```
>>> 1+2*3
7
>>>
```

31) In the screenshot to the right, circle the argument delimiters.

```
>>> type((4)+(2))
<class 'int'>
```

32) In the screenshot to the right, circle the expression delimiters.

```
>>> type((4)+(2))
<class 'int'>
```

a) Draw an expression diagram for the previous screenshot.

b) List the order that each thing in it is evaluated.

c) At the end, does REPL show the evaluation?

33) In the screenshot to the right, circle the integer literals.

```
>>> 2*3+4
10
```

a) Draw an expression diagram for the previous screenshot.

b) List the order that each thing in it is evaluated.

34) In the screenshot to the right, label each part of the input.

a) Draw the expression diagram.

b) List the order each thing in the REPL input is evaluated.

c) At the end, does REPL show the evaluation?

```
>>> print(2*(3+4))
14
```