

Name: _____ Partner: _____

Python Activity 29: Type Conversion

Learning Objectives

Students will be able to:

Content:

- Explain the use of built-in type conversion functions.
- Identify error message from type conversion.

Process:

- Write code to determine data type and convert between data types.

Prior Knowledge

- Activities 1-28, built-in data types, generators, range, classes

If you encounter any issues/typos, let Iris know! Questions? Ask Iris or the POGILing forum

Critical Thinking Questions:

1. Examine the sample code below from interactive python:

```
Interactive Python
0 >>> type(3.5)
1 <class 'float'>
2 >>> mynum = 4
3 >>> type(mynum)
4 <class 'int'>
5 >>> type('hello')
6 <class 'str'>
7 >>> type(4 == 3)
```

- a. What is the output for the following code?

`type(3.5)` → `<class '_____>`

`type(mynum)` → `<class '_____>`

`type('hello')` → `<class '_____>`

- b. What type of objects are 3.5, mynum, and 'hello'?

- b. What does the built-in function `type(object)` do?

- c. What are some additional example values of `object` to call `type(object)` on?

d. What might be the output for line 7? `<class '_____>`

2. Examine the sample code below from interactive python:

```
Interactive Python
0 >>> mynum = 4
1 >>> float(mynum)
2 4.0
3 >>> int(3.5)
4 3
```

a. What type of object is mynum? _____

b. What is returned on line 2? What is its relation to the value stored in mynum?

c. What does the built-in function `float(object)` do?

d. What is the built-in function `int(object)` doing on line 3?

3. Examine the sample code below from interactive python:

```
Interactive Python
0 >>> str(3.5)
1 '3.5'
2 >>> float('4.6')
3 4.6
4 >>> float('hello')
```

b. What is returned on line 1? What is its relation to the argument on line 0?

c. What is returned on line 3? What is its relation to the argument on line 2?

d. Line 4 results in the following error, why?
`ValueError: could not convert string to float: 'hello'`

4. Match up special methods on the left-hand column with the code that implicitly calls them in the right-hand column (make educated guesses using special method names and parameters!):

Type Conversion Built-in Function	Description
a. <code>int(object)</code>	Returns <code>object</code> as an immutable sequence
b. <code>float(object)</code>	Returns <code>object</code> as an integer
c. <code>str(object)</code>	Error thrown when <code>object</code> can't be converted.
d. <code>list(object)</code>	Returns <code>object</code> as a string of characters
e. <code>set(object)</code>	Returns <code>object</code> 's data type
f. <code>tuple(object)</code>	Returns <code>object</code> as a floating point number
g. <code>bool(object)</code>	Returns a list-version of <code>object</code>
h. <code>type(object)</code>	Returns <code>object</code> as a set of unique values
i. <code>ValueError: could not convert <origType> to <newType>: <value></code>	Returns <code>object</code> as True or False

Confirm your responses by checking the python3 documentation:

<https://docs.python.org/3/library/functions.html>

5. Examine the sample code below from interactive python:

```

Interactive Python
0 >>> r = range(1, 10, 2)
1 >>> r
2 range(1, 10, 2)
3 >>> list(r)
4 [1, 3, 5, 7, 9]

```

- a. What would be the output of `type(r)`? How do you know?
-
- b. If we were to print each of the values produced by `r`, what might they be?
-
- c. How does the actual output produced by line 1 differ from your previous answer?
-
- d. Why does python produce the output by the code on line 3?
-

6. Examine the sample code below from interactive python:

```
Interactive Python  
0 >>> def oneCharGen(word):  
1 ...     for ch in word:  
2 ...         yield ch  
3 >>> g = oneCharGen('Why?!')  
4 >>> type(g)  
5 <class 'generator'>  
6 >>> list(g)
```

- a. What type of object is `g`? How do you know? _____
- b. What type of object is yielded by `g`? How do you know? _____
- c. Write a few lines of code that will display all of the data types of the values generated by a `oneCharGen` generator object:

- d. What might the output of line 6 be? (*Hint*: it will be similar to the Q5L6 output).

- e. If we made a generator that produces an infinite stream of values, what might calling `list(generator)` on that object do? Try it out in python!

7. Examine the sample code below from interactive python:

```
Interactive Python  
0 >>> class Garden:  
1 ...     """ A User-defined type """  
2 >>> mygdn = Garden()  
3 >>> type(mygdn)  
4 <class '__main__.Garden'>  
5 >>> str(mygdn)  
6 '<__main__.Garden object at 0x1074e89e8>'
```

- a. What type of object is `mygdn`? _____
- b. How does `mygdn`'s type differ from the previous examples we've looked at?

- c. How does the output of `type(object)` on line 3 differ from previous examples?

-
- d. If we were to make a new user-defined type, `Flower`, and call `type(Flower())` what might the output be? _____
- e. What is the programmer trying to do on line 5? _____
- f. Is the output on line 6 in-line with what the programmer likely wants? Why/why not? _____
- g. How might the programmer achieve the desired output on line 6? _____
-

8. Examine the sample code below from interactive python:

Interactive Python	
0 >>> ord('a')	6 >>> chr(97)
1 97	7 'a'
2 >>> ord('b')	8 >>> chr(98)
3 98	9 'b'
4 >>> ord('c')	10 >>> chr(99)
5 99	11 'c'

- a. What is the type of the argument passed to the built-in function `ord` on lines 0, 2, & 4? _____
- b. If we called `len(object)` on the arguments passed to `ord`, what would the return value be in all cases? _____
- c. What does line 0 and line 7 have in common? (also: lines 3 & 8, lines 5 & 10) _____
- d. What does line 1 and 6 have in common? (also: lines 2 & 9, lines 4 & 11)? _____
- e. If we added a line 12, `chr(ord('d'))` what might the output be? _____
- f. What does the built-in `ord(aChar)` function do? _____
- g. What does the built-in `chr(anInt)` function do? _____
- h. What might the output of `ord('A')` be? _____
- i. If the output of `ord('A')` is 65, what might the output of `ord('B')` be? _____
- j. If the output of `ord('A')` is 65, how might we predict the output of `ord('Z')`? _____
-

Application Questions: Use the Python Interpreter to check your work

1. Write a function, `safeConvertToFloat(strNum)` that takes a string `strNum`, and checks to see if the string can be converted to a floating point number. If it can, the function returns the value as a float, if it cannot, it prints an explanatory message: "String is not a float!"

2. Write a function, `shiftLetter(letter, shift)` that takes a single letter (as a string) and shifts it over the alphabet by `shift` letters. Check your answer on page 81 of the textbook. Be sure to handle the case of lower and uppercase letters, as well as handle what happens when you want to shift the letter 'z'!
