2.6. Type conversion functions

Sometimes it is necessary to convert values from one type to another. Python provides a few simple functions that will allow us to do that. The functions `int`, `float` and `str` will (attempt to) convert their arguments into types `int`, `float` and `str` respectively. We call these **type conversion** functions.

The `int` function can take a floating point number or a string, and turn it into an int. For floating point numbers, it *discards* the decimal portion of the number - a process we call *truncation towards zero* on the number line. Let us see this in action:

Save & Run 4/17/2021, 12:21:39 PM - 2 of 2 Show in CodeLens

```
print(3.14, int(3.14))
print(3.9999, int(3.9999))       # This doesn't round to the closest int!
print(3.0, int(3.0))
print(-3.999, int(-3.999))        # Note that the result is closer to zero

print("2345", int("2345"))        # parse a string to produce an int
print(17, int(17))                # int even works on integers
print(int("23bottles"))

```

```
3.14 3
3.9999 3
3.0 3
-3.999 -3
2345 2345
17 17
```

Activity: 1 -- ActiveCode (ac2_6_1)

Error

```
ValueError: invalid literal for int() with base 10: '23bottles' on line 8
```

Description

A ValueError most often occurs when you pass a parameter to a function and the function is expecting one type and you pass another.

To Fix

The error message gives you a pretty good hint about the name of the function as well as the value that is incorrect. Look at the error message closely and then trace back to the variable containing the problematic value.

The last case shows that a string has to be a syntactically legal number, otherwise you'll get one of those pesky runtime errors. Modify the example by deleting the `bottles` and rerun the program. You should see the integer `23`.

The type converter `float` can turn an integer, a float, or a syntactically legal string into a float.

Save & Run 4/17/2021, 12:21:53 PM - 2 of 2 Show in CodeLens

```
print(float("123.45"))
print(type(float("123.45")))

```

```
123.45
<class 'float'>
```

Activity: 2 -- ActiveCode (ac2_6_2)

The type converter `str` turns its argument into a string. Remember that when we print a string, the quotes are removed in the output window. However, if we print the type, we can see that it is definitely `str`.

Save & Run 4/17/2021, 12:21:56 PM - 2 of 2 Show in CodeLens

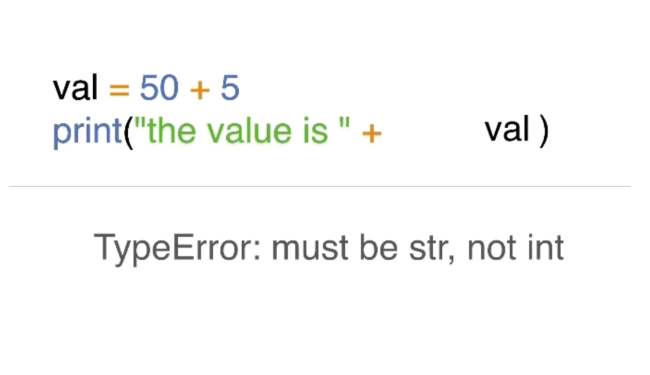
```
print(str(17))
print(str(123.45))
print(type(str(123.45)))

```

```
17
123.45
<class 'str'>
```

Activity: 3 -- ActiveCode (ac2_6_3)

One common operation where you might need to do a type conversion is when you are concatenating several strings together but want to include a numeric value as part of the final string. Because we can't concatenate a string with an integer or floating point number, we will often have to convert numbers to strings before concatenating them.



```
val = 50 + 5
print("the value is " +         val )
```

```
TypeError: must be str, not int
```

Check your understanding

data-6-1: What value is printed when the following statement executes?

```
print(int(53.785))
```

○ A. Nothing is printed. It generates a runtime error.

● B. 53

○ C. 54

○ D. 53.785

Check me Compare me

✓ The int function truncates all values after the decimal and prints the integer value.

Activity: 4 -- Multiple Choice (question2_6_1)

You have attempted 5 of 4 activities on this page

✓ Completed. Well Done!

2.5. Data Types">

a Types">

2.7. Variables">

| Back to top