**Type Casting**

1. Fill in the table below with the returned value of calling the given function on the given value. Circle the items in the table that you feel unsure of to double check later. Watch out, some of these functions may cause errors!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| value | type(value) | str(value) | int(value) | float(value) |
| "dragons" | str | dragons | Error | Error |
| 1978 | int | 1978 | 1978 | 1978.0 |
| "number 2" | str | number 2 | Error | Error |
| -4.6789 | float | -4.6789 | -4 | -4.6789 |
| 45 | int | 45 | 45 | 45.0 |
| "-12" | str | -12 | -12 | -12.0 |
| 3.14159 | float | 3.14159 | 3 | 3.14159 |

1. Fill in the blanks in the code below so that it will execute without errors.

age1 = input("how old are you? ")

age2 = input("how old is your best friend? ")

age\_sum = int(age1) + int(age2)

print("Together, you are: " + str(age\_sum))

1. Whenever we use the input( ) function, what type is the value that it gives back to us? Does it remain the same whether the user types “strawberries” or “7” or “93.2”?

It always gives a string value.

1. Finish the following code that computes the mean of three grades.

lab1 = 10

lab2 = 8.73

lab3 = 9.82

mean = (lab1+lab2+lab3)/3

print("Your mean so far: " + str(mean))

rounded = int(mean)

print("Your mean rounded down to the nearest int: " + str(rounded))

Stop! Don't proceed until your group has answered questions 1 and 2 on the conceptual questions sheet.

**Errors & Scope**

1. Which of the following code snippets will run without throwing errors? If they produce an error, what error will they produce? If they do not produce an error, what will the output be?

|  |  |
| --- | --- |
| Code | No errors/Error (and why?) |
| x = 1  y = 7  z = x \* y  print(z) | No errors. The types of x and y agree with the operation in the definition of z. |
| name = "alex"  age = 43  print(name + ": " + age) | Error. There is a type mismatch between the str name and the int of age |
| a = a + 1  b = a + 1  print(str(b) + " is more than " + str(a)) | There is a NameError here: a is not initially defined, so its self reference does not make sense. |
| five = 5  one = 1  print(one / (five - 5.0)) | Error: this results in division by 0. |
| height = 5.5  age = 10  print("feet per year: " + str(height / age)) | No errors. The / operation works between a float and an int, and converting it to a string means that it can be printed with the string. |

**Using Functions**

1. To call a function, what do you need to know?

You need to know its name, and the inputs.

1. If a function that you are calling **does not** have a return value (for instance, print()), what is the general format for calling it?

Just writing out the name of the function and its inputs suffices.

1. If a function that you are calling **does** have a return value (for instance, input()), what is the general format for calling it?

Store the value of the function in a variable, by defining a variable as an instance of that function.