**Review - Data Types in Python**

Every **variable** stores a **value**. For example:

x = 9

This means the variable x stores a value of 9. The value 9 is an **integer** **literal** and x is an **integer variable**. The integer data type is only one of the possible Python data types that we can store in a variable. All programming languages have various **data types** they accept. The most common data types are:

**Integer**: positive or negative whole numbers

**Floating point number**: positive or negative rational (decimal) number

**String**: set of characters, delimited by quotation marks.

**Boolean**: a true or false value.

A variable can be assigned a value of any data type in an **assignment statement**. Some examples of assignment statements:

age = 19 # integer

distance = 15.6 # float

password = “happy99” # string

passwordAccepted = True # Boolean

Programmers must be aware that computers treat different data types differently. For one thing, each data type requires a different amount of space in the computer’s memory. For another thing, all data is stored as binary numbers in the computer. So when the computer sees a value, say “1110 1101” it needs to know: is this a letter? Is it a whole number? Is it a decimal number? Knowing the answer to these questions is important, because different data types can be interpreted differently. For example, what does a + b mean?

Addition of integers:

a = 1

b = 2

print (a+b)

Addition of strings:

a = "hello"

b = "goodbye"

print (a+b)

**Converting Between Data Types**

You can **convert** between data types:

a = "3.5" # a is a string

x = float(a) # convert to a float

y = int(x) # convert to an integer

print(x, y)

We have used this already, to convert string input into integers:

z = int (input (“Enter a number”)) # converts what the user enters into an integer

print (z)

Some things can’t be converted:

a = "hello" # a is a string

x = int(a) # OOPS! can’t convert “hello” into an integer

print(x)

Running the above code will produce this error:

>>>builtins.ValueError: invalid literal for int() with base 10: 'hello'

This also won’t work:

password = "hello" + 99 # can’t add a string and a number

print (password)

But this will:

password = "hello" + str(99) # convert the number to string first!

print (password)

This can be useful when building up a long string that includes numbers:

cats = 3

dogs = 1

sentence = “I have “ + str(cats) + “cats and “ + str(dogs) + “ dog in my house.”

Putting strings together is called “concatenation”.

**Keywords: *Data type, integer, float, string, variable, literal, concatenation.***