DAY 2 24 October 2023

**Introduction to Variables**

Variables are temporary storage space in a computer’s memory.

* When the variable changes the whole state of the programming changes.
* A variable acts as a container to hold a different number of data and values.

Variables can be in three declarations:

* Variable creation (declaration)
* Variable assignment (assigning)
* Variable changed (Execution)

Naming of the variables contain lower case letters (A-Z, a-z), number or the underscore character. You can not start by a number or put spaces when naming your variables.

A screenshot of a computer

Description automatically generated

**Using Variables**

When using the variables you have to assign them to a data type like a string or an integer. In Python a variable is assigned automatically to an appropriate data type e.g a variable can be assigned to a string data type if an input or value that is given contains letters or words.

**Casting**

Casting can be done in two ways **Implicitly** & **Explicitly**.

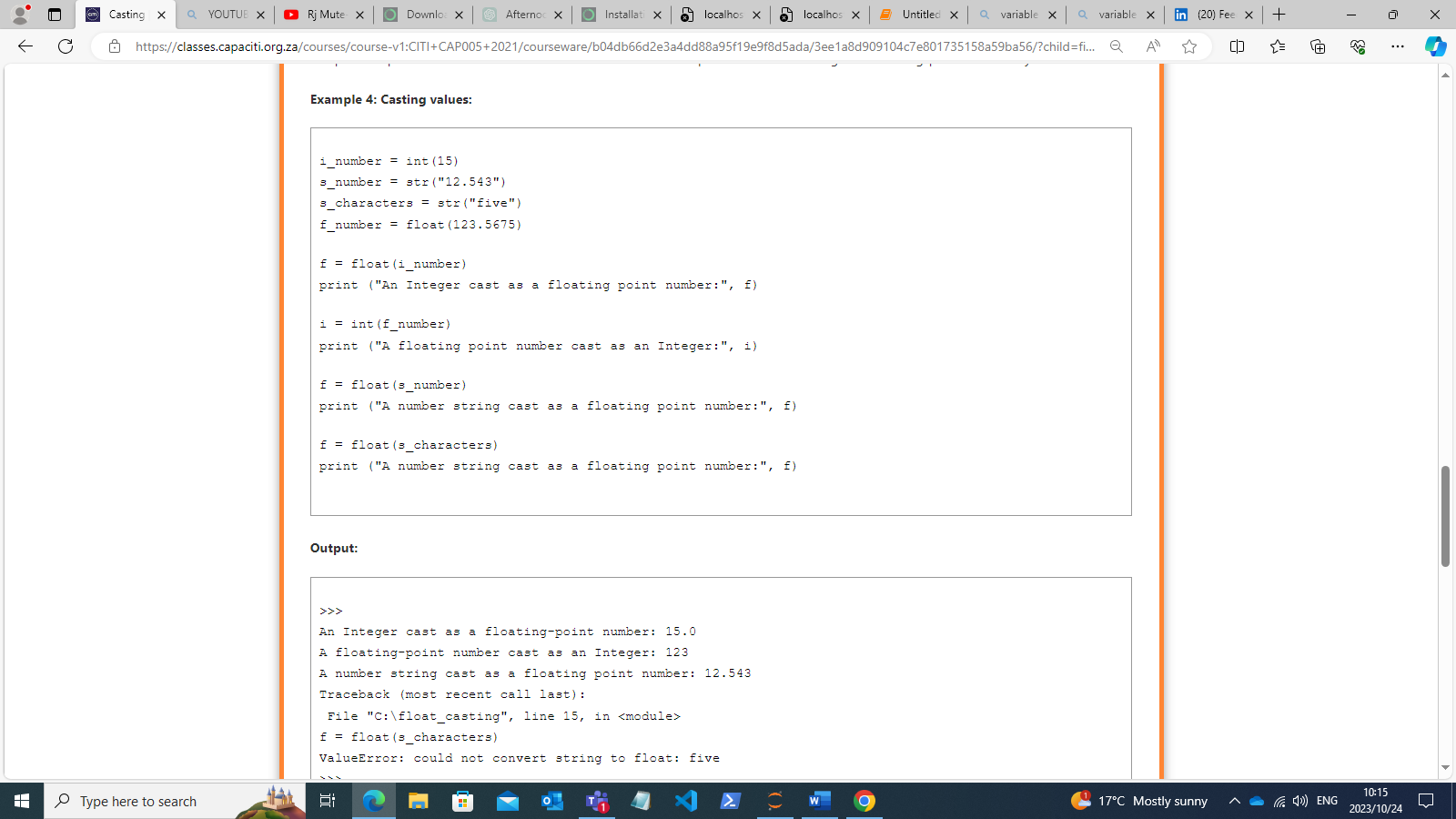
Implicitly: The compiler automatically casts a value from one data type to another when assured that there will be no data loss.

Explicitly: A value cannot be automat cast from one data data type to another if it will result in data loss.

**NOTE!!!** No errors occur when converting from an integer to a float. This is because an integer can be cast to float data implicitly because there will be no data loss.

So a programmer needs to know that an integer can be cast to a float data type but a float data type cannot be cast to an integer data type because there will be data loss. This cannot happen because the floating-point value does not get rounded off to the nearest whole number when implicitly converting to an integer.

For example.



DAY 3

**DATA TYPES**

Data types are fundamental concept in programming and computer science. They define what kind of data a particular variable can hold and how that data can be manipulated.

The Data types that I will be covering:

1. **Integer (int):** Represents whole numbers, both positive and negative, without any fractional part. Examples include 1, -5, and 0.
2. **Float (float):** Represents numbers with decimal points. Examples include 3.14, -0.5, and 2.0.
3. **String (str):** Represents sequences of characters, such as text or symbols. For example, "Hello, World!" is a string.
4. **Boolean (bool):** Represents binary values, typically either **True** or **False**. Booleans are often used for making logical decisions in code.
5. **Complex numbers:**  Complex numbers represent numbers as a pair of double-precision numbers.

**Integers**

Integers are always whole numbers. Integers include negative and positive numbers. The only factor that determines the range of an integer variable is the amount of memory a machine has available.

The four main operators used when doing calculations, are ‘**+**’, ‘**-**’, ‘**\***’, and ‘**/**’.

**Booleans**

Only two possible values that a Boolean variable can have, **True(1)** or **False(0)**.

Three logical operators used to test conditions are:

* **AND** operator
* **OR** operator
* **NOT** operator

The **AND** gate:

Here the inputs are **on** in order the gate to have an output to be **on**.

The **NOT** gate:

The operator coverts **true** to false and **false** to true.

x

y z

|  |  |  |
| --- | --- | --- |
| 0 | 0 | **1** |
| 0 | 1 | **1** |
| 1 | 0 | **1** |
| 1 | 1 | **0** |

z = str(not bool(x and y))

 bool(x and y): The value is then converted from the number form (**0** or **1**) to the word form of Boolean (True or False). not bool(x and y) converts **True** to **False**, or False to True. z is then converted to a string.

**Floating Point Numbers**

The float data type can be called as a function with a zero or 1 argument of any data type.

* The **R%.2f** formats the number into 2 decimal places for example 6.54354 is 6.54.
* The **R%f** formats the number and give out the last digit which is zero e.g 6.675 will be 6.6750.
* The **R%+.2f** combines the 2 format’s e.g “R%+.2f” % 0.5 then the output will be R+0.50

The f indicates the number to be formatted is a float.

The + sign indicated that the changed amount must be signed.

*The following table shows tokens that can be used to format a string:*

*A screenshot of a computer

Description automatically generated*

**Complex numbers**

Complex numbers are two numbers contained in a single variable.

Complex numbers are used in Python to combine two numbers into one manageable number.

Imaginary numbers are real multiples of the imaginary unit, written with a suffix of j (J). The imaginary part is the square root of -1. Python has built-in support for complex numbers. The latter notation is written as follows: 4+8j.

**Strings**

Strings are a sequence of Unicode characters which form a single manageable string.

The ‘+=‘ operator adds values to an existing variable.

The end of line escape sequence (\) can also help to make code more readable.

**LAMBDA Expressions**

Lambda expression also known as a lambda function is a compact way to define small, anonymous(unnamed) functions in programming.

In Python, a lambda expression is defined using the “lambda” keyword, followed by a list of arguments, a colon, and an expression.

The syntax is: *lambda arguments: expression*

Here is an example of lambda expression that adds two numbers:

add = lambda x, y: x + y

result = add(3, 5)

print(result) # Output: 8