Lecture 1 Notes

4th of October

Lecture 1: Basic Syntax

In this part we will look at the basic syntax of the language, the basics that you will use throughout the whole course.

Printing on the screen

```python
print("Hello World")
```

This code will print show on the screen the message : "Hello World". `print` is an example of a function, we will be covering what functions are later on, but there are some functions

Variables and types

Variables are used to store information to be referenced and manipulated in a computer program. They also provide a way of labeling data with a descriptive name, so our programs can be understood more clearly by the reader and ourselves. It is helpful to think of variables as containers that hold information. Their sole purpose is to label and store data in memory. This data can then be used throughout your program.

```python
a = 1
b = "Hello"
c = 1.2
d = 'c'
e = False
```

This is how you **define** variables in Python. Python is a **dynamically-typed** language, which means that the **types** of variables don't have to be specified

when defining them, and that a variable can change its type and value after being defined. So what are types? Taking the example above, variable `a` is set to `1`, the type of `1` is and **integer**. There are many different types in computer science but for now we will cover:

- Integers, `a`
- Strings, `b`
- Floats, `c`
- Char, `d`
- Boolean, `e`

```
a = "Hello World"
print(a)
```

In this example we have defined a string variable `a` and set its value to `"Hello World"` and then **parsed** that variable into the `print` function, this will print the same thing as our previous example.

```
a = "Hello World"
print(a)
a = 1
print(a)
```

This is an example showing that Python is dynamically-typed language, as you can see we are **re-defining** variable `a` from a `string` to an `integer`. This will print :

```
Hello World
1
```

We can also convert variables to different types (not always), for example:

```
a = 1
a = str(a)
a = int(a)
```

In this example we use the function `str` to convert the variable a from an integer to a string, then we use the function `int` to convert it back to an integer.

Working with integers and floats

Lets start with the most basic operations:

```python
a = 1
b = 5
c = 2
result = (a+b)/c
print(result)
```

As you can see in the example we have defined three integer variables `a`, `b` and `c`, then we have defined a `result` variable which contains the sum of `a` and `b`, divided by c. The `result` variable is of type `float`, this happens because of the `/` (division) operator which in Python returns a float number. This means that the value of `result` is `2.0`. If you wanted to get an integer simply use `//` instead of `/`, fortunately in Python integers and floats are treated similarly, which means that we can add a float and an integer without having to convert any of them.

Operator	Action
+	Addition
-	Substraction
	Multiplication
/	Division
//	Floor Division
	Exponent
%	Modulus (remainder)

These are the basic operators, with the introduction of **libraries** which we will explain further on the arithmetic you can do will expand.

These operators can also be used when we assign a value to a variable. For example:

```python
x = 2
x = x*2
```

This declares a variable `x` setting its value to 2, then updates the value to its current value * 2. Which can also be written as:

```python
x = 2
x *= 2
```

The operator `*=` is a combination of `*` and the equal sign we use to assign a value to a variable. The operator, in this case `*`, can be replaced by any of the operators showed before E.g. : `//=`, `+=`, `-=`...

Input and output

If we want to make our program interactive we need to have some sort of input from the user. In this section we will cover the most basic way of getting an input, through the command line

```
a = input("Hello user! Write whatever you like : ")
print("Here's what you wrote : " + a)
```

The `input` function prints on the screen the argument parsed (we will cover arguments in future lessons) and will be expecting an input from the user, the input will be submitted when the enter key is pressed. The input of the user will be stored into the variable `a` as a string. We will then use the `print` function to show the user whan the input was. The `+` sign is also used for strings, this is called **string-concatenation**, which is when we simply add two strings together into one string.

Challenge

With the knowledge you got from this lecture your task is to create a simple program which asks the user to input **two integers** and then it takes those two numbers and executes at least **3 different operations** on them and outputs the result to the user.V

Hint : keep in mind the different types! What type is an input stored as?