**Section 5: Python Statements**

**09.02.**

**34. If Elif and Else Statements in Python**

In Python we can use as many elif statements as wanted

But starting with Python 3.10 there was introduced the structural pattern matching

This would be the equivalent to case statements

match cond:

    case val\_one:

        …

    case val\_two:

        …

case \_:

…

**35. For Loops in Python**

We can use \_ in a for loop if we won’t use the current variable in the iterator

for \_ in ‘Hello’:

print(‘My text’)

This is just a convention because \_ is a valid variable in Python

In the previous case we can use print(\_) to print each character

Tuple unpacking can be used in the following way:

l = [(1,2), (3,4), (5,6)]

For x in l: For (a, b) in l: For a, b in l:

print(x) print(a, b) print(a, b)

Actually, the parenthesis for tuple unpacking aren’t needed

By default when we are iterating over a dictionary, we get the keys

**36. While Loops in Python**

While condition:

…

Else:

…

We can use an else branch that will be executed if / when the while condition is not true

There are some additional statements:

Break – will exit the closest enclosing loop

Continue – will go to the next iteration in the closest enclosing loop

Pass – won’t do nothing and can be used in function too

If we have a function with the first statement ‘pass’, the following statements will be executed

**10.02.**

**37. Useful Operators in Python**

range(first\_el ,last\_el–1, step) is a generator that don’t store in memory the entire sequence

list(range(first\_el, last\_el-1, step)) can be used as a list instead

enumerate can be used with every iterable

zip is a generator and can be used with as many iterables as wanted

if the size of iterables is different, the results will be short as the shortest

In the previous scenario an error won’t be thrown

We can cast the zip with list or dict, but for dict just 2 iterables are allowed

Min(list) and max(list) for ordinary lists

If we are dealing with numpy array we can use min(np\_array) or np\_array.min()

Input function input(prompt\_to\_display)

The returned type of input function is str

**38. List Comprehensions in Python**

The basic syntax is [el for el in list\_name]

Including an if: [el for el in list\_name if condition]

Including an if else [el\_one if condition else el\_two for el in list\_name]

Nested list comprehensions: [el\_one\*el\_two for el\_one in list\_one for el\_two in list\_two]

In the scenario above, then list\_two is iterated as the inside one