7.79 - Leaflet - Lists in Python

July 23, 2018

0.1 Lists

Lists allow you to directly save multiple entries at once. For example, here we have a list of 4 students:

```
In [1]: students = ["Max", "Monica", "Eric", "Paula"]

        last_student = students.pop()
        print(last_student)
        print(students)
```

```
Paula
['Max', 'Monica', 'Eric']
```

With the + - operator you can link 2 lists with each other!

```
In [2]: students = students + ["ABCDEF"]
        print(students)
```

```
['Max', 'Monica', 'Eric', 'ABCDEF']
```

The `del` command removes an entry from a list at a specific index. Indexes of Lists start with 0.

```
In [4]: students = ["Max", "Monica", "Eric", "Paula", "ABCDEF"]
        del students[3]
        print(students)
```

```
['Max', 'Monica', 'Eric', 'ABCDEF']
```

The `.remove()` - function removes an entry by value. In other words, the entry "Monica" is removed from the list here.

```
In [5]: students = ["Max", "Monica", "Eric", "Paula", "ABCDEF"]
        students.remove("Monica")
        print(students)
```

```
['Max', 'Eric', 'Paula', 'ABCDEF']
```

0.2 List Comprehensions

With the help of List Comprehensions you can easily convert a list into another list:

```
In [6]: xs = [1, 2, 3, 4, 5, 6, 7, 8]

        ys = [x * x for x in xs]
        # ys = []
        # for x in xs:
        #     ys.append(x * x)

        print(xs)
        print(ys)
```

```
[1, 2, 3, 4, 5, 6, 7, 8]
[1, 4, 9, 16, 25, 36, 49, 64]
```

0.2.1 List Comprehensions can do much more!

```
In [7]: students = ["Max", "Monica", "Eric", "Paula"]

        lengths = [len(student) for student in students]

        #lengths = []
        #for student in students:
        #    lengths.append(len(student))

        print(lengths)
```

```
[3, 6, 4, 5]
```

0.2.2 Useful also for drawing graphics!

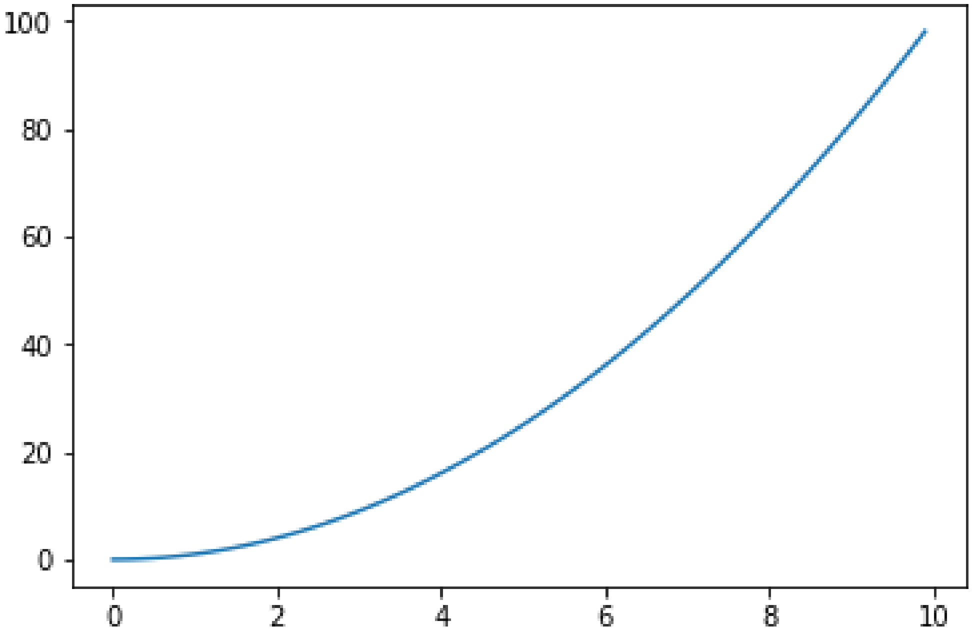
```
In [8]: %matplotlib inline
        import matplotlib.pyplot as plt
```

```
In [9]: xs = [x / 10 for x in range(0, 100)]
        ys = [x * x for x in xs]

        print(len(xs))
        print(len(ys))

        plt.plot(xs, ys)
        plt.show()
```

```
100
100
```



0.3 Nesting lists

In Python it is allowed to nest lists. This allows us to model a matrix, for example:

```
In [10]: liste = [
             ["New York", "Chicago", "Los Angeles"],
             ["Budapest", "Pécs", "Sopron"]
         ]

In [11]: liste[0][0]

Out[11]: 'New York'
```