

Data structures

Many useful data structures are implemented in Python. The ones we will use most often are `lists` and `dicts`. Lists are pretty much self-explanatory, they contain a set of arbitrary things, even other lists, or even themselves:

In [20]:

```
a = 7

testlist = []
print(type(testlist))

testlist.append("apple")
testlist.append(15)
testlist.append(a)
testlist.append(testlist)

print(testlist)
print(testlist[-2])
```

```
<class 'list'>
['apple', 15, 7, [...]]
7
```

Square brackets access a certain element in a list. You can also access list elements using negative numbers, in which case they'll start referencing elements starting from the last element in a list. You can also use slicings, choosing a range of elements from a list. We will see how these work after getting to know one of the most beautiful concepts in Python; **list comprehensions**:

In [25]:

```
import math as m

n = 15

squares = [x**2 for x in range(n)]

list_of_lists = [[x**2, m.sin(y)] for x in range(3) for y in range(3)]

print(squares)
print(list_of_lists)

print("Slicing: ", squares[1:3])
print("Negative elements: ", squares[-1])
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196]
[[0, 0.0], [0, 0.8414709848078965], [0, 0.9092974268256817], [1, 0.0],
[1, 0.8414709848078965], [1, 0.9092974268256817], [4, 0.0], [4, 0.8414
709848078965], [4, 0.9092974268256817]]
Slicing:  [1, 4]
Negative elements:  196
```

format:

In [7]:

```
d = {"first": "some text", \
     "second": a, \
     "third": squares}
```

```
print(d["first"])
print(d["second"])
print(d["third"])
```

some text

7

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196]

You can print all available keys and values:

In [10]:

```
print(d.keys())
print(d.values())
```

dict_keys(['first', 'second', 'third'])

dict_values(['some text', 7, [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196]])