**Create a list**

As opposed to int, bool etc., a list is a **compound data type**; you can group values together:

a = "is"

b = "nice"

my\_list = ["my", "list", a, b]

# Create list with different types

A list can contain any Python type.

# List of lists

house = [["hallway", hall],

["kitchen", kit],

["living room", liv],

["bedroom", bed],

["bathroom", bath]]

# Subset and conquer

Subsetting Python lists is a piece of cake. Take the code sample below, which creates a list x and then selects "b" from it. Remember that this is the second element, so it has index 1. You can also use negative indexing.

x = ["a", "b", "c", "d"]

x[1]

x[-3] # same result!

# Slicing and dicing

Selecting single values from a list is just one part of the story. It's also possible to slice your list, which means selecting multiple elements from your list. Use the following syntax:

my\_list[start:end]

The start index will be included, while the end index is not.

The code sample below shows an example. A list with "b" and "c", corresponding to indexes 1 and 2, are selected from a list x:

x = ["a", "b", "c", "d"]

x[1:3]

The elements with index 1 and 2 are included, while the element with index 3 is not.

# Slicing and dicing (2)

In the video, Filip first discussed the syntax where you specify both where to begin and end the slice of your list:

my\_list[begin:end]

However, it's also possible not to specify these indexes. If you don't specify the begin index, Python figures out that you want to start your slice at the beginning of your list. If you don't specify the end index, the slice will go all the way to the last element of your list. To experiment with this, try the following commands in the IPython Shell:

x = ["a", "b", "c", "d"]

x[:2]

x[2:]

x[:]

# Subsetting lists of lists

You saw before that a Python list can contain practically anything; even other lists! To subset lists of lists, you can use the same technique as before: square brackets.

x = [["a", "b", "c"],

["d", "e", "f"],

["g", "h", "i"]]

x[2][0]

x[2][:2]

x[2] results in a list, that you can subset again by adding additional square brackets.

# Replace list elements

Replacing list elements is pretty easy. Simply subset the list and assign new values to the subset. You can select single elements or you can change entire list slices at once.

x = ["a", "b", "c", "d"]

x[1] = "r"

x[2:] = ["s", "t"]

# Extend a list

If you can change elements in a list, you sure want to be able to add elements to it, right? You can use the + operator:

x = ["a", "b", "c", "d"]

y = x + ["e", "f"]

# Delete list elements

Finally, you can also remove elements from your list. You can do this with the del statement:

x = ["a", "b", "c", "d"]

del(x[1])

Pay attention here: as soon as you remove an element from a list, the indexes of the elements that come after the deleted element all change!