PYTHON SEMINAR 2020

JENS HAHN

THEORETICAL BIOPHYSICS

TODAY

I Objects recap

II Data structures

III Flow control

I. ASSIGNMENT I

Tasks

- *Fetch* from *upstream* and *merge* - get course material
- Replace *None* by *True/False* - work on assignment
- *Commit* file and *push* - save progress in repo

I. OBJECTS RECAP

- Duck typing - Python choses the *object type* automatically!
- Object - Every *object* has *attributes* and *methods* (functions)
- Attention!! - *objects* can be **mutable** or **immutable**

II. DATA STRUCTURES - LOGIC

Boolean (*bool*)

- ***`True`*/*`False`***

Answer of comparisons

`<` `>` `<=` `>=` `==` `!=` `is` `in`

Logic operators: ***`not`*** ***`or`*** ***`and`***

None (*NoneType*)

- ***`None`***

II. DATA STRUCTURES - NUMBERS

Integer (*int*)	:	`1   2`
Real numbers (*float*)	:	`1.3   1e-6`
Complex (complex)	:	`4+j3`

Mathematical operators:

+	addition
-	subtraction
*	multiplication
/	division
//	floor division
%	modulo
**	exponent

II. DATA STRUCTURES

Strings (*str*)

```
’this is a string’

”this is also a string”

’’’this is a string as well’’’
```

Bytes (*bytes*)

Mutable bytes (*bytearray*)

Encodings:

ASCII	- 128 chars
ISO8859	- 256 chars
utf-8	- 2,164,864 chars

II. DATA STRUCTURES - CONTAINERS

Lists (*list*)	:	`[1, ”hallo”, 4+4j, True]`
Tuples (*tuple*)	:	`(1, ”hallo”, 4+4j, True)`
Sets (*set*)	:	`{1, ”hallo”, 4+4j, True}`
Immutable sets (*frozenset*)		

II. DATA STRUCTURES - SLICES

```
my_list = [1,2,3,4,5,6,7,8,9,10]
```

`my_list[0]`	-	**first** element of list	-	`1`
`my_list[8:-1]`	-	from **9th** element to **last**	-	`[8,9]`
`my_list[0:10:2]`	-	**1st** to **11th** element, **2** steps	-	`[1,3,5,7,9]`

```
[start:end:step]
```

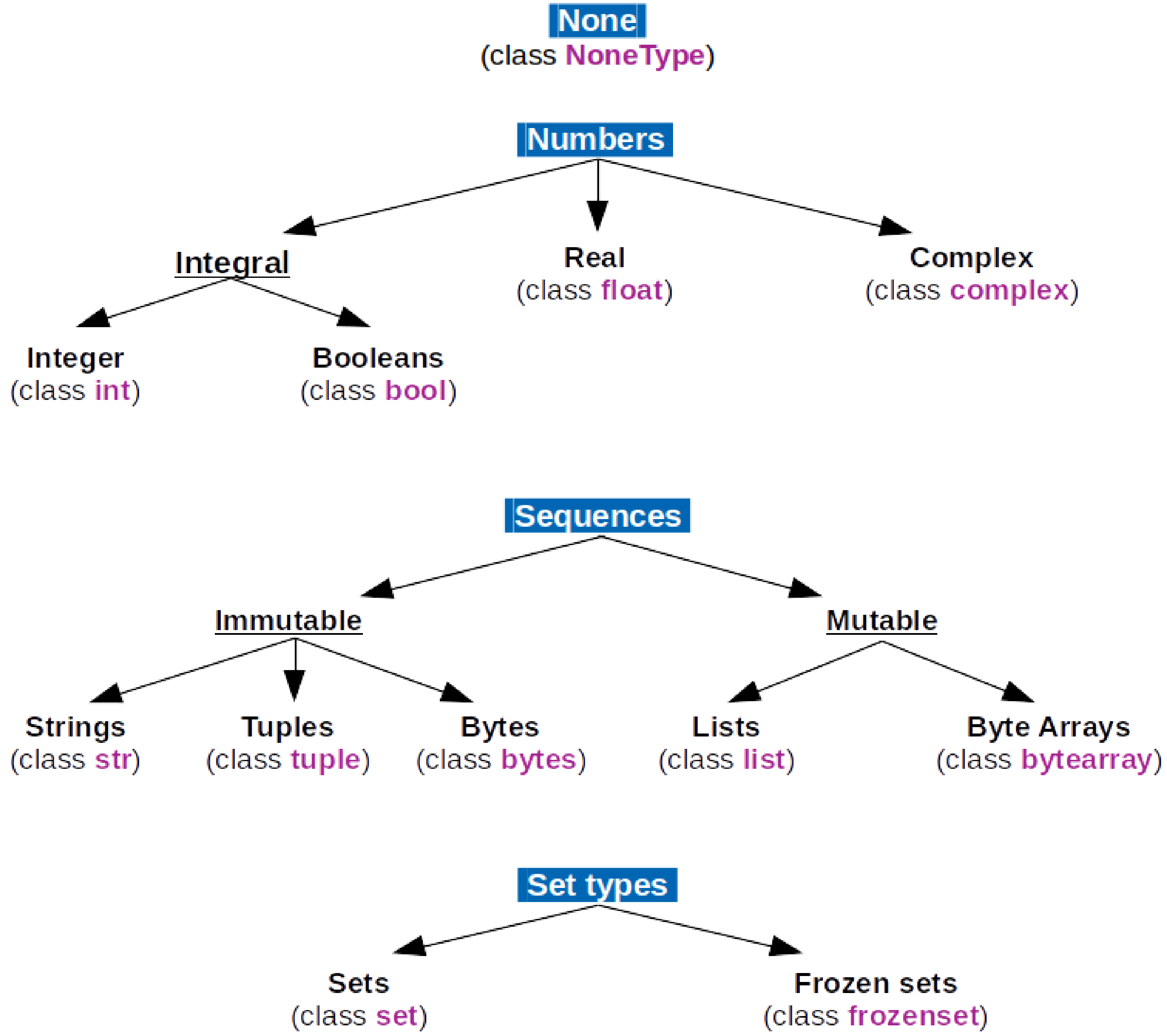
II. DATA STRUCTURES - DICTIONARIES

Dictionaries (*dict*)

```
room_numbers = {‘Jens’: 518, ‘Katja’: 518, ‘Edda’: 516}

room_numbers[‘Edda’]

     516
```



None
(class NoneType)

Numbers

Integral

Integer
(class int)

Booleans
(class bool)

Real
(class float)

Complex
(class complex)

Sequences

Immutable

Strings
(class str)

Tuples
(class tuple)

Bytes
(class bytes)

Mutable

Lists
(class list)

Byte Arrays
(class bytearray)

Set types

Sets
(class set)

Frozen sets
(class frozenset)

"Python 3. The standard type hierarchy.png"
by Максим Пе is licensed under CC BY-SA 4.0

III. FLOW CONTROL - LOOPS

for loop

```
for list_element in my_list:
----print(list_element)
```

while loop

```
index = 0
while index <= 10:
----print(my_list[index])
----index += 1
```

III. FLOW CONTROL - CONDITIONAL

if / else / elif

```
if type(my_list) == list:
----print(‘this is a list’)
elif type(my_list) == set:
----print(‘this is a set’)
else:
----print(‘I don’t even know what it is’)
```

IV. ASSIGNMENT - ROMAN NUMERALS

Roman numerals

- I, II, III, IV, V, VI, VII, VIII, IX, X
- Addition if smaller numeral is right
- Subtraction if smaller numeral is left
- e.g. MCMXCV - 1995

Arabic numerals

- 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

I	–	1	V	–	5
X	–	10	L	–	50
C	–	100	D	–	500
M	–	1000			

III. FURTHER READING

Python data types

- Programiz – Python data types

  https://www.programiz.com/python-programming/variables-datatypes

- Unicode – character table

  https://unicode-table.com/en/#control-character

Python control flow

- Python documentation

  https://docs.python.org/3/tutorial/controlflow.html

- Roman and Arabic numerals converter Python

  https://stackoverflow.com/questions/28777219/basic-program-to-convert-integer-to-roman-numerals