



PyQB

Monga

Dictionaries

Sets

Comprehensions

Programming in Python¹

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia

`mattia.monga@unimi.it`

Academic year 2024/25, I semester



PyQB

Monga

Dictionaries

Sets

Comprehensions

Lecture VII: Other Composite Objects



State of the homework

	Students	Solved by
One Triangle	16	8

- 1 Click on the link and accept the assignment: this will create your own git repository on GitHub with the homework
- 2 Clone the repository on your machine (the easiest way is to use the GitHub Desktop app)
- 3 Work on the solution in the file `exercise.py`
- 4 Commit your work (again with GitHub Desktop it is easy)
- 5 Push (again with GitHub Desktop it is easy) the solution on GitHub so I can comment on it
- 6 Read my comments
- 7 If the solution can be improved, go back to step 3

PyQB

Monga

Dictionaries

Sets

Comprehensions



Dictionaries

A composite type `dict` that implements a `mapping` between immutable `keys` and `values`.

```
d = {'key': 'foo', 3: 'bar'}
```

```
print(d['key']) # 'foo'  
print(d[3])     # 'bar'  
print(d[2])     # error!
```

Notation is similar to lists/tuples, but `dicts` are not sequences indexed by numbers, you must use only the existing keys (`d.keys()`).

```
if x in d.keys():  
    print(d[x])
```

A sequence of values can be obtained with `d.values`. A sequence of 2-tuples (key, value) with `d.items()`.

PyQB

Monga

Dictionaries

Sets

Comprehensions



A `set` is a composite object with no duplicate (non mutable) elements. Common set operations are possible.

- Set literals: `{1,2,3}` `set()`
- `{1,2,3}.union({3,5,6})`
`{1,2,3}.intersection({3,5,6})`



Comprehensions

Comprehensions are a concise way to create lists, sets, maps... It resembles the mathematical notation used for sets

$$A = \{a^2 | a \in \mathbb{N}\}.$$

```
squares = [x**2 for x in range(10)]
```

equivalent to:

```
squares = []  
for x in range(10):  
    squares.append(x**2)
```

filtering is possible

```
odds = [x for x in range(100) if x % 2 != 0]
```

with a set

```
s = {x for x in range(50+1) if x % 5 == 0}
```

with a dict

```
d = {x: x**2 for x in range(10)}
```

PyQB

Monga

Dictionaries

Sets

Comprehensions