Type hints usage in Python projects

Software Composition Seminar

Student: Jovana Mićić
Supervisor: Nevena Lazarević

15. May 2018.

Example

```python
def fetch(artist: str, song: str, album: str) -> str:
    Helpers = hlp.HELPERS
    lyrics = None
    for helper in helpers:
        try:
            lyrics = helper.fetch(artist, song, album)
        except requests.ConnectionError as error:
            log.warning('Connection error: %s', error)
        if lyrics:
            break
    return lyrics
```
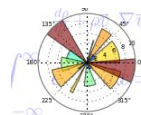
Who uses type hints?

- 21 Python libraries and projects analysed

Who uses type hints?

	Name	Num of lines		Name	Num of lines
1	sckit-learn	215122	**12**	Tale	23698
2	matplotlib	213210	**13**	pendulum	22241
3	networkx	116963	**14**	GitPython	21075
4	pytorch	89896	**15**	urllib3	20726
5	scikit-image	81402	**16**	requests	9602
6	mypy	74699	**17**	httpie	6370
7	pytype	47473	**18**	pydantic	5529
8	scrapy	35816	**19**	MonkeyType	4423
9	neupy	34980	**20**	pyquery	2848
10	cherrypy	29506	**21**	httpbin	2317
11	paramiko	25031			

How many of you know for type hints in Python?

Let's see the results...

Usage in projects and libraries
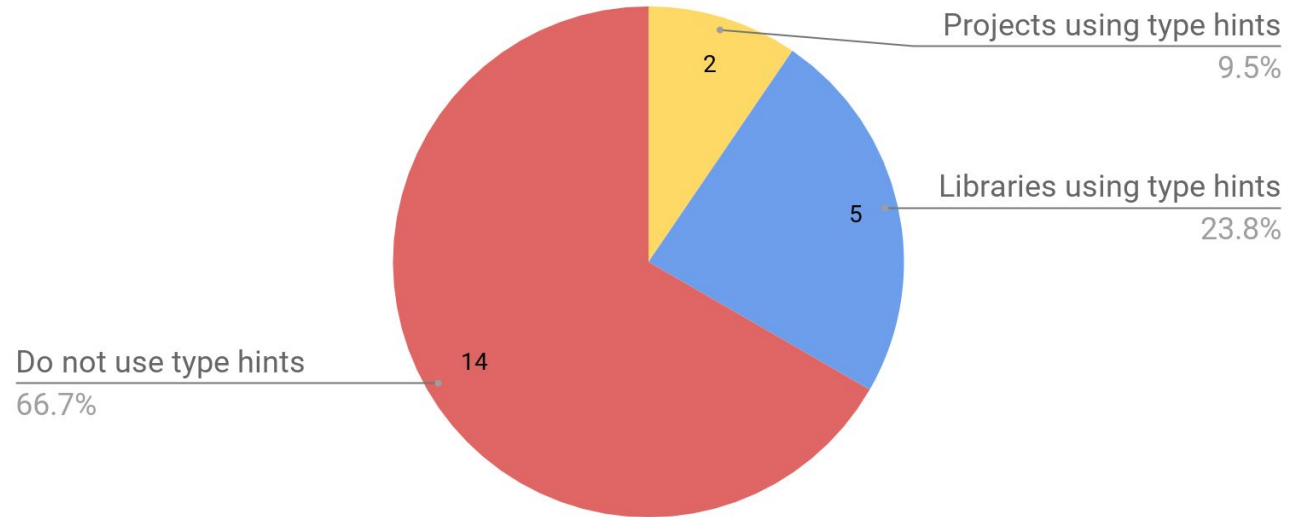
Number of projects using and not using type hints

Projects using type hints
9.5%
2

Libraries using type hints
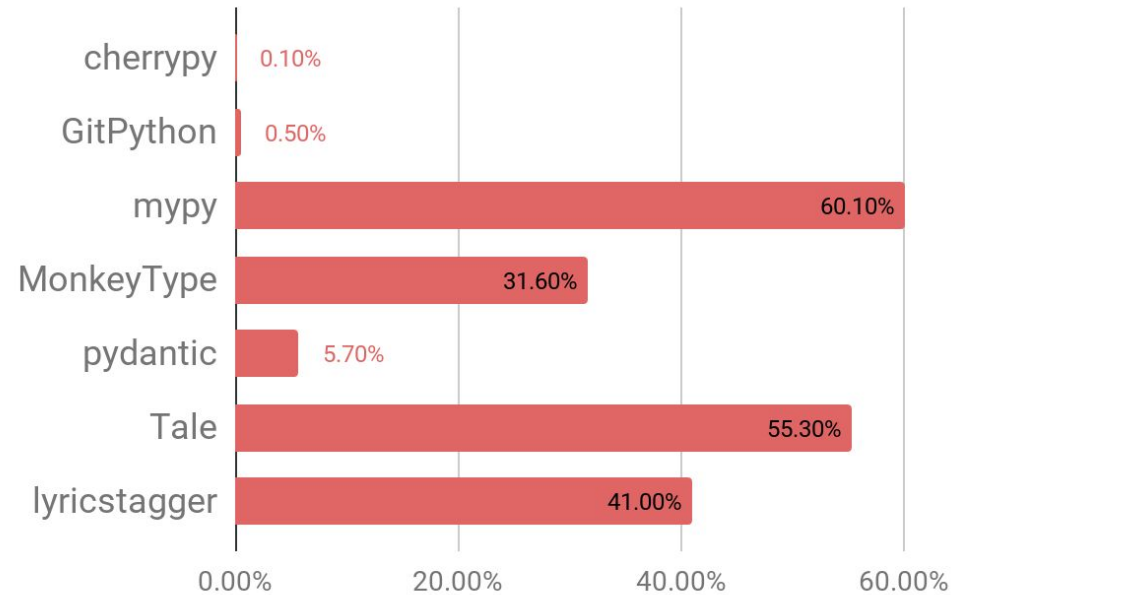23.8%
5

Do not use type hints
66.7%
14

Why are type hints not popular?

- Lot of people do not know about it
- Typing is not that popular among Python developers
- Not helpful documentation and tutorials
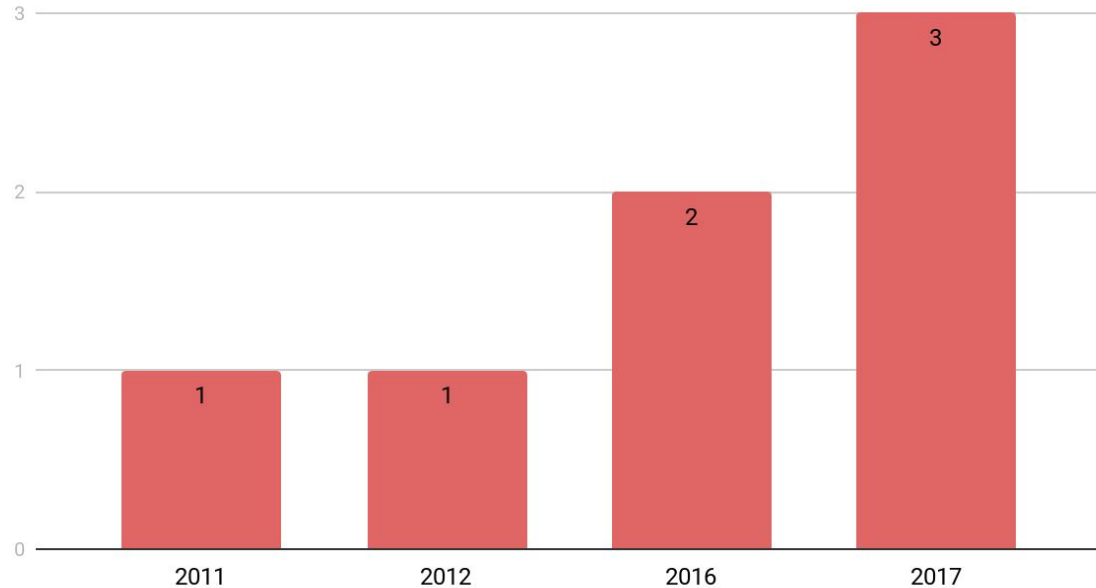- mypy and PyCharm support is partial and buggy
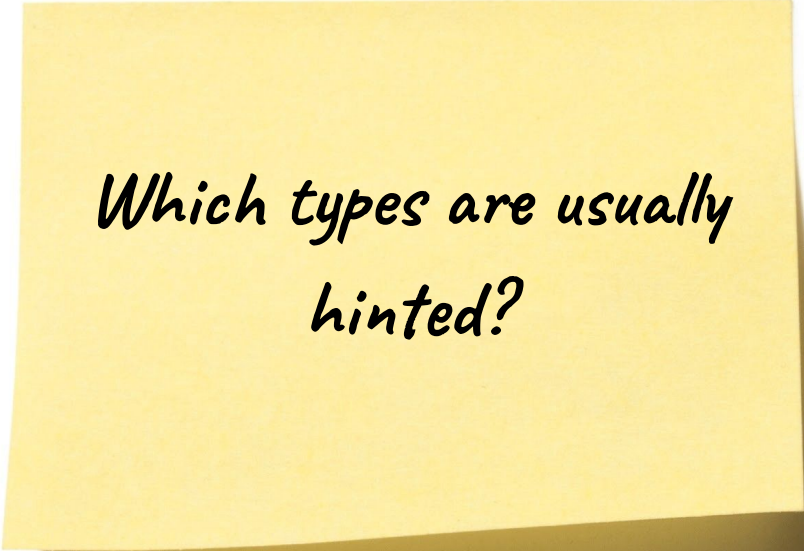
Usage in projects and libraries

Usage of Type Hints in projects and libraries

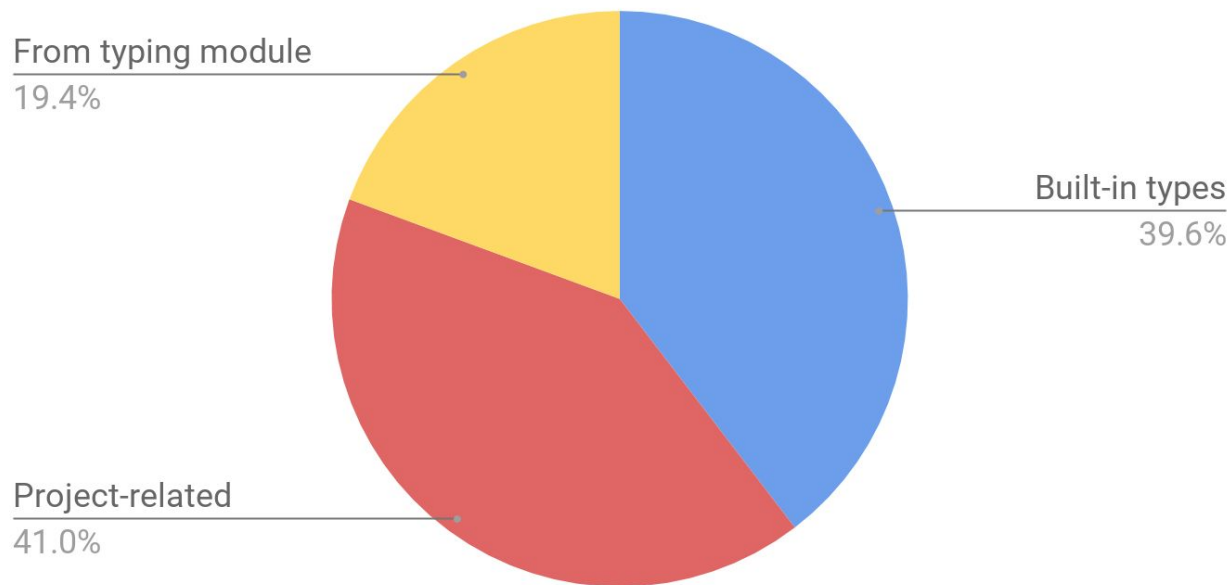| Library	Usage
cherrypy	0.10%
GitPython	0.50%
mypy	60.10%
MonkeyType	31.60%
pydantic	5.70%
Tale	55.30%
lyricstagger	41.00%

Usage in projects and libraries

Creation year of projects and libraries using type hints

Which types are usually hinted?

Most hinted types

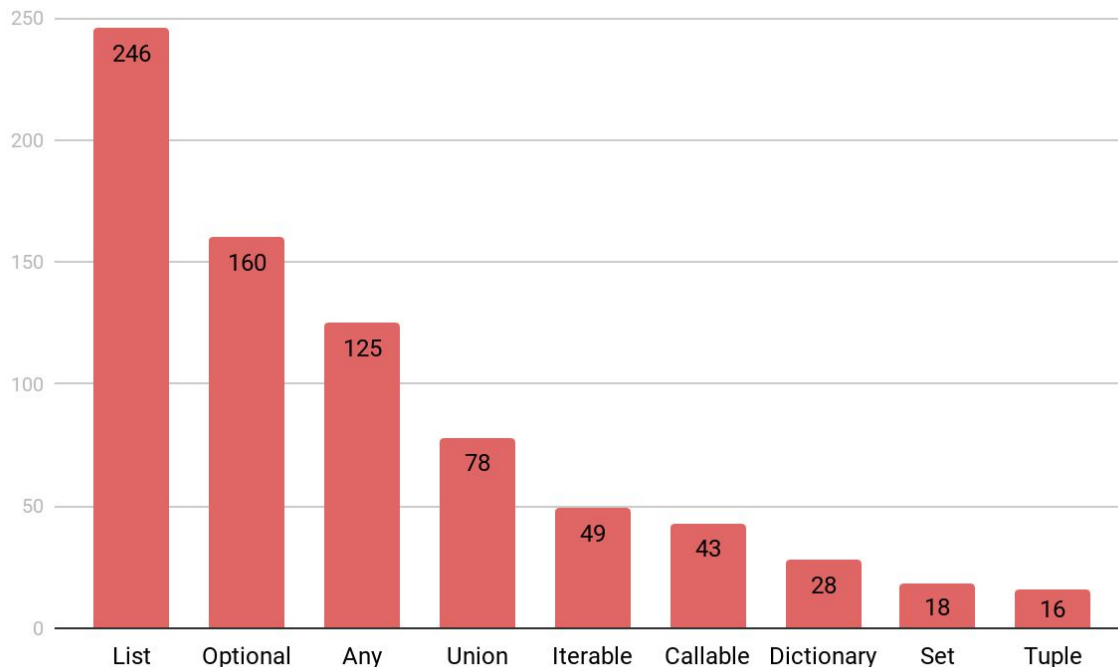From typing module
19.4%

Built-in types
39.6%

Project-related
41.0%

Hinted built-in types
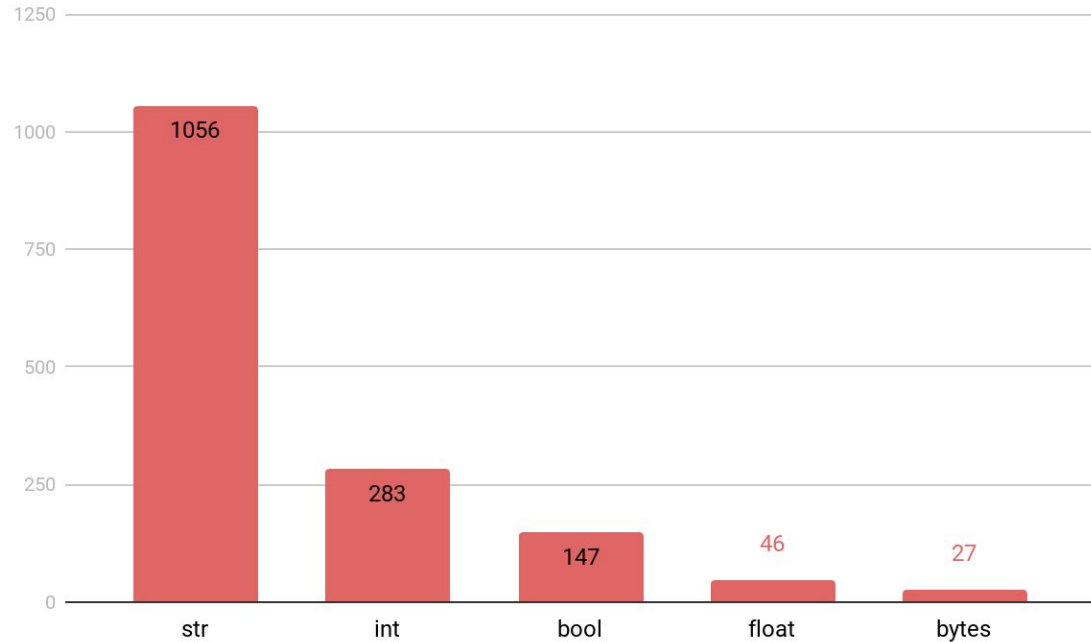
Hinted types from typing module

Conclusion

- Programmers **do not use** often type hints
- Type hints are usually used **in libraries** implementation
- Mostly hinted types are **project-related** types
- The most hinted built-in type is **string**

Conclusion

It is **not always possible** to predict an argument type by analysing its name...

```python
def fetch(artist: str, song: str, album: str) -> str:
    Helpers = hlp.HELPERS
    lyrics = None
    for helper in helpers:
        try:
            lyrics = helper.fetch(artist, song, album)
        except requests.ConnectionError as error:
            log.warning('Connection error: %s', error)
        if lyrics:
            break
    return lyrics
```
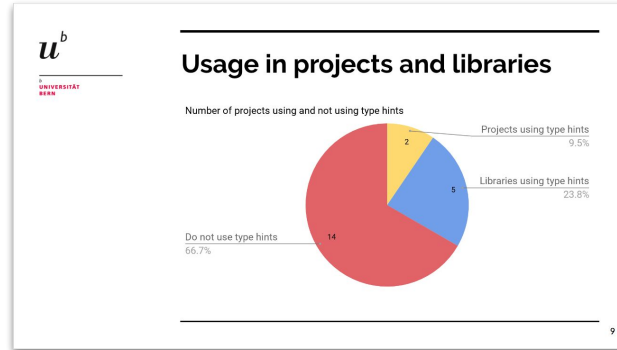
Conclusion

…but we can make assumptions:

- a name starting with a verb (e.g. **has_connected**) is very likely to be **boolean**
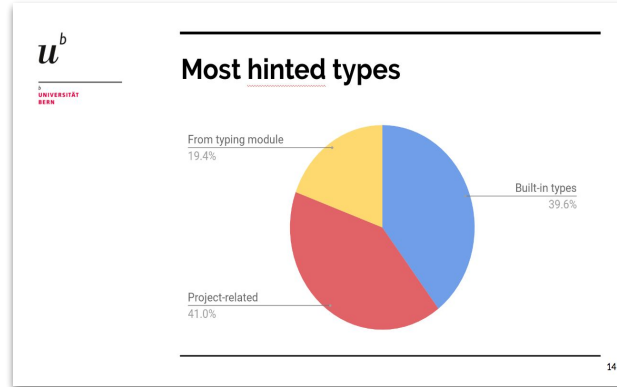- a noun in plural form (e.g. **connections**) probably represents a **collection**

Future work

- Find more projects which use type hints
- Work more on argument type prediction
- Perhaps write a paper for conference

Summarize

Usage in projects and libraries

Number of projects using and not using type hints

- Projects using type hints 9.5%
- Libraries using type hints 23.8%
- Do not use type hints 66.7%

Why are type hints not popular?

- Lot of people do not know about it
- Typing is not that popular among Python developers
- Not helpful documentation and tutorials
- mypy and PyCharm support is partial and buggy

Most hinted types

- From typing module 19.4%
- Built-in types 39.6%
- Project-related 41.0%

Conclusion

- Programmers **do not use** often type hints
- Type hints are usually used **in libraries** implementation
- Mostly hinted types are **project-related** types
- The most hinted built-in type is **string**