Improve Your Code with Type Checking

Reindert-Jan Ekker

@rjekker http://nl.linkedin.com/in/rjekker

Overview

Static typing in Python

- Type hints
- Gentle introduction
- Why?

Not: type theory

Not: generics, custom types, etc.

Static vs Dynamic Typing

Static typing (Java, C#, ...)

- Type declarations in code
- Variable types
- Function argument types
- Can be checked statically (compile-time)

Dynamic typing (Python)

- No type information in code
- Type checking done at runtime

Demo

Static vs. dynamic typing%

Static Typing in Python

Type hints

- Optionally add type information
- Ignored by Python interpreter
- Running gives the same results

Type checker: mypy (or Pycharm)

Work in progress

- Will become part of Python language

Best used with Python 3.6 or newer

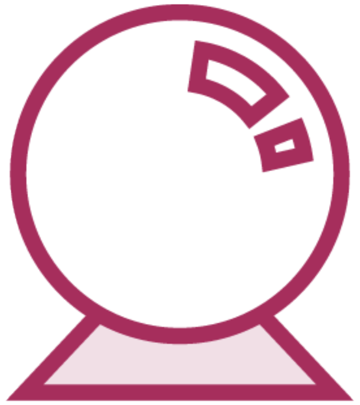
Variable Hints

```
# Declare the type of a variable in Python 3.6
age: int = 1

# In Python 3.5 and earlier use a comment
age = 1  # type: int
```

Function Hints

```
# Two int arguments, return a float
def plus(num1: int, num2: int) -> float:
    return num1 + num2


# Add default value for an argument after the type
annotation
def f(num1: int, my_float: float = 3.5) -> float:
    return num1 + my_float
```

Static Typing in Python: Why?



Find bugs at compile-time

Easier maintenance

- Type hints document your code
- Improved IDE support

Better program design

But: no need to use it always

Demo

Applying type hints in a project

Demo

Mypy

- Command line type checker

Summary

Gentle intro into type hints

Adding simple type hints

Static type checking with mypy/Pycharm

Prevent errors

Improve maintainability