Monkey Type, Monkey Do

One Million Monkeys Will Eventually Write Correct Type Hints

Moshe Zadka – https://cobordism.com

2020

Acknowledgement of Country

Belmont
Ancestral homeland of the Ramaytush Ohlone

Python Type Hints

- Not a new idea
- ...but now, checked!

Example

```
def add(x: float, y: float) -> str:
    return str(x + y)
```

Progressive typing

```
def add_with_extra(x: float, y: float) -> str:
    return str(x + y + extra())
def extra():
    return "0.1"
```

Progressive typing

Type Error in `add_with_extra`

```
def add_with_extra(x: float, y: float) -> str:
    return str(x + y + extra())
def extra() -> str:
    return "0.1"
```

Progressive typing

Type Error in `extra`

```
def add_with_extra(x: float, y: float) -> str:
    return str(x + y + extra())
def extra() -> float:
    return "0.1"
```

Progressive typing

```
def add_with_extra(x: float, y: float) -> str:
    return str(x + y + extra())
def extra() -> float:
    return 0.1
```

Making Progress on Typing

Now do the same

Making Progress on Typing

Now do the same
on your 1000 line project

Making Progress on Typing

Now do the same
on your 1000 line project
or your 10,000 line project

Progressive Non-linear Benefits

Simple model:
$P_B(Q) = Q^2$
So

- $P_B(0.9)$ 0.8 Yay

Progressive Non-linear Benefits

Simple model:
$P_B(Q) = Q^2$
So

- $P_B(0.9)$ 0.8 Yay
- $P_B(0.5)$ 0.25 OK...

Progressive Non-linear Benefits

Simple model:
$P_B(Q) = Q^2$
So

- $P_B(0.9)$ 0.8 Yay
- $P_B(0.5)$ 0.25 OK...
- $P_B(0.2)$ 0.04 Can't make a case to management

How Do You Know What the Types Are?

- Comments?
- Docs?
- Ducks?

Classical Empiricism

The only way to gain knowledge is to interact with the world

monkeytype: Type instrumentation

- Slow-down
- Actual data

monkeytype in prod

Like Facebook, you have .... millions of servers?

Maybe not monkeytype in prod

That sounds a bit scary

Your Test Suite

- Runs through all your code (right?)
- Over and over
- On many machines

Tests

- Common case
- Corner cases
- Weird cases?

Test Isolation

Little risk

Running Tests Normally

```
$ tox -e py38
```

Environment Ready, Now Observe

```
$ ./.tox/py38/bin/pip install monkeytype
```

Environment Ready, Now Observe

```
$ ./.tox/py38/bin/monkeytype run -m virtue regret
                                 ^^
                                 Like Python's -m
```

Environment Ready, Now Observe

```
$ ./.tox/py38/bin/monkeytype run -m virtue regret
                                    ^^^^^^^
                                    Test runner
```

Environment Ready, Now Observe

```
$ ./.tox/py38/bin/monkeytype run -m virtue regret
                                           ^^^^^^
                                           Location
```

Data Ready, Calculate Types

```
$ ./.tox/py38/bin/monkeytype stub regret
```

Types Ready, Write Code

```
$ ./.tox/py38/bin/monkeytype apply regret._api
```

Check the Monkey

```
$ git diff
--- a/regret/_api.py
+++ b/regret/_api.py
```

Check the Monkey

```
-       def inheritance(self, version):
+       def inheritance(self, version: str) -> Callabl
```

Check the Monkey

```
     def callable(
             self,
-            version,
-            replacement=None,
-            removal_date=None,
-            addendum=None,
-    ):
+            version: str,
+            replacement: Optional[Callable]=None,
+            removal_date: Optional[date]=None,
+            addendum: Optional[str]=None,
+    ) -> Callable:
```

Take Credit From Monkey

```
$ git checkout -b add-types
$ git commit -a -m 'Add types to functions. I did i
```

Take Credit From Monkey

```
$ git checkout -b add-types
$ git commit -a -m 'Add types to functions. I did i
```

Open a PR

Circular Imports

MonkeyType does not support them

Circular Imports

MonkeyType does not support them
and they're everywhere.

Circular Imports

MonkeyType does not support them
and they're everywhere.
(But probably good to get rid of anyway.)

Types Specificity

Under

Types Specificity

Under
and over

Conclusion

- Types help non-linearly
- You already have tests (right?)
- Get a leg up from the monkey