

Types in programming languages (discussing worksheet and more)

① Why types? (what if not?)

* - debugging harder? what are we supposed to be?

- what happens with + when you give it a string?

- error?

- suppose it to concatenate?

→ ... but how does it know?

e.g.

- Can't do any error checking relating to types * compile vs runtime

- had to guarantee behavior

- hard to implement language

(? or easy, cause can't do much)

What is a type?

- defines set of possible values
- how stored/represents

[standardized?
not for C, for example]
DESIRABLE

- attributes
 - len, +h (strings)
 - # of dimensions (list)
- possible built-in operations/behavior

In Java, an Integer has a
toString method that returns
a string representation

④ checking before program runs
or while running

Critical difference in PLs: ^{static type checking}
- do I check for correct use of types before program runs [by checking all code first],
or at runtime [while the program is running?]
^{dynamic type checking}

True or False? To do static type checking, I need a compiler.

- You could totally do it with an interpreter. Checking for type accuracy \neq translating code.
- The first step of the interpreter could be to scan the whole program looking for type errors.

True or False: dynamic type checking means there are no types.

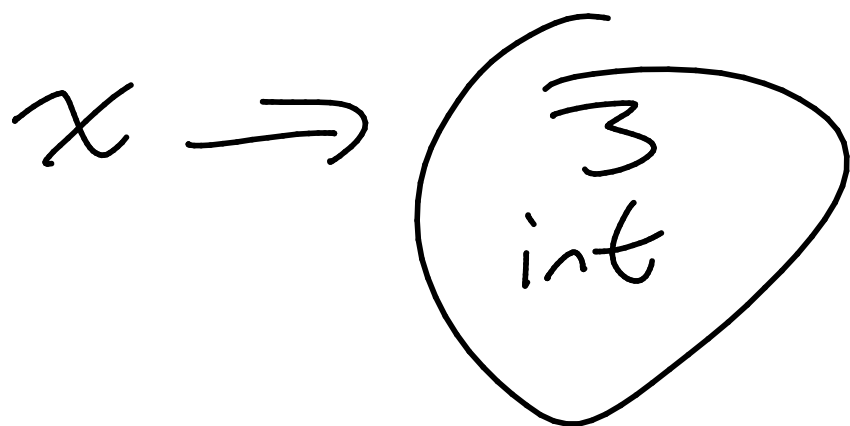
(Python, for example.)

[T or ~~F~~: Python has no types.]
T: Python has no type declarations

In Python, there are no types in the variables. $x = 3$

x, the variable, has no type.

Data has types. 3 is an int



Pros/cons of static vs dynamic type checking:

- static: can catch errors before running program (pro)
 - dangerous errors, know before running
 - during runtime, it might depend on path through the code, which is harder to diagnose
- dynamic - checks need to be made while program is running, every time an op is done → takes more memory to store type info
 - slows down program
- dynamic: easier to write code (faster) ↖ only advantage(?)

Most langs aren't completely one or the other.

Where does Python do something static-ish?

- see `globalfun.py`

Where does Java/Kotlin do something dynamic-ish?

- casting types of classes from one to another, in inheritance structures

```
String s = "hello"
```

```
Object obj = (Object) s;
```

```
String t = (String) obj;
```

fails if [^] obj was not a String originally

static type checking is pretty good
Python now has a retrofitted
Static type checker on top of it.