

Gradual Typing

Aka Progressive Typing

Whazzat??



Typing?

Static versus Dynamic Typing

Weak versus Strong Typing



Static versus dynamic typing

Java is statically typed = variable types are fixed at compile time -> type errors are detected at compile time.

```
Import java.io.*;
public class Main {
    public static void main(String [] args) {
        int i;
        i = 100;
        i = System.out;
    }
}
```



Static versus dynamic typing

```
[erwan@lookfar ~] $ javac Main.java
```

```
Main.java:12: incompatible types
```

```
found   : java.io.PrintStream
```

```
required: int
```

```
i = System.out;
```

```
      ^
```

```
1 error
```



Static versus dynamic typing

Perl is dynamically typed = types are discovered and re-assigned at runtime.

```
my $a;           # no type declaration
$a = 1;         # $a is an int
$a = new Foo::Bar(); # now $a is a ref
```



Weak versus Strong typing

Python is strongly typed: a typed variable can only contain one type of data. Assigning a different type requires an explicit cast.

```
i = 0          # now i is an integer
i = i + 'Boom!!' # error: strong type checking
i = 'OK'       # now it's a string: dynamic typing
```



Weak versus Strong typing

Perl is weakly typed: automatic type cast at runtime.

```
my $a;      # no type declaration
```

```
a = "123" + 12; # silent type cast
```



Popular Beliefs

”Statically typed languages are safer than dynamically typed ones.”

Because:

”It's better to catch type errors at compile time than during runtime.”



Popular Beliefs

”Strongly typed languages are safer than weakly typed ones.”

Because:

”It's better to get a type error than no error at all.”



Consequence

Dynamic/weakly typed languages are considered good for prototyping only.

“Serious applications” tend to be written in strongly/statically typed languages.



Consequence

Most applications written in dynamic/weakly typed languages (perl/python) are at some point re-written into strongly and statically typed languages (Java, C, C#).

It's called a Big Bang.

It often ends up with a Big Crash.



How to avoid rewrite?

Gradual Typing!!



Gradual Typing

Allow both static and dynamic typing within the same language: some variables are typed at compile time and the compiler can validate them, while some are dynamically typed.



Gradual Typing

Allow both weak and strong typing within the same language: typed variables require explicit casts, untyped variables trigger silent type casting.



Gradual Typing in practice:

Files/classes/modules are first written in a weakly/
dynamically typed fashion hence allowing for fast
prototyping.



Gradual Typing in practice:

As the code matures and stabilizes, types emerge and the same file/class/module can progressively be rewritten to use strongly/statically typed syntax.

The compiler and the virtual machine do the magic.



Gradual Typing:

Challenges:

- Compiler and virtual machine need to be specially designed
- Unclear how to handle some specific cases
- Still a debate whether it would really lead to safer code



The bigger picture

This is really about how to have fast flexible prototyping *and* safe coding within the same language.

Variable typing is just one aspect.

Everything that leads to better error detection is relevant.



Gradual Typing and Perl6:

Interesting features:

- Sub/method signatures
- Static variable type checking
- Roles
- Constraints
- More?



PPM rekryterar

Intresserad?
Fråga mig sen :-)

