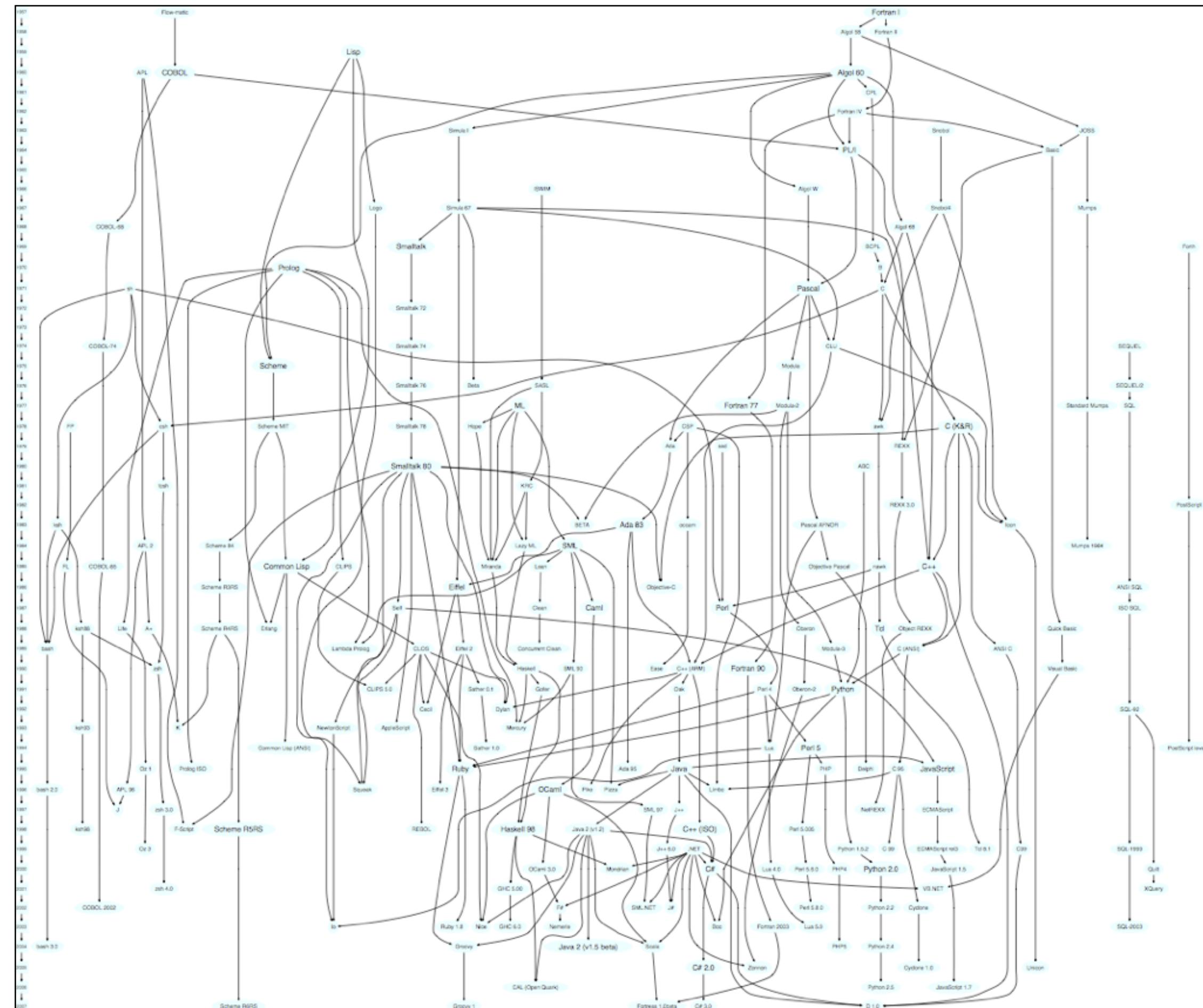


Topic List

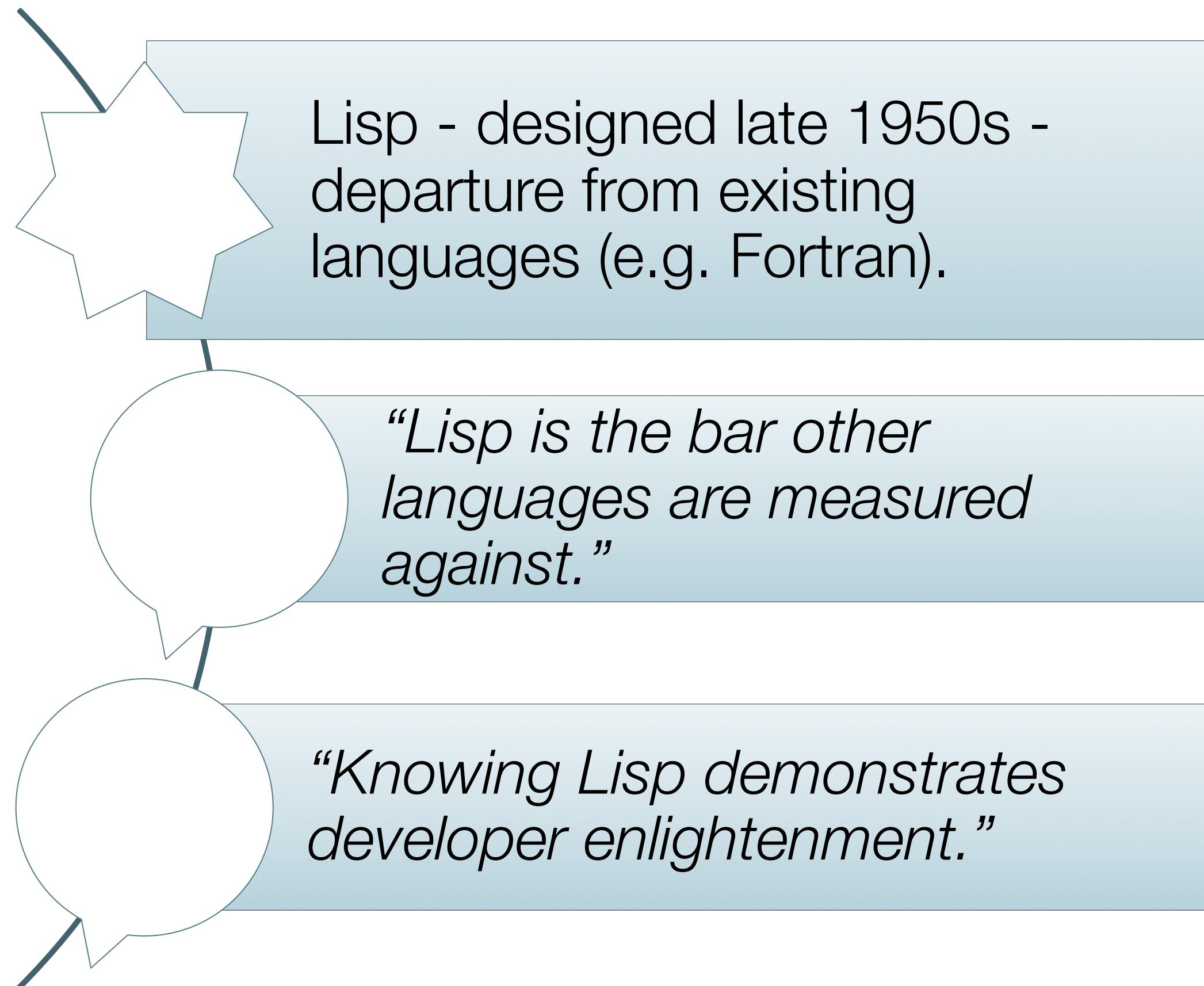
Programming Languages:

- Family Trees.
- Typing Spectrum.

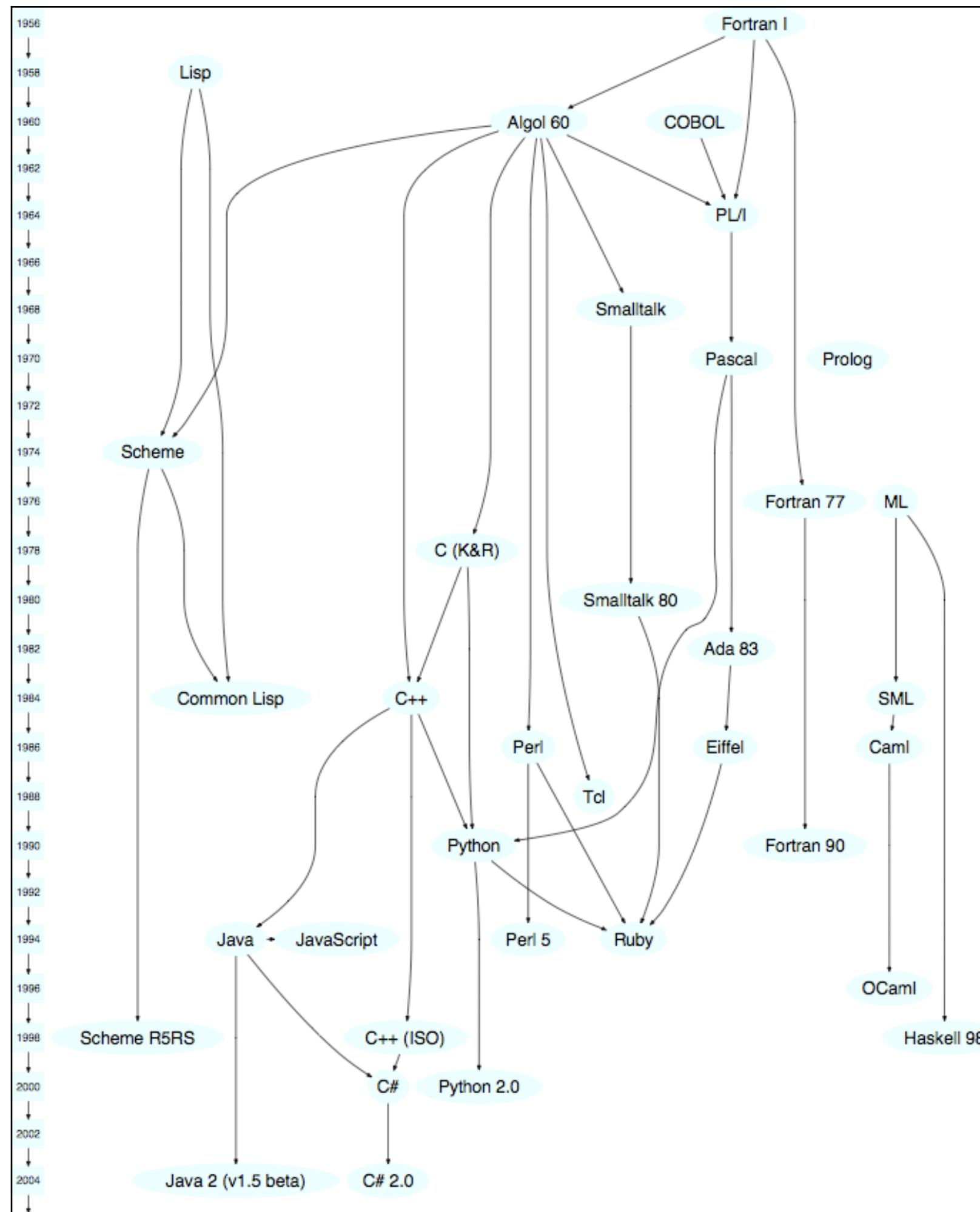
Family Tree



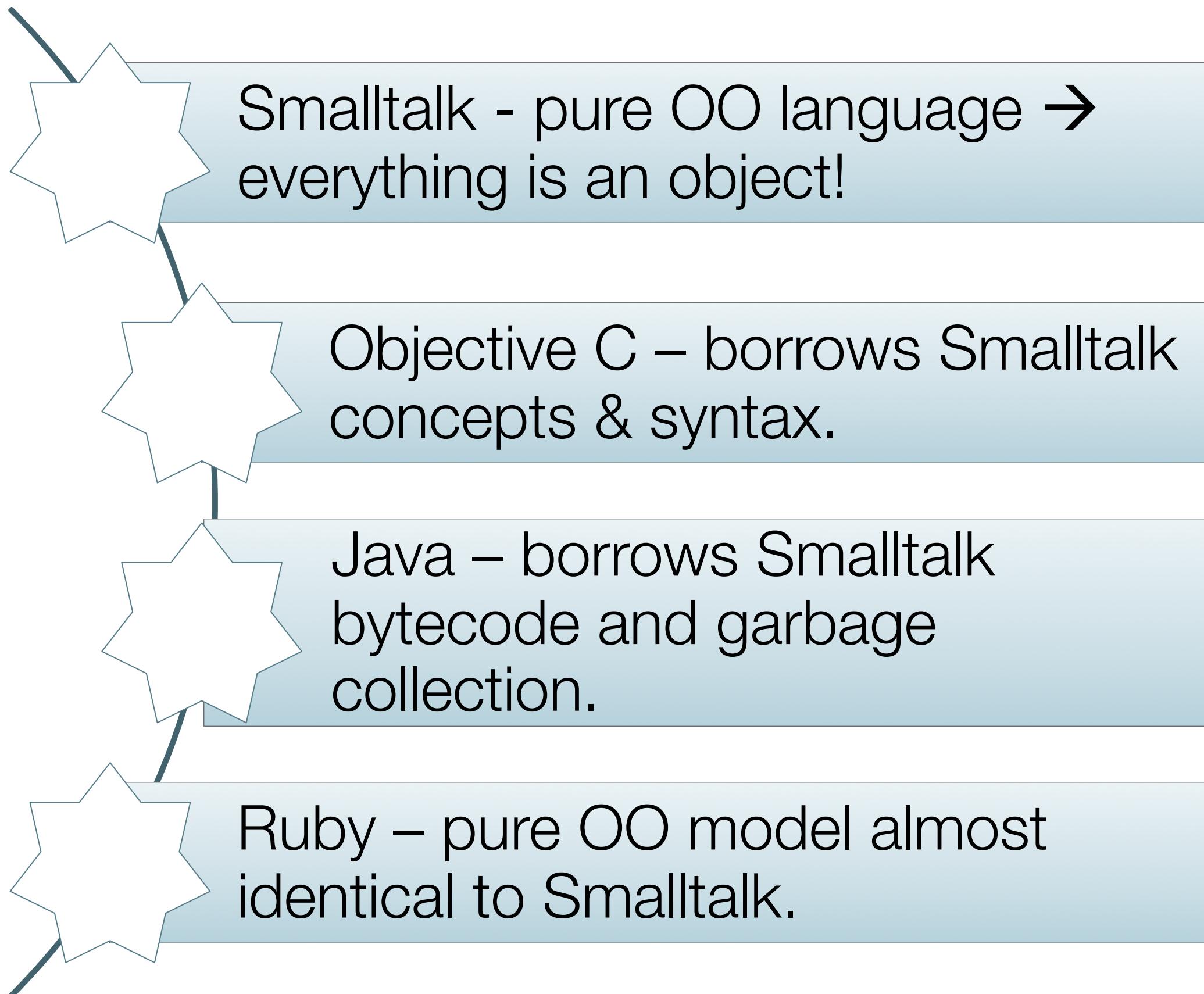
Family Tree



Family Tree

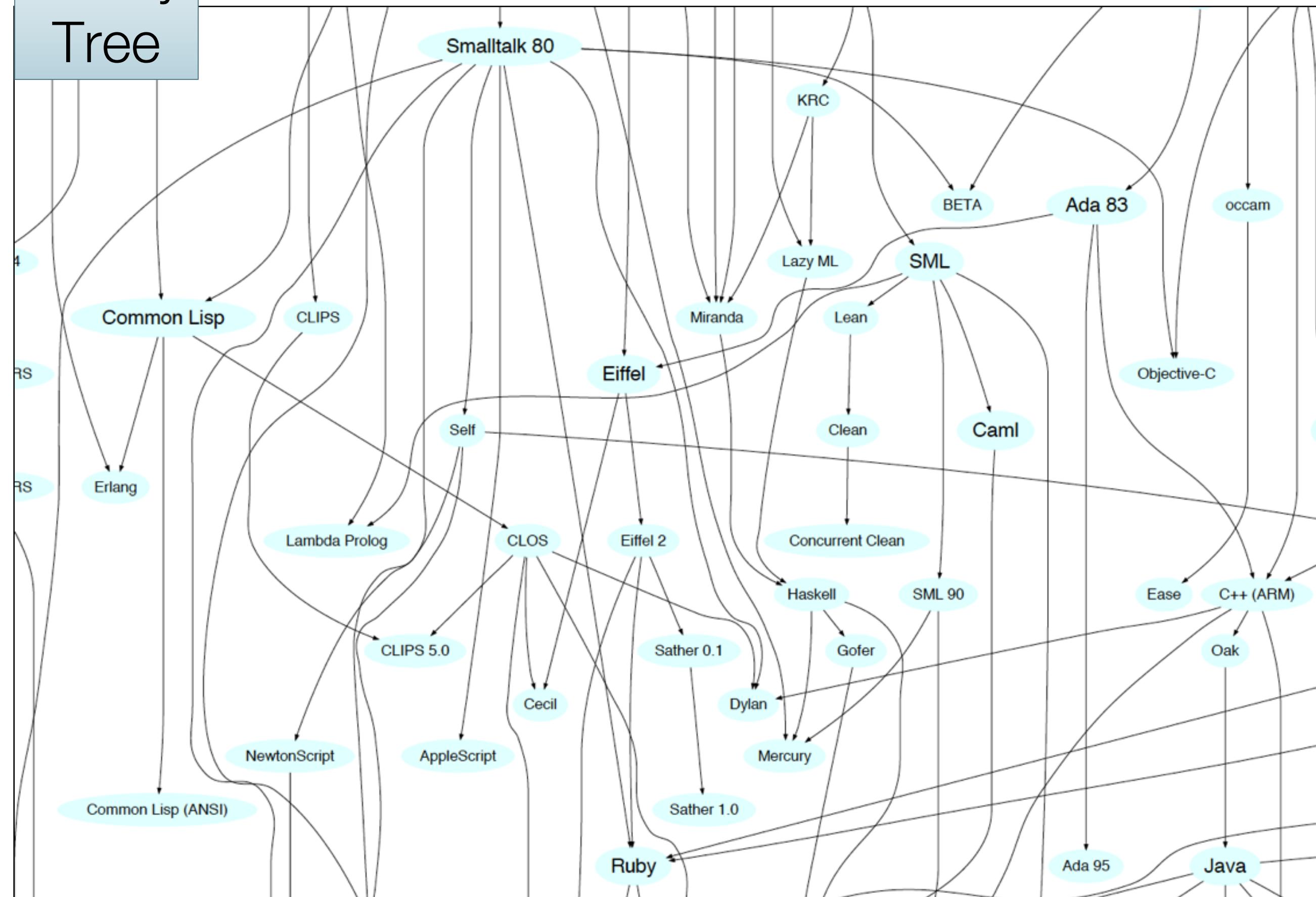


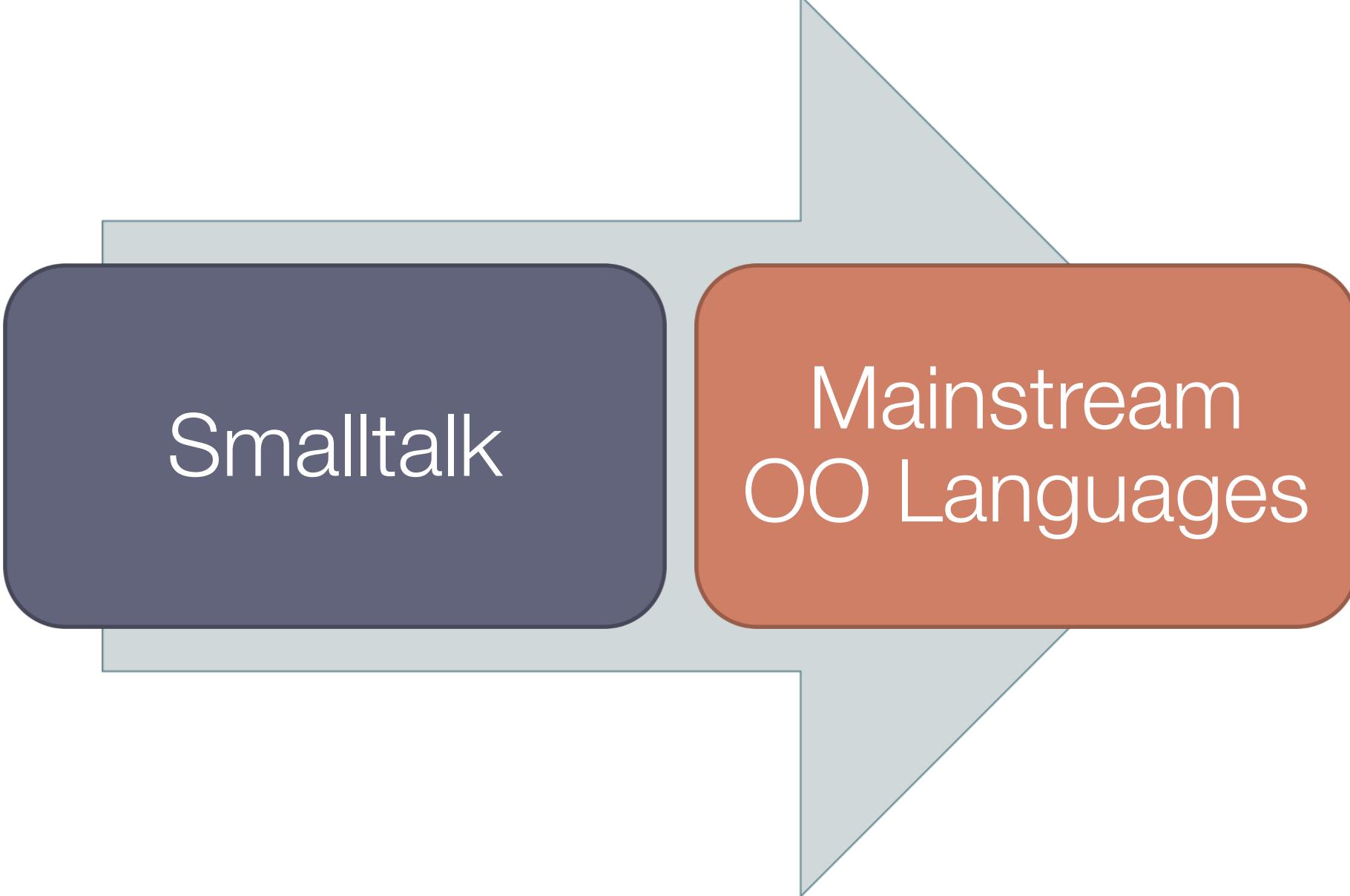
Family Tree



Family Tree

Smalltalk Cluster



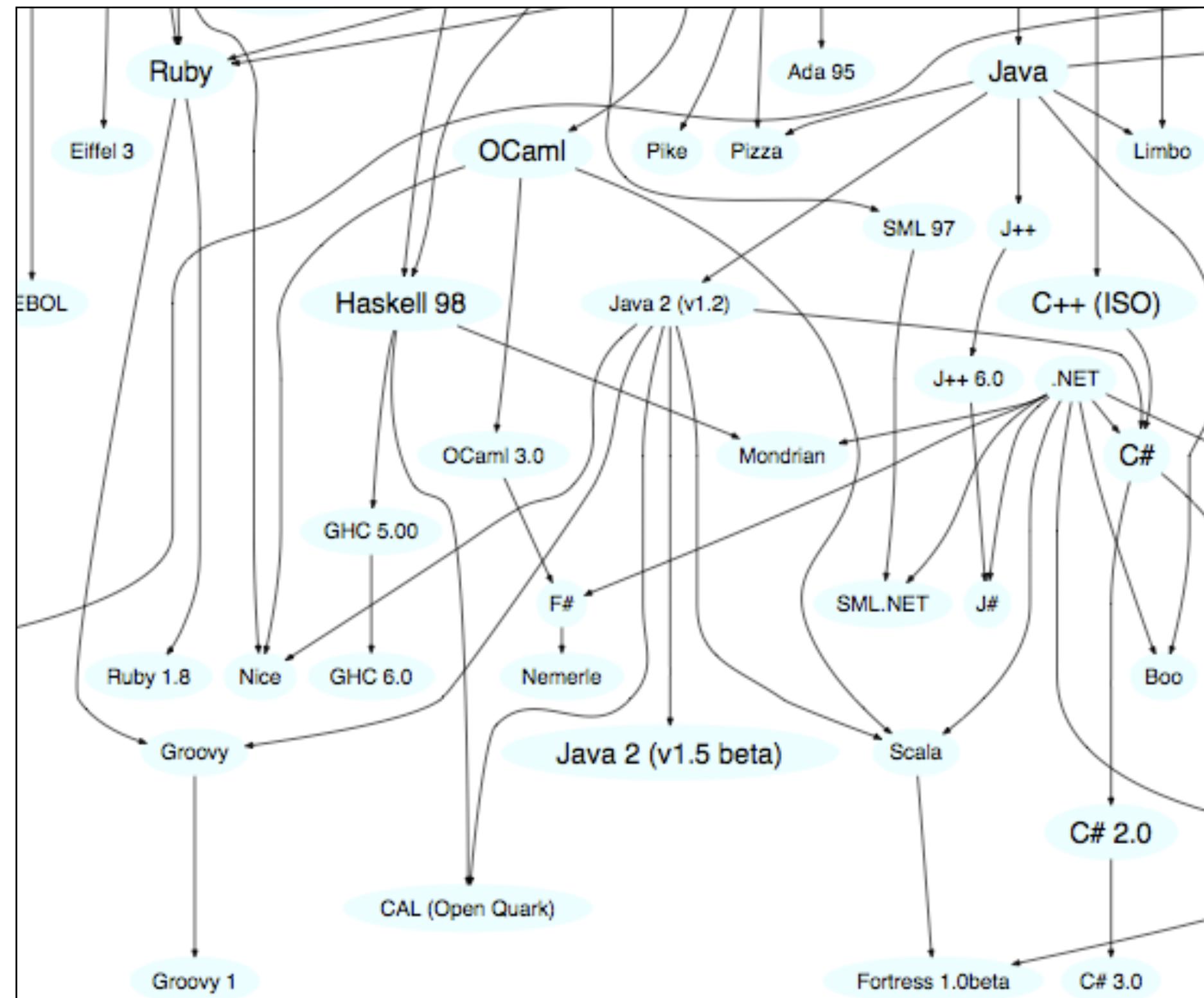


Smalltalk

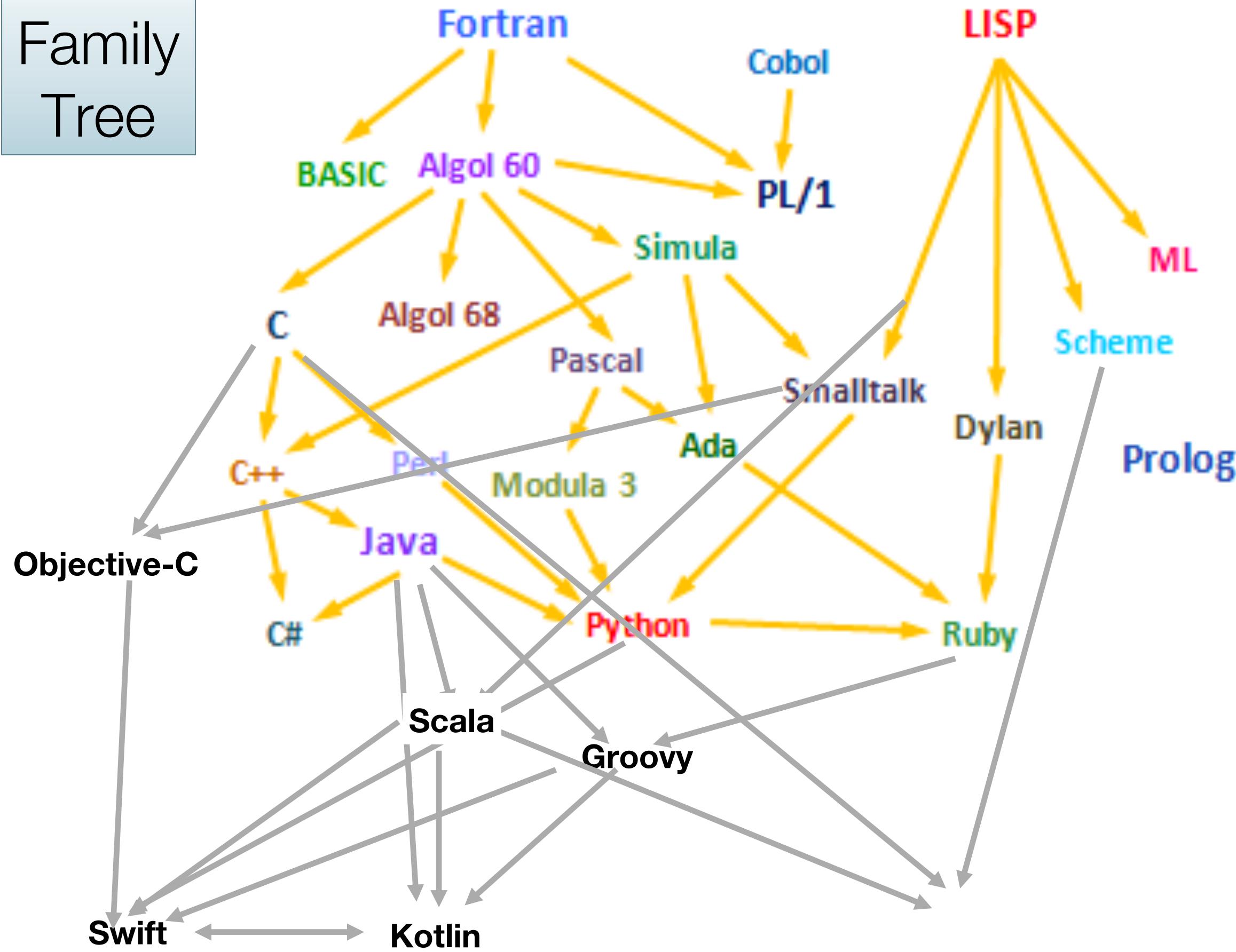
Mainstream
OO Languages

Family Tree

Ruby, Groovy, Java, Scala Cluster



Family Tree

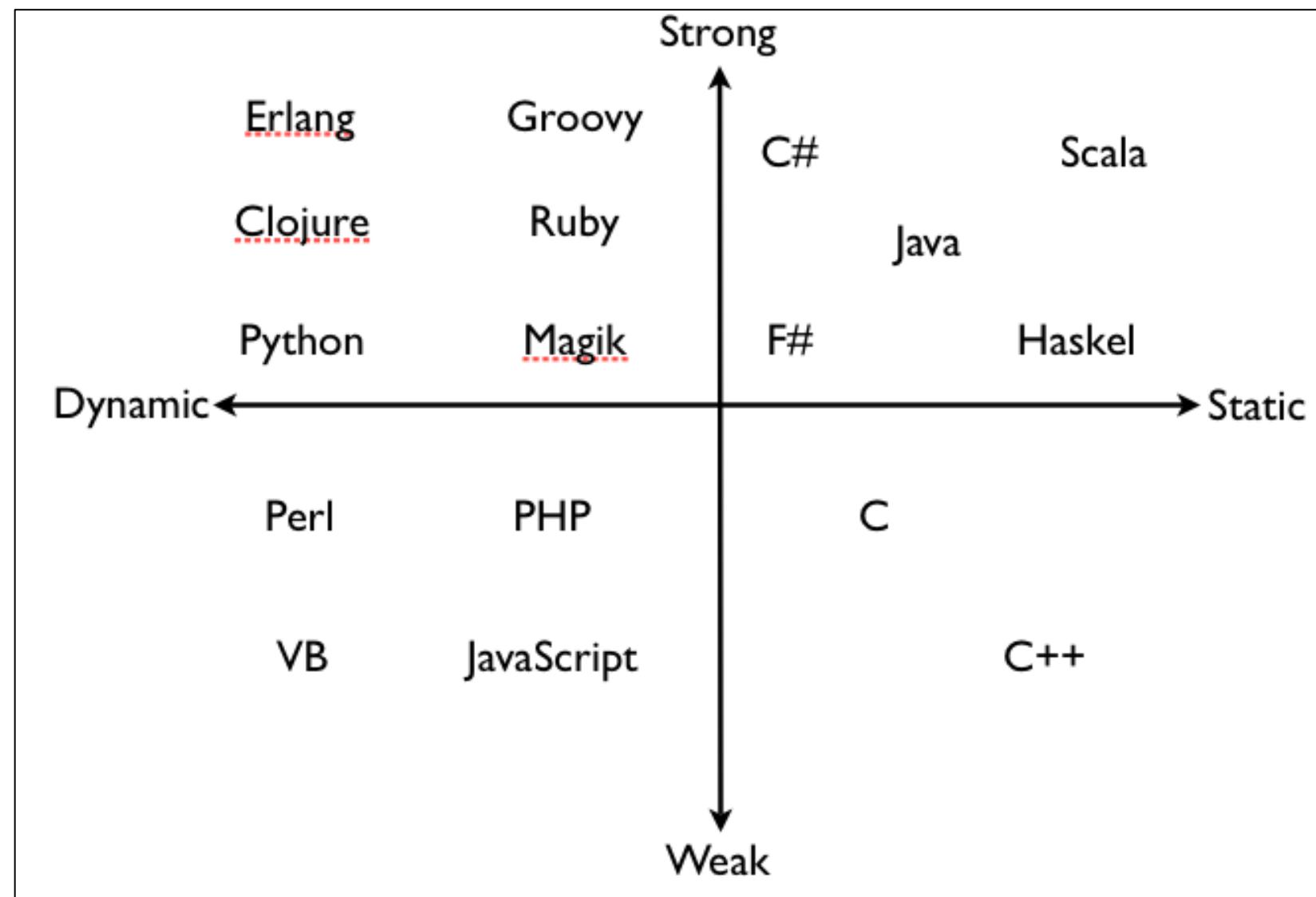


Typing

The concept of applying a “type” to a variable

Typing Spectrum

Languages are often classified based on their approach to typing...



Defining Typing Terms...

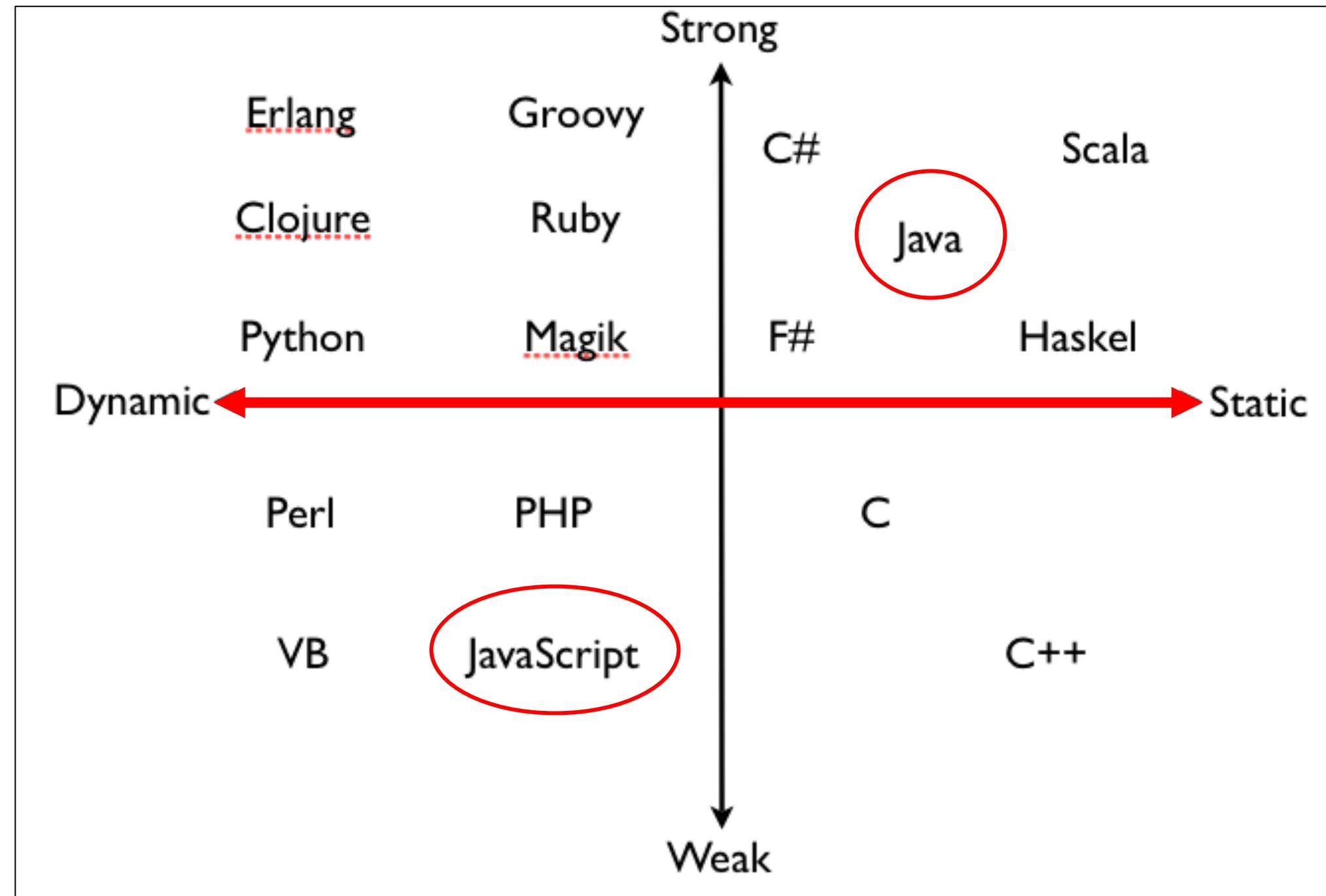
“There is widespread confusion or disagreement about the meanings of the words

static, dynamic, strong and weak

when used to describe the type systems of programming languages”

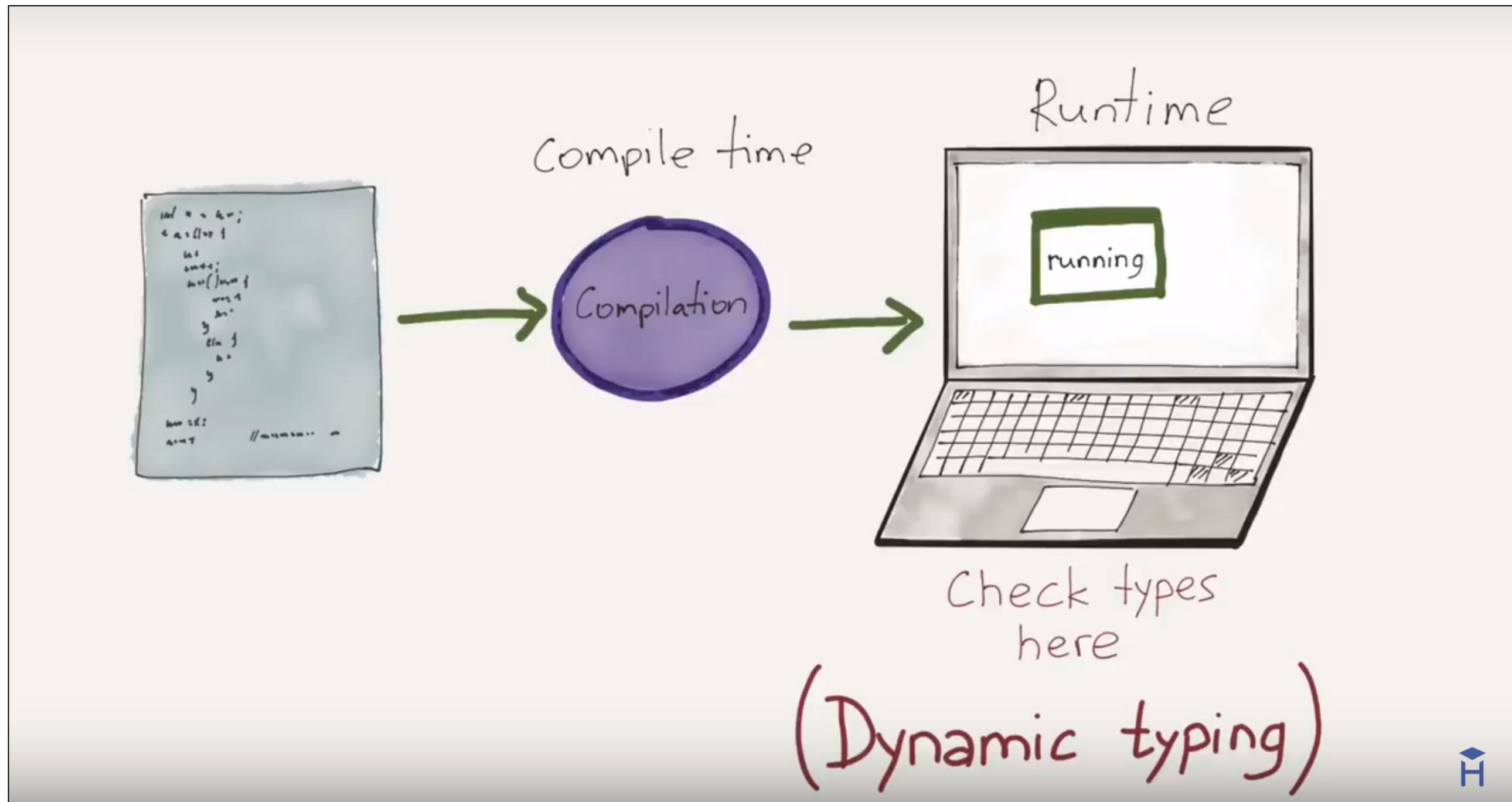


Dynamic ↑ Static



Dynamic Typing

*“Variables’ type declarations
are not mandatory
and they will be
generated/inferred on the fly,
by their first use.”*

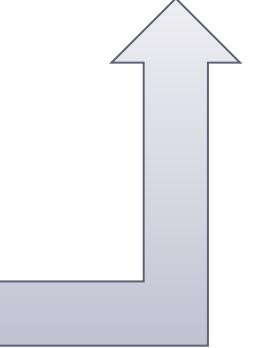


Dynamic Typing – Example



```
var greeting = ; //undefined  
var someRandomInteger = 100;  
var aDoubleVariable = 2.2;  
greeting = "Hello!";
```

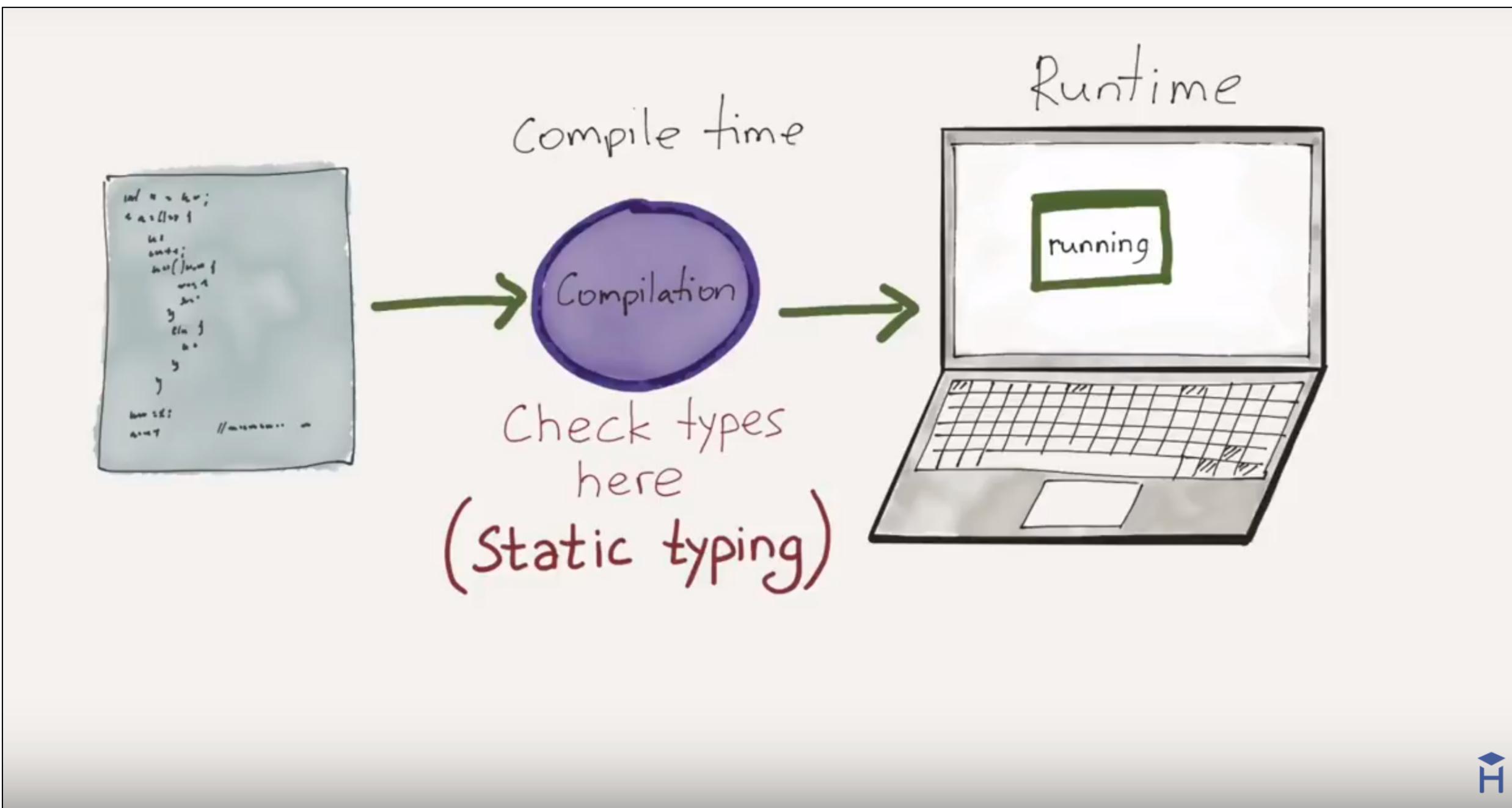
A type is NOT assigned to the variables.



The JavaScript engine will choose a type that it feels best describes the data that's contained inside of your variable → assign datatype behind the scenes.

Static Typing

*“Variable declarations are mandatory
before usage, else
results in a compile-time error”*



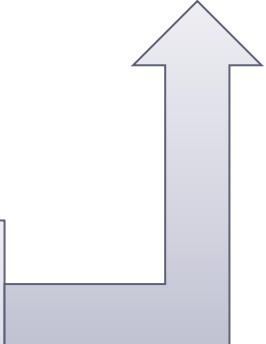
H

Static Typing – Example



```
String greeting = "Hello!";  
int someRandomInteger = 100;  
double aDoubleVariable = 2.2;
```

A type is assigned to each variable.

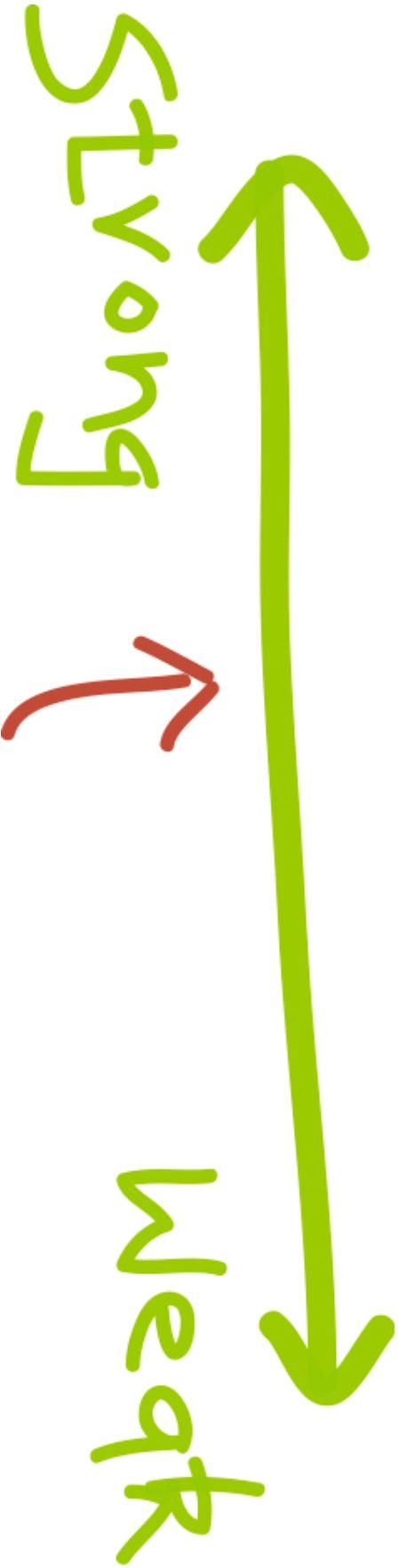


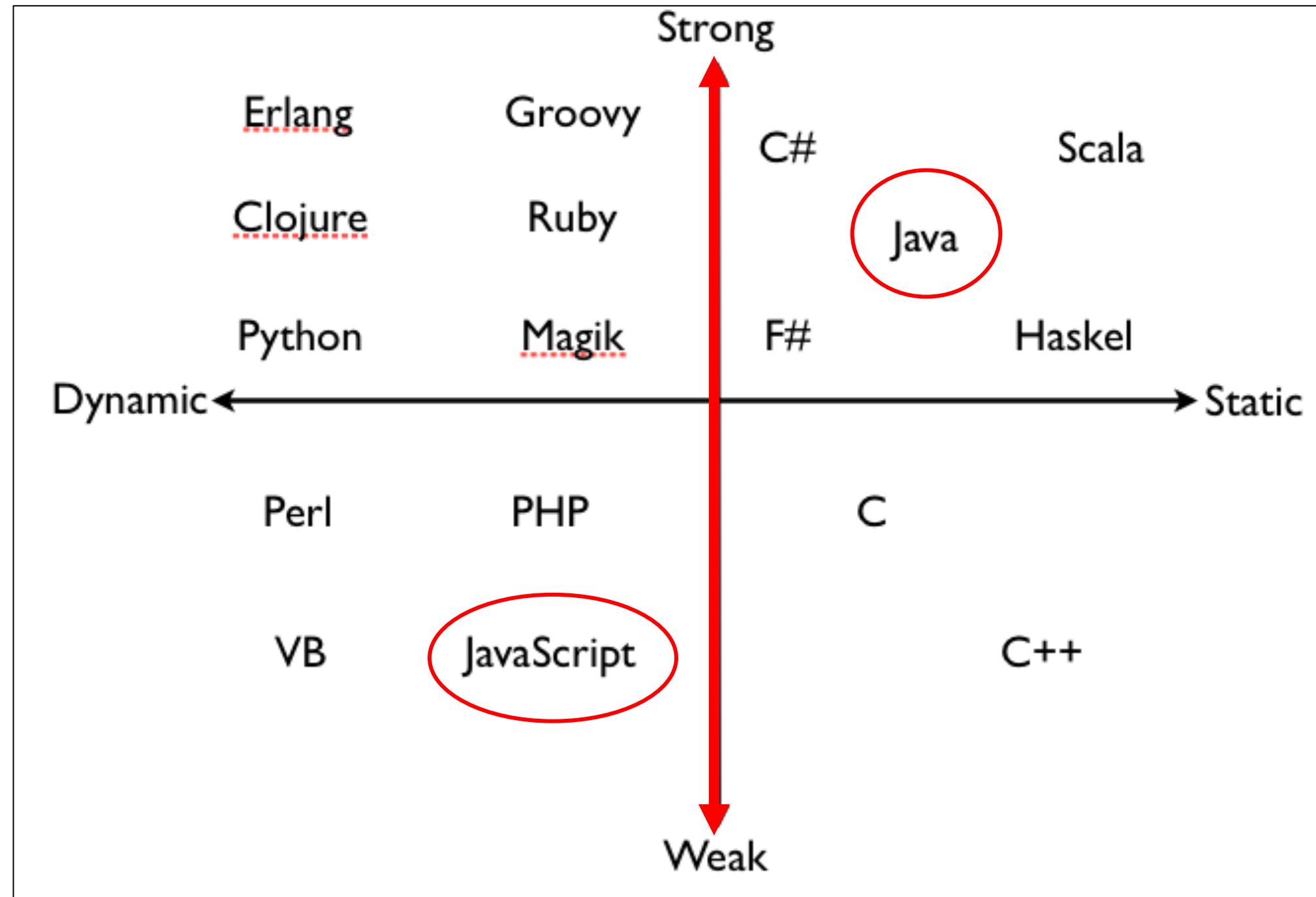
In Java, if we don't assign a type, we get a compiler error
→ Java is statically typed.

Types determine the operations we can perform on the variables.



Amount of type checking enforced by the compiler vs. leaving it to the runtime





Strong Typing

“Once a variable is declared as a specific data type, it will be bound to that particular data type.

You can explicitly cast the data type though.”

Strong Typing – Example 1



```
*StrongTyping.java ✘
1
2 public class StrongTyping {
3
4     public static void main(String[] args) {
5
6         int numberOne = 4;          //static typing
7         int numberTwo;
8
9         numberTwo = 4.6;
10
11    }
12
13 }
14
```

A screenshot of a Java code editor showing a type mismatch error. The code defines a class named 'StrongTyping' with a main method. In the main method, there is a line of code: 'numberTwo = 4.6;'. The editor highlights the decimal point '4.' with a red underline. A tooltip box appears over the line, displaying the error message: 'Type mismatch: cannot convert from double to int' and '2 quick fixes available:'. The first fix is 'Add cast to 'int'' and the second is 'Change type of 'numberTwo' to 'double''. At the bottom of the tooltip box, it says 'Press 'F2' for focus'.

Strong Typing – Example 1 (fix with casting)



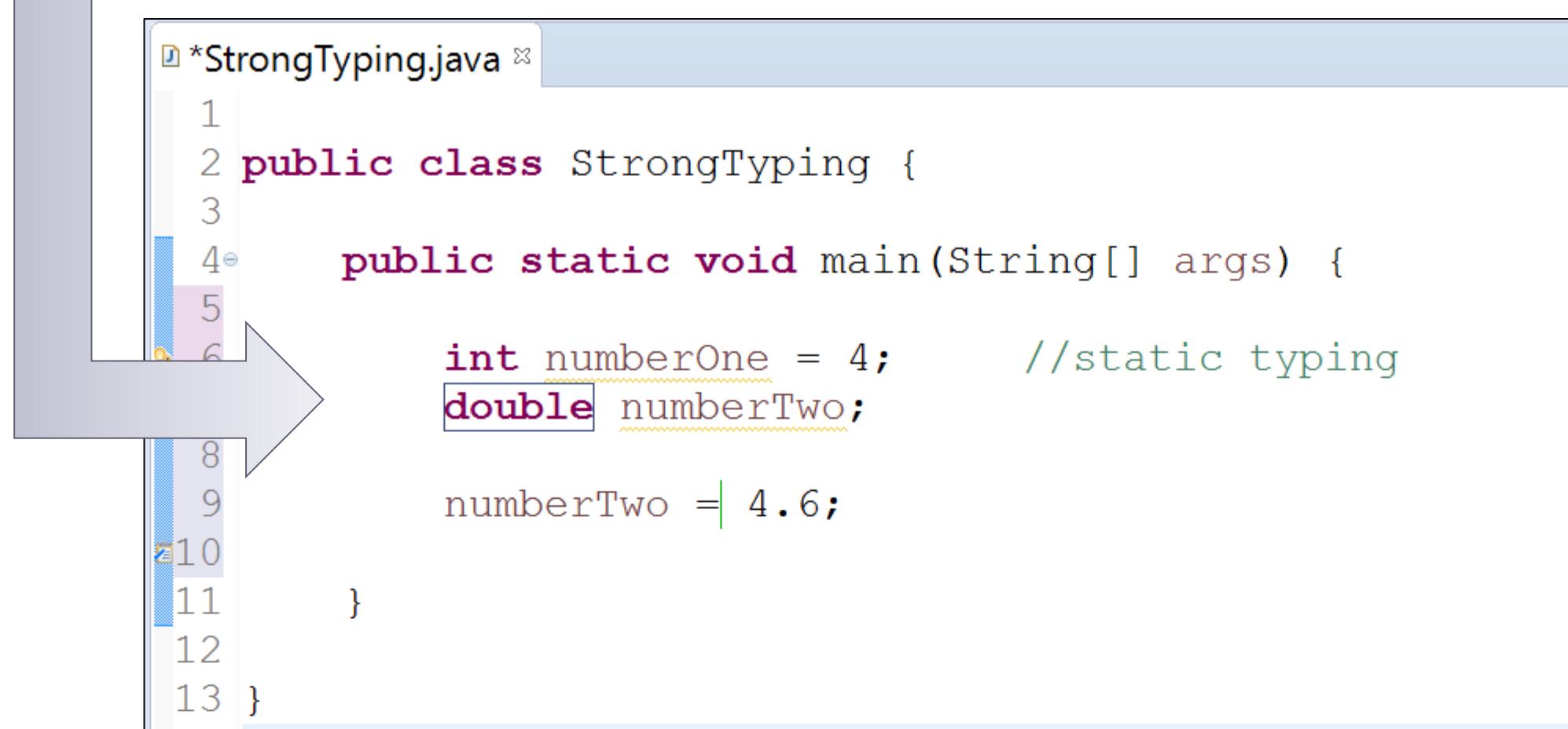
```
*StrongTyping.java
1
2 public class StrongTyping {
3
4     public static void main(String[] args) {
5
6         int numberOne = 4;          //static typing
7         int numberTwo;
8
9         numberTwo = (int) 4.6;
10
11    }
12
13 }
```

Casting resolves
the type
mismatch error

Strong Typing – Example 1 (fix with type)



Changing type
to double from
int.



A screenshot of a Java code editor showing the file `StrongTyping.java`. The code demonstrates static typing by changing the variable type from `int` to `double`.

```
1
2 public class StrongTyping {
3
4     public static void main(String[] args) {
5
6         int numberOne = 4;          //static typing
7         double numberTwo;
8
9         numberTwo = 4.6;
10
11    }
12
13 }
```

The code editor interface includes a title bar with the file name, a code area with syntax highlighting, and a vertical scroll bar on the left. A large grey arrow points from the explanatory text above towards the code editor window.

Strong Typing – Example 2



```
StrongTyping.java
1
2 public class StrongTyping {
3
4     public static void main(String[] args) {
5
6         int a = 4;          //static typing
7         String b = "8";   //static typing
8
9         System.out.print("a * b gives: ");
10        System.out.print(a *| b);
11    }
12
13 }
14
```

The code shows a Java program named `StrongTyping.java`. It contains a `main` method that prints the result of multiplying the integer `a` by the string `b`. A tooltip appears over the multiplication operator `*`, indicating a type error: `The operator * is undefined for the argument type(s) int, String`.

Weak Typing

“Variables are not of a specific data type.

However it doesn’t mean that variables are not “bound” to a specific data type.

In weakly typed languages, once a block of memory is associated with an object it can be reinterpreted as a different type of object.”

Weak Typing – Example 1



codingground SIMPLY EASY CODING | Online Javascript Editor (Javascript)

Preview | Embed index.htm

```
i 1 <html>
2 <script>
3 function weakTyping() {
4     var a = 4;          //dynamic typing
5     var b = '8';        //dynamic typing
6     document.write("a + b gives: ");
7     document.write(a + b); //weak typing
8 }
9 weakTyping();
10 </script>
11 </html>
```

Result
a + b gives: 48

Weak Typing – Example 2



codingground SIMPLY EASY CODING | Online Javascript Editor (Javascript)

Preview | Embed index.htm

```
i 1 <html>
2 <script>
3 function weakTyping() {
4     var a = 4;          //dynamic typing
5     var b = '8';        //dynamic typing
6     document.write("a * b gives: ");
7     document.write(a * b); //weak typing
8 }
9 weakTyping();
10 </script>
11 </html>
```

Result
a * b gives: 32

Weak Typing – Example 3



Online Javascript Editor (Javascript)

Preview | Embed index.htm

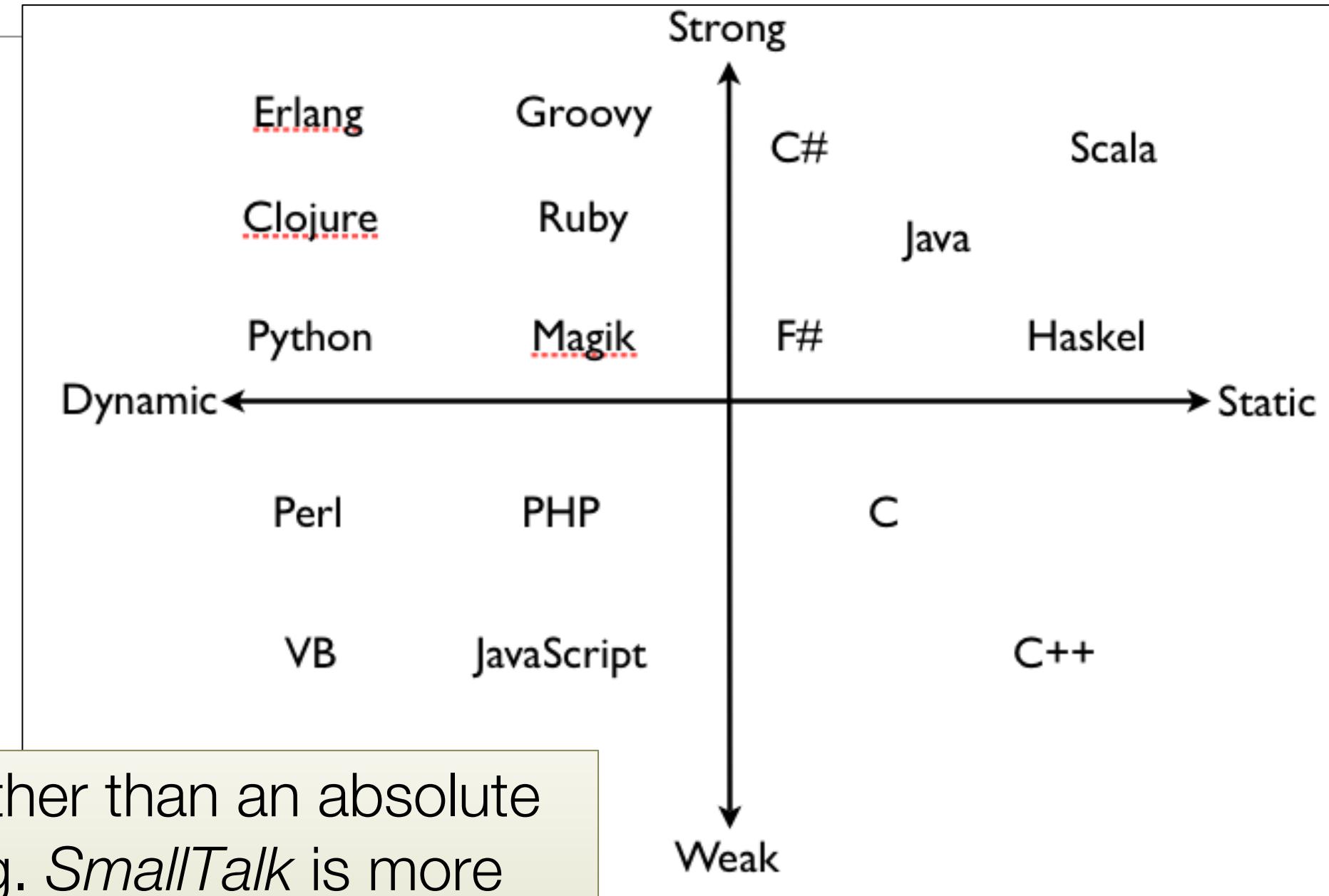
```
i 1 <html>
2 <script>
3 function weakTyping() {
4     var a = 4;          //dynamic typing
5     var b = true;       //dynamic typing
6     document.write("a + b gives: ");
7     document.write(a + b); //weak typing
8 }
9 weakTyping();
10 </script>
11 </html>
```

Result
a + b gives: 5



How the runtime constrains you from treating
objects of different types (in other words
treating memory as blobs or specific data types)

Typing Spectrum



Continuum rather than an absolute measure; e.g. SmallTalk is more strongly typed compared to Python which is more strongly typed than JavaScript.