

Learn you ALL Programming Languagesin 30 minutes!

By Wally

Which languages should I use ?

C/C++, Java, Python, Scala Haskell...





Understanding the question

- 01 New programmer: which language should I use to learn programming
- 02 Experienced programmer: which language should I use for a project / for my team.
- 03 A new programming language just come out, should I learn it?

Answer?



Answer ?

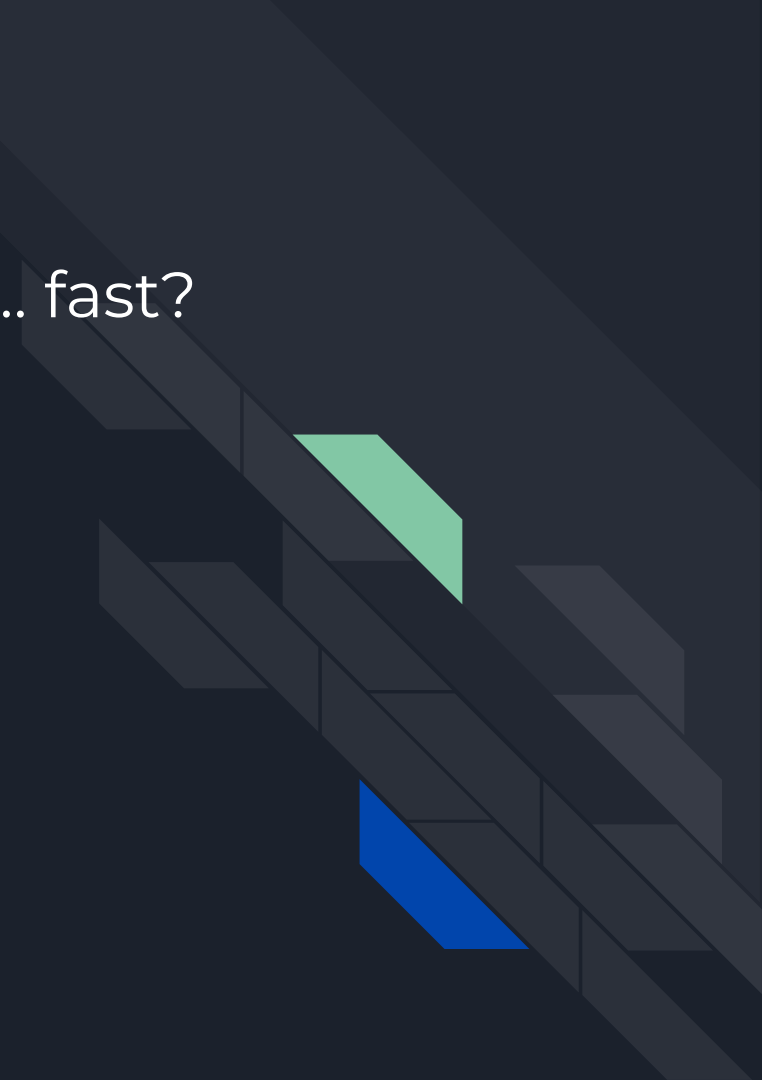
It depends!



How to learn a new language... fast?



How to learn all languages... fast?





Language Features vs. Languages

- Dell XPS 15
- Intel® Core™ i7-8750H
- 32GB DDR4-2666MHz
- NVIDIA® GeForce® GTX 1050Ti
- ThinkPad X1C
- Intel® Core™ i7-8650U
- 16GB LPDDR3 2133MHz
- Integrated Intel® UHD Graphics 620



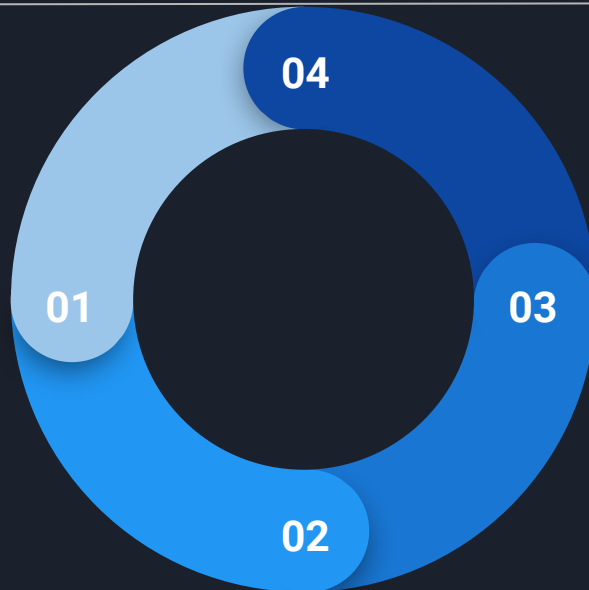
Language Features

Variable & Types

Static vs. Dynamic, Strong
vs. Weak, Type inference...

Control Flow

Condition statements,
Loop statements



Other

Community, Toolchain,
License

Structure

Functions, Classes,
Memory Management

Languages Comparison



The image displays three Vim editor windows side-by-side, each showing a different programming language implementation of a 'Hello' program. The windows are titled 'vim hello.cpp', 'vim Hello.java', and 'vim Main.scala'. Each window has a red header bar with the filename and dimensions. The code is syntax-highlighted, and the Vim interface elements like line numbers and status bars are visible.

```
vim hello.cpp 44x25
1 #include <iostream>
2
3 using namespace std;
4
5 int main(void)
6 {
7     int x = 10;
8     if (x > 3) {
9         for (int i = 0; i < 10; i++) {
10             cout << "HELLO" << endl;
11         }
12     }
13
14     return 0;
15 }
```

1,13 All

```
vim Hello.java 53x25
1 public class Hello {
2
3     public static void main(String[] args) {
4         int x = 10;
5         if (x > 3) {
6             for (int i = 0; i < 10; i++) {
7                 System.out.println("HELLO");
8             }
9         }
10     }
11 }
```


10,1 All

```
vim Main.scala 34x17
1 object Main extends App {
2     val x: Int = 10;
3     if (x > 3) {
4         for (i <- 0 until 10) {
5             println("HELLO");
6         }
7     }
8 }
```

1,3 All



Special Language Features

- C++: Multiple Paradigm, High Performance
 - Rust: Ownership memory management, zero-cost abstraction
 - Go: Goroutine, high concurrency
 - Haskell: First-class and higher-order functions
 - Python: Quick scripting & deploying
- 



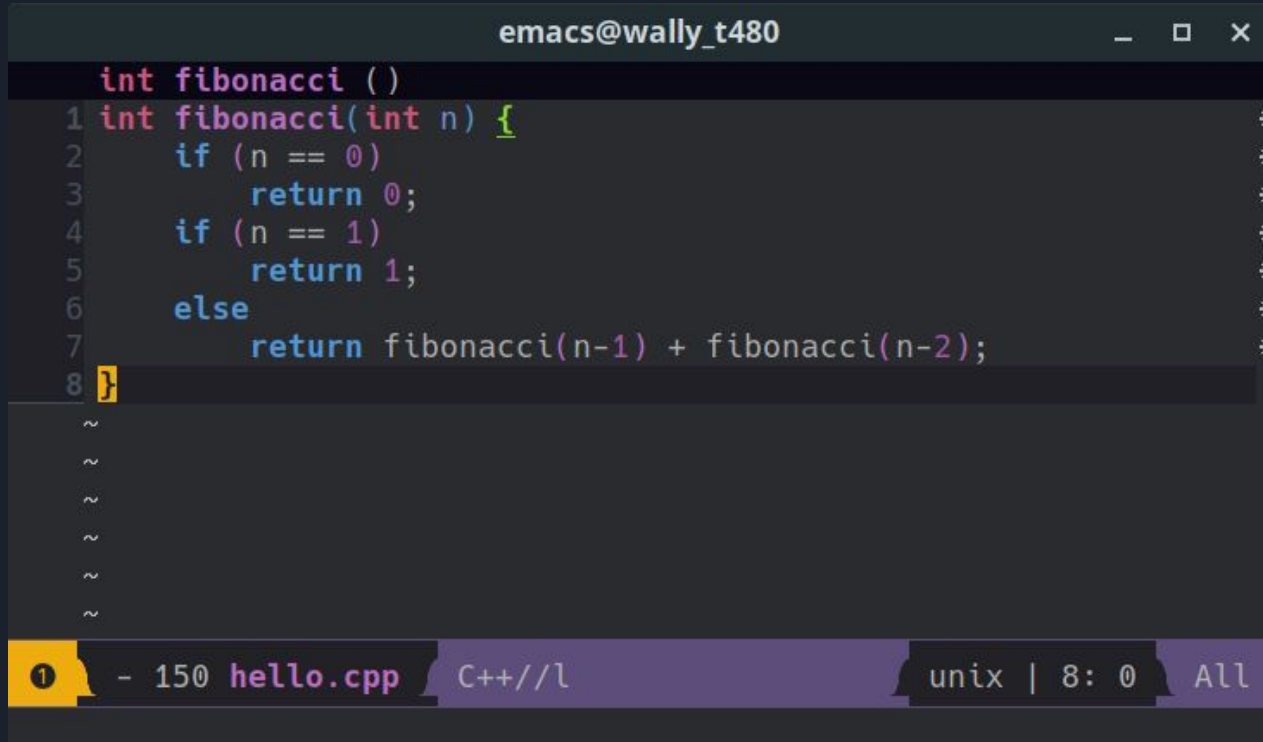
Programming Paradigm

- Procedural Programming
- Object-Oriented Programming
- Functional Programming*

Functional Programming

```
emacs@wally_t480  _  □  ×  
1 fibonacci 0 = 0  
2 fibonacci 1 = 1  
3 fibonacci n = fibonacci (n-1) + fibonacci (n-2)  
4 main = putStrLn (show (fibonacci 10))  
~  
~  
① - 118 hello.hs Haskell unix | 3:36 All  
Wrote /home/wally/temp/hello.hs
```

Procedural Programming



The screenshot shows an Emacs editor window titled "emacs@wally_t480". The code is a C++ function named "fibonacci" that calculates the nth Fibonacci number. The function is defined as follows:

```
int fibonacci ()  
1 int fibonacci(int n) {  
2     if (n == 0)  
3         return 0;  
4     if (n == 1)  
5         return 1;  
6     else  
7         return fibonacci(n-1) + fibonacci(n-2);  
8 }  
~  
~  
~  
~  
~  
~
```

The status bar at the bottom indicates the file is "hello.cpp", the language is "C++//l", and the mode is "unix | 8: 0".

Function as variable

```
emacs@wally_t480
int fibonacci ()
1 int fibonacci(int n) {
2     if (n == 0)
3         return 0;
4     if (n == 1)
5         return 1;
6     else
7         return fibonacci(n-1) + fibonacci(n-2);
8 }

10 void printWith10(function func) {
11     cout << func(10) << endl;
12 }

14 int main(void) {
15     printWith10(fibonacci);
16     auto f = [](int a) -> a + 10;
17     printWith10(f)
18 }
~|

1 - 318 hello.cpp C++//l (a)(y)(f)(p)(K) unix | 19: 0 All
End of buffer
```

The background is a dark blue-grey gradient. In the top-left corner, there are two overlapping geometric shapes: a blue parallelogram and a light green parallelogram. In the bottom-left corner, there is a circular inset showing a detailed, high-contrast image of a printed circuit board (PCB) with various electronic components. In the top-right corner, there is a faint, grey, 3D-rendered pattern of interlocking cubes or a circuit board layout.

Questions?