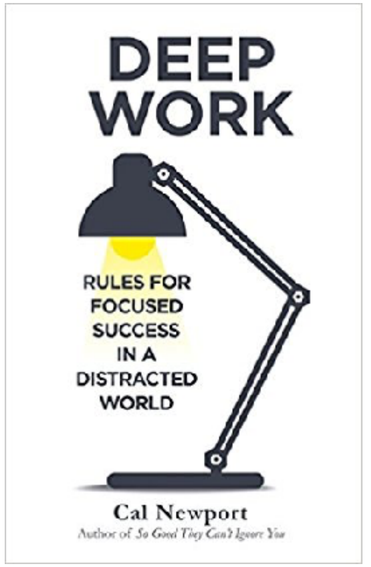
Introduction to Programming

“The complex reality of the technologies that real companies leverage to get ahead emphasizes the absurdity of the now common idea that exposure to simplistic, consumer-facing products – especially in schools – somehow prepares people to succeed in a high-tech economy. Giving students iPads or allowing them to film homework assignments on YouTube prepares them for a high-tech economy about as much as playing Hot Wheels would prepare them to thrive as auto mechanics.”

Deep Work
Cal Newport



Lesson 1

Programming (and learning it) requires long stretches of uninterrupted time.

Hello World

A look at four different programming languages

Hello World -


```java
public class HelloWorld {

  public static void main(String[] args) {
    System.out.println("Hello, World");
  }

}
```

Hello World - C

```c
#include <stdio.h>
int main()
{
    printf("Hello, World!");
    return 0;
}
```

Hello World - Python

```
1 print("Hello World!")
```

Hello World - Ruby

```ruby
puts "Hello World!"
```

GitHub

Basecamp

airbnb

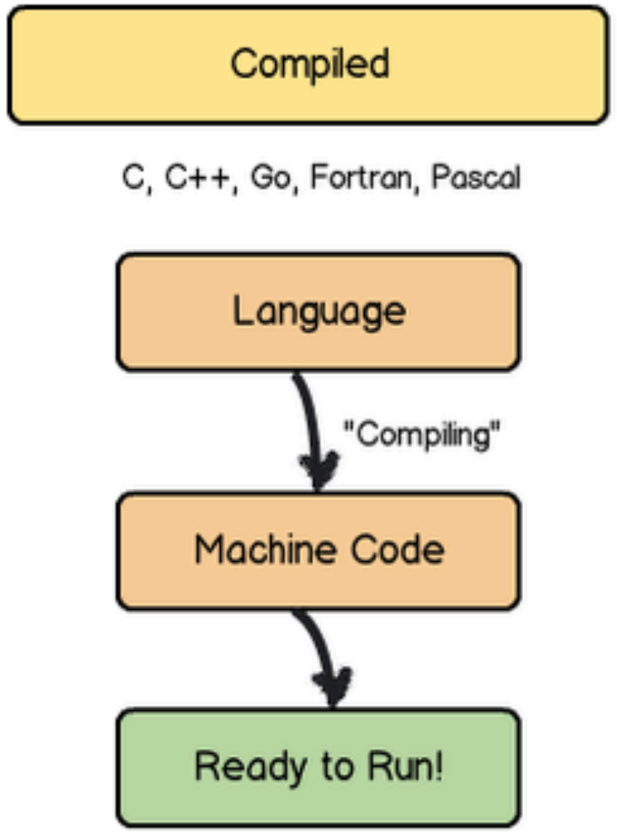
KICKSTARTER

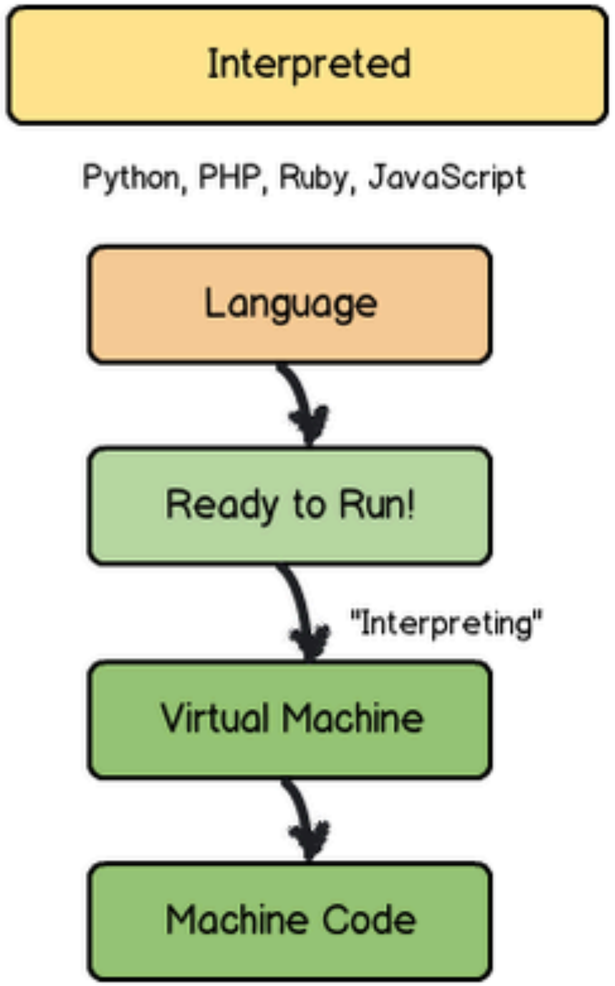
shopify

Lesson 2

You can build anything with any programming language.

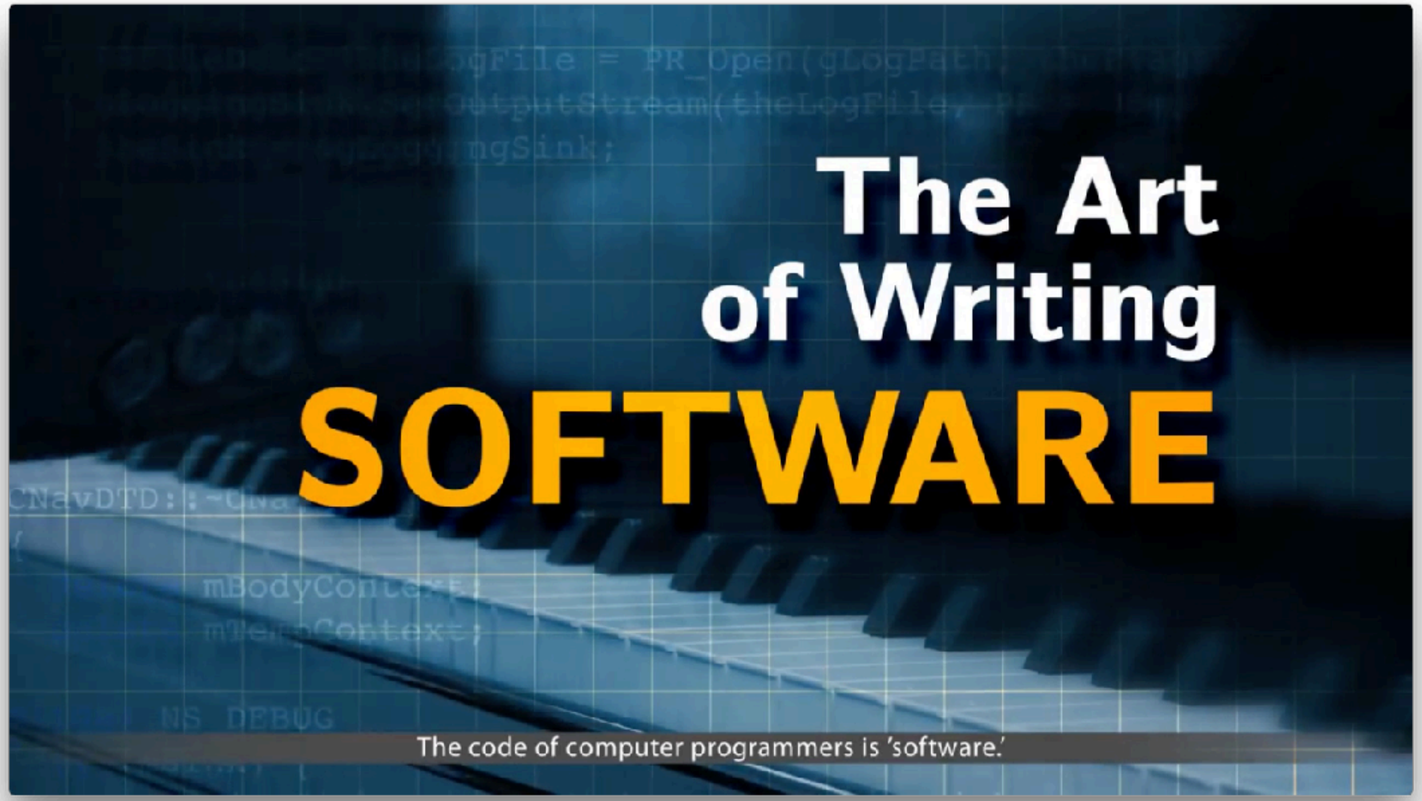
Compiled or Interpreted





0110001 0110010 0010000 0110100 0110010 0010000 0110100 0110
0110

The art of writing Software



https://www.youtube.com/watch?v=QdVFvsCWXrA

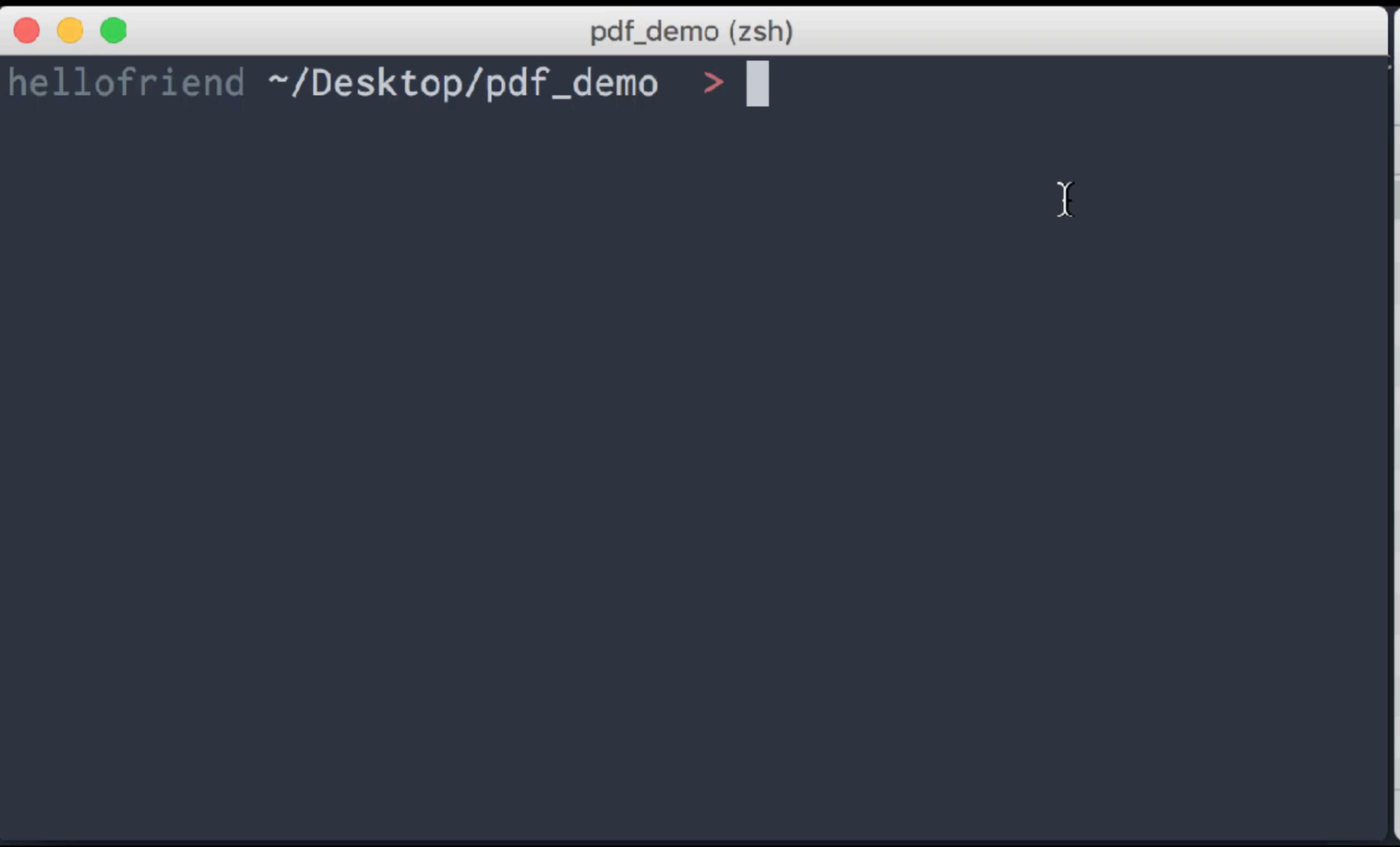
How programming can help you

pdf_renamer.rb

```ruby
require 'pdf-reader'
require 'date'

def get_name(pdf_filename)
  reader = PDF::Reader.new(pdf_filename)
  text = reader.pages.first.text.split(" ")
  name = text[5] + " " + text[6]
end

def rename_pdf(original_file, name)
  current_month = "#{Date.today.year}_#{Date.today.month}"
  File.rename(original_file, "#{current_month}_#{name}.pdf")
end

pdfs = Dir["./*.pdf"]

pdfs.each do |pdf|
  employee_name = get_name(pdf).downcase.gsub(" ", "_")
  rename_pdf(pdf, employee_name)
end
```

~

~


pdf_demo (zsh)
hellofriend ~/Desktop/pdf_demo >
pdf_demo
Name
Date Modified
234209423742.pdf
Today, 17:28
pdf_renamer.rb
Today, 17:38
1 of 2 selected, 142.22 GB available

Ruby

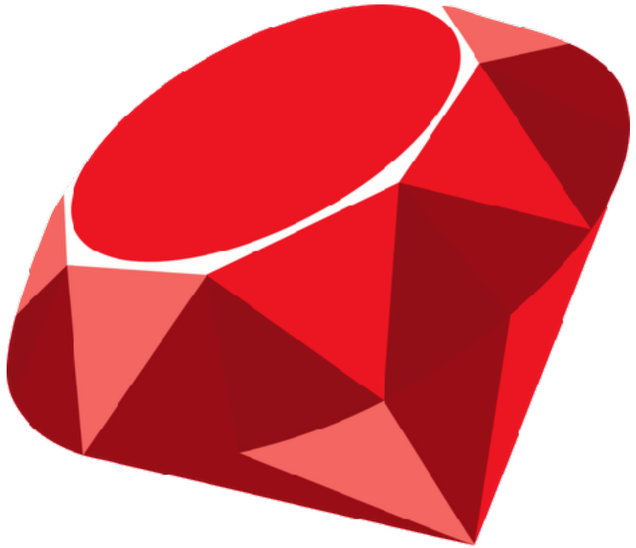
Good programming is like good writing.

Lesson 3

Good programming is like good writing.

Ruby design goals

- Focus on the **human factor**
- Principle of **Least Surprise**
- Principle of **Least Effort**
- In short: **Focus on Programmer Happiness**



An example

Read the following aloud to yourself.

```
5.times { print "Odelay!" }
```

Read the following aloud to yourself.

```
exit unless "restaurant".include? "aura"
```

Read the following aloud to yourself.

```
['toast', 'cheese', 'wine'].each { |food| print food.capitalize  }
```

Lesson 4

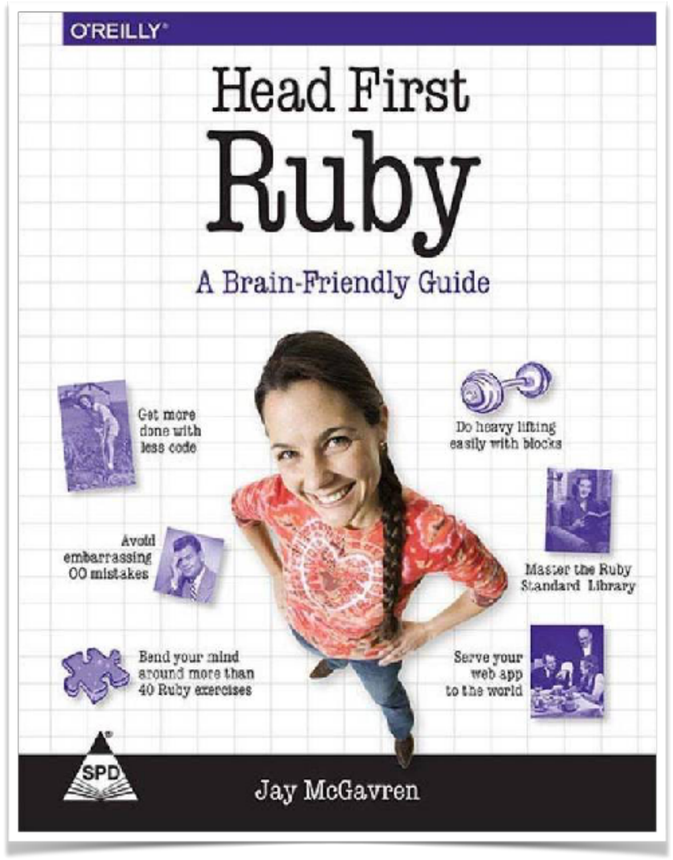
Make sure you like the tools that you work with.

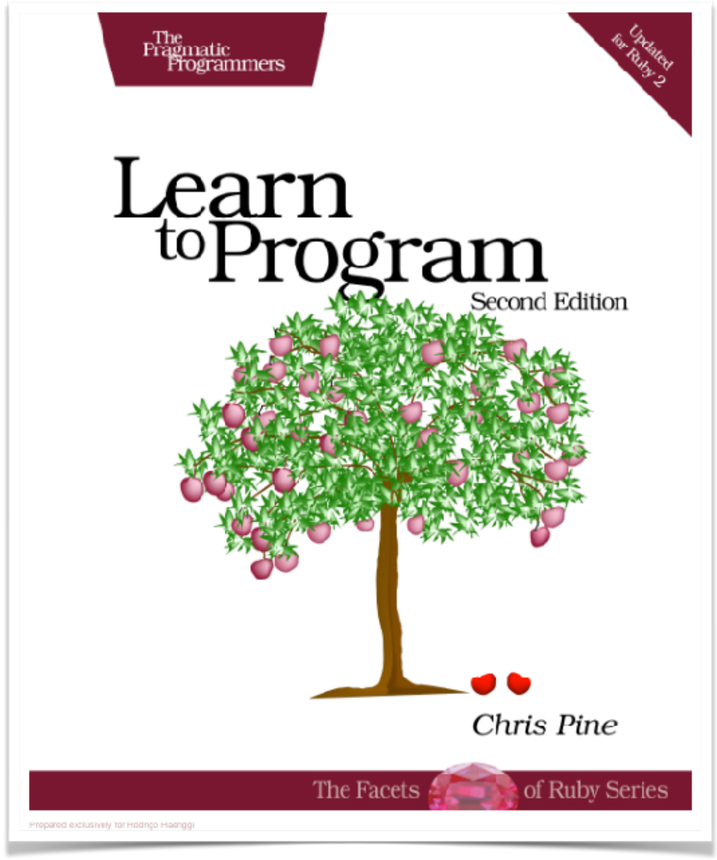
Ruby basics

Running your first program 🏃‍♀️

Your first Ruby expressions

Books


O'REILLY
Head First
Ruby
A Brain-Friendly Guide
Get more done with less code
Do heavy lifting easily with blocks
Avoid embarrassing OO mistakes
Master the Ruby Standard Library
Bend your mind around more than 40 Ruby exercises
Serve your web app to the world
SPD
Jay McGavren


The Pragmatic Programmers
Updated for Ruby 2
Learn to Program
Second Edition
Chris Pine
The Facets of Ruby Series