

# Reading and Writing Data with Ruby

---

## WORKING WITH FILES



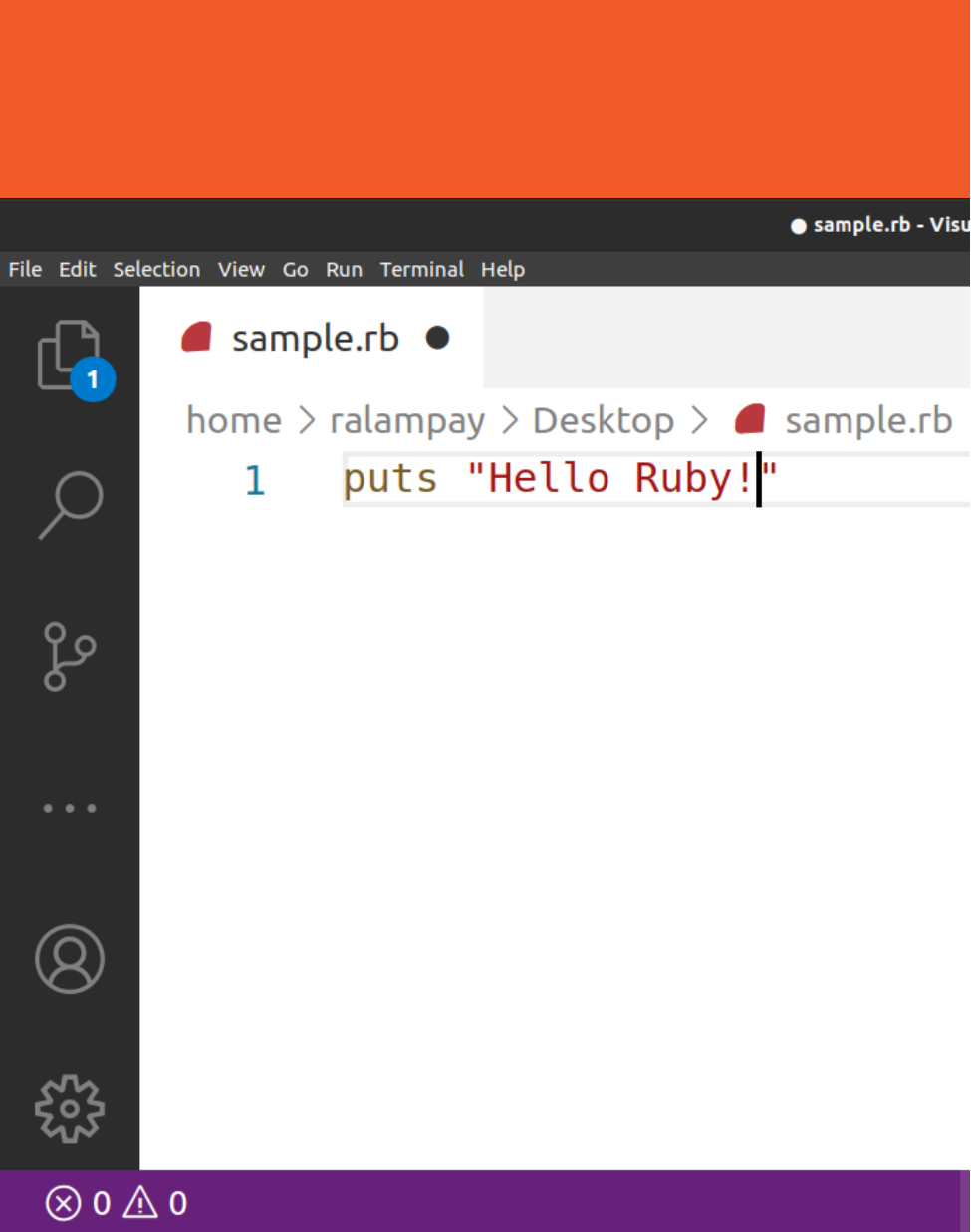
**Raphael Alampay**

DEVELOPER

@happyalampay [github.com/ralampay](https://github.com/ralampay)

# Development Setup

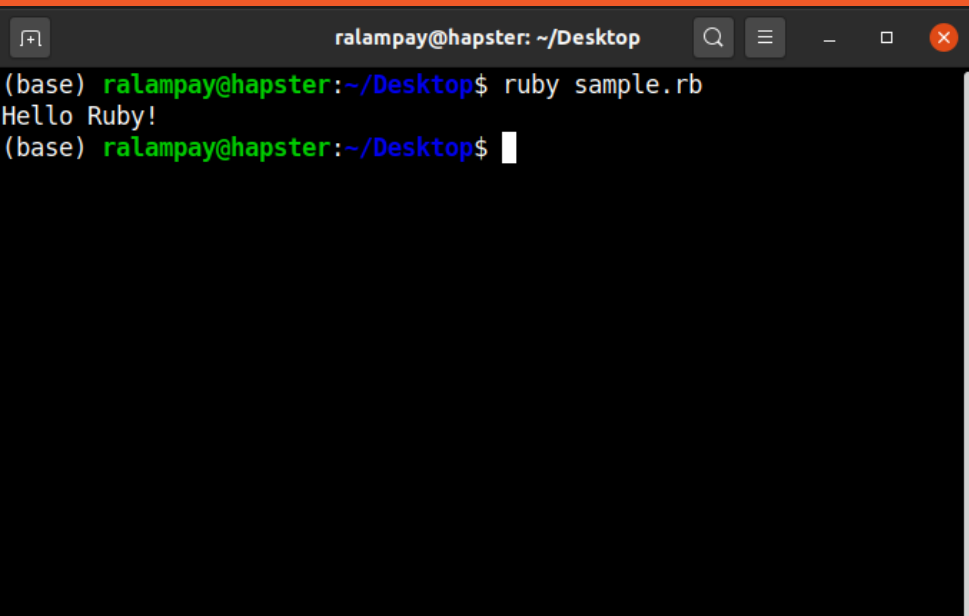
Our tools of the trade



## Text Editor to Write Code In

### Recommended Editors

- Visual Studio Code
- Atom Text Editor
- Sublime Text Editor

A terminal window with a dark background and a title bar. The title bar contains the text 'ralampay@hapster: ~/Desktop' and standard window control icons. The terminal shows a prompt '(base) ralampay@hapster:~/Desktop\$' followed by the command 'ruby sample.rb'. The output 'Hello Ruby!' is displayed on the next line. A second prompt '(base) ralampay@hapster:~/Desktop\$' is shown with a cursor, indicating the command has finished.

```
(base) ralampay@hapster:~/Desktop$ ruby sample.rb
Hello Ruby!
(base) ralampay@hapster:~/Desktop$
```

Terminal to run Ruby scripts

Files should be in the same directory

# The Main Program

A single file to run our examples

**main.rb**

```
require 'csv'  
require 'json'  
require 'net/http'  
puts "Hello Ruby!"
```

# Reading a File

---

# Reading a File (Verbose)

## Location of File

Use a variable to represent the path or location of a file

## Opening a File

Invoke the `File.open()` method which returns a reference to the file

## Extract Contents

Returns the contents of a file as a string to be stored in a variable

```
file_location = "sample.txt"  
file = File.open(file_location)  
content = file.read  
puts content
```

- ◀ Variable representing the file location
- ◀ Open the file and return a reference
- ◀ Extract contents of file
- ◀ Display content



```
file_location = "sample.txt"  
content = File.read(file_location)  
puts content
```

## Reading a File (Short and Sweet)

**A one-liner to extract contents of a file**

# Demo



Read a file from a location

Print out contents of the file

# Writing to a File

---

# Approaches to Writing to a File

## Replace

**Completely overwrites a file**

## Append

**Adds content to a file**

# Methods for Performing a Write

## Default

Overwrite a file completely with the `write()` method

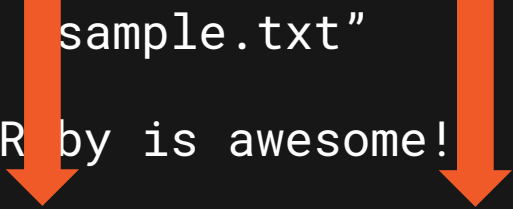
## Pass Mode Value

Pass the mode symbol with value 'a' to perform an append

## Code Block

Pass an anonymous function with `File.open()` method


```
file_location = "sample.txt"  
new_content = "Ruby is awesome!"  
File.write(file_location, new_content)
```



## Writing to a File (Default)

**Performs an overwrite to the file or creates the file if it doesn't exist**

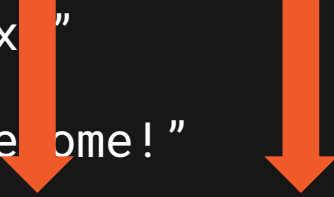
```
file_location = "sample.txt"  
new_content = "Ruby is awesome!"  
File.write(file_location, new_content, mode: 'a')
```



## Writing to a File (Pass Mode Value)

**Appends content to the existing file**

```
file_location = "sample.txt"
new_content = "Ruby is awesome!"
File.open(file_location, "a") do |file|
  file.write(new_content)
end
```



## Writing to a File (Code Block)

**Open a file and pass an anonymous function to write to the file reference**



# Demo



Write a file using the default approach

Perform an append

Append more content using a code block