

# RUBY QUICKSTART BOOTCAMP

**PLEASE FEEL FREE TO INTERRUPT AND ASK QUESTIONS!**

Pulled from the Ruby Quickstart

<https://www.ruby-lang.org/en/documentation/quickstart/>

Getting more help

<http://apidock.com/ruby>

# QUICK INTRODUCTION

Ruby comes with a program that you can use to run code "on the fly". It's called irb and stands for Interactive Ruby

If you're using Windows open Interactive Ruby from the ruby section of the start menu.

If you're using Mac or Linux open a terminal and type  
irb

# IRB AS A CALCULATOR

## 1. Addition

$$3+2$$

## 2. Multiplication

$$3*2$$

## 1. Powers

```
3**2
```

## 2. Square Root

```
Math.sqrt(9)
```

## BREAKING IT DOWN -

```
Math.sqrt(9)
```

Math is a built-in module for mathematics. Modules serve two roles in Ruby. This shows one role about how to group similar methods together under a familiar name. Math also contains methods like `sin()` and `tan()`

## BREAKING IT DOWN

```
Math.sqrt(9)
```

Dot (.) is how you identify the receiver of a message. The message in this case is `sqrt(9)`, which means call the method `sqrt` (short for square root) with the parameter of 9

## BREAKING IT DOWN

The result of this is the value 3.0, notice it's not just 3.



# SAVING RESULTS (VARIABLES)

```
a = 3 ** 2
```

```
b = 4 ** 2
```

```
Math.sqrt(a+b)
```

# HELLO WORLD!

```
irb(main):001:0> "Hello World"  
=> "Hello World"
```

# HELLO WORLD!

```
irb(main):002:0> puts "Hello World"
```

```
Hello World
```

```
=> nil
```

# MAKING IT OUR OWN METHOD

```
def hi  
  puts "Hello World!"  
end  
hi  
hi()
```

# SAYING HI TO SOMEONE

Adding a parameter to the method

```
def hi(name)
  puts "Hello #{name}!"
end
hi("Will")
hi # error!
```

## WHAT IS THAT "#{NAME}" BIT?

```
def hi(name)
  puts "Hello #{name}!"
end
```

## MAKING IT A LITTLE BIT SMARTER

```
def hi(name = "World")  
  puts "Hello #{name.capitalize}!"  
end  
hi "will"  
hi
```

# INSPECTING THINGS IN IRB

```
name = "will"  
name.inspect  
name.methods  
name.methods.sort  
name.capitalize  
name.upcase  
name.upcase!  
name
```



# ARRAYS

```
a = [1, 2, 3, 4, 5]
```

```
a.size
```

```
a.max
```

```
a.reverse
```

```
a.first
```

```
a.last
```

# LOOPING

```
a = [1,2,3,4,5]  
a.each do |number|  
  puts number  
end
```

1

2

3

4

5

# MORE LOOPING

```
5.times do |number|  
  puts number  
end
```

0

1

2

3

4

How can we make the output look like the first loop?

# MAKING IT LOOK LIKE THE FIRST LOOP

```
5.times do |number|  
  puts number + 1  
end
```