

### How to print something out on the screen

You can use "print". For e.g., print "Hello World"

You can use "puts". For e.g., puts "Hello World"

When you use print, everything gets printed out on the same line

When you use puts, everything gets printed out on a new line

### Variables

Let's say we want to print out the message:

```
I am 18 years old. I like ice skating. I went ice skating for the first time when I was 18.
```

You can create variables:

```
character_age = " 18 "
```

```
character_sport = "ice skating"
```

To print this message, write:

```
character_age = " 18 "
```

```
character_sport = "ice skating"
```

```
puts ("I am " + character_age + " years old.")
```

```
puts ("I like " + character_sport)
```

```
puts ("I went " + character_sport + " for the first time when I was " + character_age)
```

You can also modify the message half way:

```
character_age = " 18 "
```

```
character_sport = "ice skating"
```

```
puts ("I am " + character_age + " years old.")
```

```
puts ("I like " + character_sport)
```

```
character_sport = " cycling"
```

```
puts ("I went " + character_sport + " for the first time when I was " + character_age)
```

It will output:

```
I am 18 years old. I like ice skating. I went cycling for the first time when I was 18.
```

### Data Types

String /Text

```
name = " Mike "
```

Integer (A full number)

```
age = 18. Note that you do not
```

need to include quotations.

Floating Point (A number that

has a decimal)

```
number = 18.9
```

Boolean (on/off, true/false,

```
1/0)
```

```
isfemale = true
```

Nil (referring to " nothing"

in "g". Absence of a value)

```
flaws = nil
```

### What does gets.chomp do?

```
puts "Enter your name"
```

```
name = gets
```

```
puts ("Hello " + name + " , you are cool!")
```

" gets" alone will put " , you are cool!" on a new line.

To mitigate this, we use gets.chomp to ensure that

```
" Hello " + name + " , you are cool!" is on one line.
```

### How to print a new line?

```
print " Hello\n World"
```

It will output:

```
Hello
```

```
World
```

### What is Array.new?

```
friends = Array.new
```

### How to calculate the length of a string?

```
phrase = "Giraffe Academy"
```

```
puts phrase.length()
```

It will output: 15

### How to get a range of text?

```
phrase = "Giraffe Academy"
```

```
puts phrase [0,3]
```

It will output: Gir

### How to find the index position of a string

```
phrase = "Giraffe Academy"
```

```
puts phrase.index ("ffe")
```

It will output the index

position it starts at: 4

### How to make a string uppercase/lower-case?

```
def
write_data_to_file(file_name)
  a_file.puts('5')
  a_file.puts('Fred')
  a_file.puts('Sam')
  a_file.puts('Jill')
  a_file.puts('Jenny')
  a_file.puts('Zorro')
  a_file.close()
end
```

The close() is an inbuilt function in Ruby.

**Parameters:** The function does not take any element.

**Return Value:** It closes the queue and does not return anything.

### Arrays

```
friends = Array["Kevin",
"Karen", "Oscar"]
puts friends[-2]
It will output: Karen
If you want to select names from
index 0 - 2:
friends = Array["Kevin", "Karen", "Oscar"]
puts friends[0, 2]
It will output: Kevin, Karen.
Note that it does
not include Oscar.
```

### Typing input into the terminal

```
puts "Enter your name: "
name = gets.chomp()
puts "Enter your age: "
age = gets.chomp()
puts ("Hello " + name + ", you
are" + age + " years old")
```

Here we're basically telling Ruby that we want friends to be an array but we don't want to put any new values into it just yet

```
friends[0] = "Molly"
```

```
puts friends[0]
```

### Calculating age when inputting year of birth

```
require 'date'
def hello()
puts("what year were you born?")
year_born = gets.chomp()
age = Date.today.year -
year_born.to_i() - 1
puts("So you are " + age.to_s +
" years old")
end
def
main()
hello()
end
main()
```

### How to print out a quotation mark in a string

```
puts "Giraffe" Academy"
```

It will output: Giraffe" Academy

### How to delete extra trailing and whitespace?

```
phrase = " Giraffe Academy "
puts phrase.strip()
It will output " Giraffe
Academy"
```

```
phrase = " Giraffe Academy "
puts phrase.upcase()
puts phrase.downcase()
It will output:
GIRAFFE ACADEMY
giraffe academy
```

### What is a class?

```
class Book
  attr_accessor :title,
:author, :pages
end
book1 = Book.new()
book1.title = " Harry Potter "
book1.author = "JK Rowling"
book1.pages = 400
puts book1
```

This class is a template or blueprint for representing a Book inside of the program. Essentially it's like creating your own Data Type.



By ragami  
[cheatography.com/ragami/](http://cheatography.com/ragami/)

Not published yet.  
 Last updated 21st March, 2023.  
 Page 2 of 2.

Sponsored by [CrosswordCheats.com](http://CrosswordCheats.com)  
 Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>