## 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:
charac  te r_age = "  18 "
charac  te r _ sport = "ice skatin  g"
To print this message, write:

charac  te r_age = "  18 "
charac  te r _ sport = "ice skatin  g"
puts ("I am " + charac  te r_age + " years old.")
puts ("I like " + charac  te r _ sport)
puts ("I went " + charac  te r _ sport + " for the first time when I was " + charac  te r _age)

You can also modify the message half way:
charac  te r_age = "  18 "
charac  te r _ sport = "ice skatin  g"
puts ("I am " + charac  te r_age + " years old.")
puts ("I like " + charac  te r _ sport)
charac  te r _ sport = "  cy c l in  g"
puts ("I went " + charac  te r _ sport + " for the first time when I was " + charac  te r _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 = "  Mi k e "
Integer (A full number)
age = 18. Note that you do not
need to include quotat  ions.
Floating Point (A number that
has a decimal)
number = 18.9
Boolean (on/off, true/f  alse,
1/0)
isfemale = true
Nil (referring to "  no t h -
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!")
" get s" 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.le ngth()
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.in dex ("ff e")
It will output the index
position it starts at: 4
```

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

## What does .close() do?

```
def
write_data_to_file(file_name)
     a_f ile.pu ts('5')
     a_f ile.pu ts( 'Fred')
     a_f ile.pu ts( 'Sam')
     a_f ile.pu ts( 'Jill')
     a_f ile.pu ts( 'Je nny')
     a_f ile.pu ts( 'Zo rro')
     a_f ile.cl ose()
end
```

The close() is an inbuilt function in Ruby.

Parameters: The function does not takes any element.
Return Value: It closes the queue and does not returns anything.

## Arrays

```
friends = Array["Kevin",
"Karen", "Oscar"]
puts friend s[-2]
It will output: Karen
If you want to select names from
index 0 - 2:
friends = Array[ " Kev in", " -
Kar en", " Osc ar"]
puts friends[0, 2]
It will output: Kevin, Karen.
Note that is does
not include Oscar.
```

## Typing input into the terminal

```
puts "Enter your name: "
name = gets.c homp()
puts " Enter your age: "
age = gets.c homp()
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.c homp()
age = Date.t oda y.year -
year_b orn.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.st rip()
It will output "  Gi raffe
Academ  y"
```

```
phrase = " Giraffe Academ y"
puts phrase.up  case()
puts phrase.do  wn c ase()
It will output:
GIRAFFE ACADEMY
giraffe academy
```

## What is a class?

```
class Book
     att r_a ccessor :title,
:author, :pages
end
book1 = Book.new()
book1.t itle = " Harry Potter "
book1.a uthor = "JK Rowlin g"
book1.p ages = 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.