

Learn to Program

Lesson 2

puts, gets, numbers and strings



irb

- irb (Interactive Ruby) is an excellent way to test things.
- From the command line type: irb
- Then you can write ruby code and see it evaluated right away.

puts

Type irb from the command line

```
>puts 3
```

```
> => 3
```

puts prints the string to the default output which is the screen.

gets

- gets is the inverse of puts
- gets waits for the user to type letters and type return

```
entry = gets  
puts entry
```


Exercise

- Full name greeting

- Write a program that asks for a person's first name, then middle then last name. Finally the program should greet them with their full name "Hello <full name>"

- Bigger Better Favorite Number

- Write a program that asks for a favorite number.
- Add one to the number and present them with the new number

Numbers

- Ruby understands numbers too.
- Integers are whole numbers (1,2,3,4, etc)
- Floats are numbers with decimals (1.0, 2.56, etc).
- $1*2$ means one times two in ruby.
- $1.0/2.0$ means one divided by 2.

Parenthesis

- For math, you use parenthesis to separate the parts of equations you want evaluated first.
- $(1.0+34.0)/23.0$
- In the example above, the $1 + 34$ take place first.

Exercise

Simple Interest Calculator

- Write a program that:
 - Asks for the amount to be put in the bank
 - Asks for the number of years to be saved (period)
 - Asks for the percent interest paid
 - The program then returns the total value at the end of the period, with interest

Flow Control

- In ruby you can get a true or false answer when you compare numbers, in irb try this:
- `puts 1>3`
- `puts 4<=2`
- `puts 1==1`
- `puts 4!=5`

Comparison

- You can compare strings, try:
- `puts "HELLO".downcase == "hello"`

if

- You can use “if” to compare an expression, and then if the expression is true, you can do something else after:

```
puts “What is your name?”
```

```
name = gets
```

```
if name == “Chris”
```

```
    puts “hello” + name
```

```
else
```

```
    puts “I don’t know you”
```

```
end
```


while

- If you want to do something over and over again, you can use “while” and tell it when to stop:

```
input = ''
```

```
while input != "stop"
```

```
    input = gets.chomp
```

```
    puts input
```

```
end
```

- Or something that runs forever

```
while true
```

```
    puts "hello"
```

```
end
```


Exercise

(using if and while)

- write a program to guess the number that the computer is thinking
- the computer will ask you what the number is
- you have to enter any number and then the computer will tell you if this number is bigger or less than the computer's number
- keep entering a number until you get the correct one