



# Ruby 101

## Ruby Basics: Methods

Methods:

perform a collection of actions  
abstract program logic into chunks  
keep code from repeating

Ruby Basics:



Parts of a Method Definition:

```
def method_name  
  ...  
end
```

Ruby Basics:



## Method Example

```
def hello_user
  puts "Enter your name"
  username = gets
  puts "Hello " + username
end
```

Calling the Method:

```
hello_user
```



## Ruby Basics:

Methods return objects:

```
def method_name
  ...
  returned_object
end

def hello_user
  puts "Enter your name"
  username = gets
  puts "Hello " + username
end
```



Modified hello\_user

```
def hello_user
  puts "Enter your name"
  username = gets
  "Hello #{username.chop}."
end
```

Ruby Basics:



Methods sometimes return Booleans

```
def hello_user
  puts "Enter your name"
  username = gets
  if username != "\n"
    "Hello #{username.chop}."
  else
    false
  end
end
```





Try it yourself:

Write a method which multiplies two arguments together. Return the value if it is positive, but if negative return false. Then use your method in a program which prompts the user for the numbers, and outputs the result. Inform the user if the result would be negative.

Ruby Basics:





## Ruby Basics:

```
def multiply_this a, b
  total = a * b
  if total < 0
    false
  else
    total
  end
end

user_input = []
puts "Input two numbers"
while user_number = gets do
  user_input[user_input.count] = user_number.to_f
  if user_input.count == 2
    break
  end
end

if result = multiply_this(user_input[0], user_input[1])
  puts result
else
  puts "Invalid Input"
end
```

