



# Ruby 101

## Ruby Basics: Loop Control

Next:

Skips the rest of the iteration and moves to the next iteration, if the condition is still true.

```
my_var = 0
while my_var < 10 do
  if my_var == 3
    my_var += 1
    next
  end
  puts my_var
  my_var += 1
end
```



## Ruby Basics:

redo:

Goes to the beginning of this iteration of the loop, whether or not the condition is still true.

```
my_var = 0
while my_var < 10 do
  puts my_var
  if my_var == 3
    my_var = 10
    redo
  end
  my_var += 1
end
```



## Ruby Basics:

break:

exit the loop immediately

puts "Type something to continue. Or nothing to quit"  
while a = gets do

    if a == "\n"  
        break

    end

    puts a

end



Try it yourself:

Write a program that works like an adding machine. As long as the user types in more numbers, continue to add them up. If the user types in an empty line, exit the program and give the total.

Ruby Basics:



## Ruby Basics:

```
total = 0
puts "Input your numbers"
while input = gets do
  if input == "\n"
    break
  end
  total = total + input.to_f
  puts "running total = #{total}"
end

puts "Total: #{total}"
```

