



Ruby 101

Ruby Basics:Conditionals Continued

Conditionals can have more than one elsif:

```
if my_var == 10
  ...
elsif my_var == 9
  ...
elsif my_var == 8
  ...
elsif my_var == 7
  ...
end
```



Ruby Basics:

Case: (AKA switch)

Simpler alternative to mutiple elsif clauses

```
case my_var
  when 10
    ...
  when 9
    ...
  when 8
    ...
  when 7
    ...
end
```



Ruby Basics:

Conditionals can test multiple conditions at once:

- Boolean AND

- &&

```
if my_var == 10 && your_var == 5
  ...
end
```

- Boolean OR

- ||

```
if my_var == 10 || your_var == 5
  ...
end
```



Ruby Basics:

Conditions can be negated:

- NOT
 - !

```
if !my_var == 10
  ...
end
```



Ruby Basics:

But we can also test for inequality

```
if my_var != 10
  ...
end
```



Ruby Basics:

We can also test strings:

```
if my_var == "Y"  
  ...  
elsif my_var == "N"  
  ...  
else  
  ...  
end
```



Try it yourself:

Write a program that asks the user a series of Yes/No questions about what objects might be sitting on his/her desk. At the end, output the results.

(Hint: The 'gets' statement will include carriage returns, "\n", at the end of each string. Be sure to take that into account when testing conditions.)



Ruby Basics:

```
puts "Hello!"
puts "Do you have a phone (Y/N)"
case gets
  when "Y\n"
    phone = true
  when "N\n"
    phone = false
end
...
puts "You have a phone." if phone
```

