
Ruby Monstas



Session 7

Agenda

- Interactive Recap
- Instance Variables
- Project: IRC Bot



Interactive Recap

Boolean Example

From the last exercise:

Find a condition with the following truth table:

A	B	result
false	false	false
true	false	true
false	true	true
true	true	false

Boolean Example

What could be the result?

Any condition with A and B in it that yields the result, e.g.:

$A \parallel B$

$A \&\& B$

$A \parallel B \&\& (\neg A \parallel B)$

$A \&\& \neg B \parallel (A \&\& B)$

A	B	result
false	false	false
true	false	true
false	true	true
true	true	false

Boolean Example

Let's take: $A \parallel B$

Desired table

A	B	result
false	false	false
true	false	true
false	true	true
true	true	false

Actual table

A	B	result
false	false	false
true	false	true
false	true	true
true	true	true

Boolean Example

So what's the result?

It's either A or B, but not both!

$(A \mid\mid B) \ \&\& \ \neg (A \ \&\& \ B)$

$(A \ \mathbf{OR} \ B) \ \mathbf{AND} \ \mathbf{NOT} (A \ \mathbf{AND} \ B)$

A	B	result
false	false	false
true	false	true
false	true	true
true	true	false

Boolean Example

Let's take it line by line

`(A || B) && !(A && B)` `(A OR B) AND NOT(A AND B)`

A	B	result	<code>(A B) && !(A && B)</code>
false	false	false	<code>(false false) && !(false && false)</code>
true	false	true	<code>(true false) && !(true && false)</code>
false	true	true	<code>(false true) && !(false && true)</code>
true	true	false	<code>(true true) && !(true && true)</code>

Boolean Example

OK, that's enough now!

Instance Variables

Instance variables

- Just like normal (local) variables
 - But: Valid outside and inside of methods
 - Start with an @-sign
-

Example

```
puts "Enter your name, then your age"  
@user_name = gets.chomp  
user_age = gets.chomp
```

```
def output_user_data(user_age)  
  puts "The user is called #{@user_name}."  
  puts "She is #{user_age} years old."  
end
```

Example

```
@counter = 0
```

```
def increase_counter
```

```
  @counter = @counter + 1
```

```
end
```

```
increase_counter
```

```
increase_counter
```

```
increase_counter
```

```
puts @counter
```

Project: IRC Bot

IRC Bot?!

What's IRC?

Internet Relay Chat

From Wikipedia, the free encyclopedia

"IRC" redirects here. For other uses, see [IRC \(disambiguation\)](#).



This article's **lead section** may not adequately **summarize** key points of its **contents**. Please consider expanding the lead to [provide an accessible overview](#) of all important aspects of the article. *(October 2012)*

Internet Relay Chat (IRC) is an [application layer](#) protocol that facilitates the transfer of messages in the form of text. The chat process works on a client/server networking model. IRC clients are computer programs that a user can install on their system. These clients communicate with chat servers to transfer messages to other clients.^[1] IRC is mainly designed for [group communication](#) in discussion forums, called [channels](#),^[2] but also allows one-on-one communication via [private messages](#)^[3] as well as [chat and data transfer](#),^[4] including [file sharing](#).^[5]

IRC

Internet Relay Chat (IRC) is an [application layer](#) protocol that facilitates the transfer of messages in the form of text. The chat process works on a client/server networking model. IRC clients are computer programs that a user can install on their system. These clients communicate with chat servers to transfer messages to other clients.^[1] IRC is mainly designed for [group communication](#) in discussion forums, called [channels](#),^[2] but also allows one-on-one communication via [private messages](#)^[3] as well as [chat and data transfer](#),^[4] including [file sharing](#).^[5]

protocol

messages

client / server networking

channels

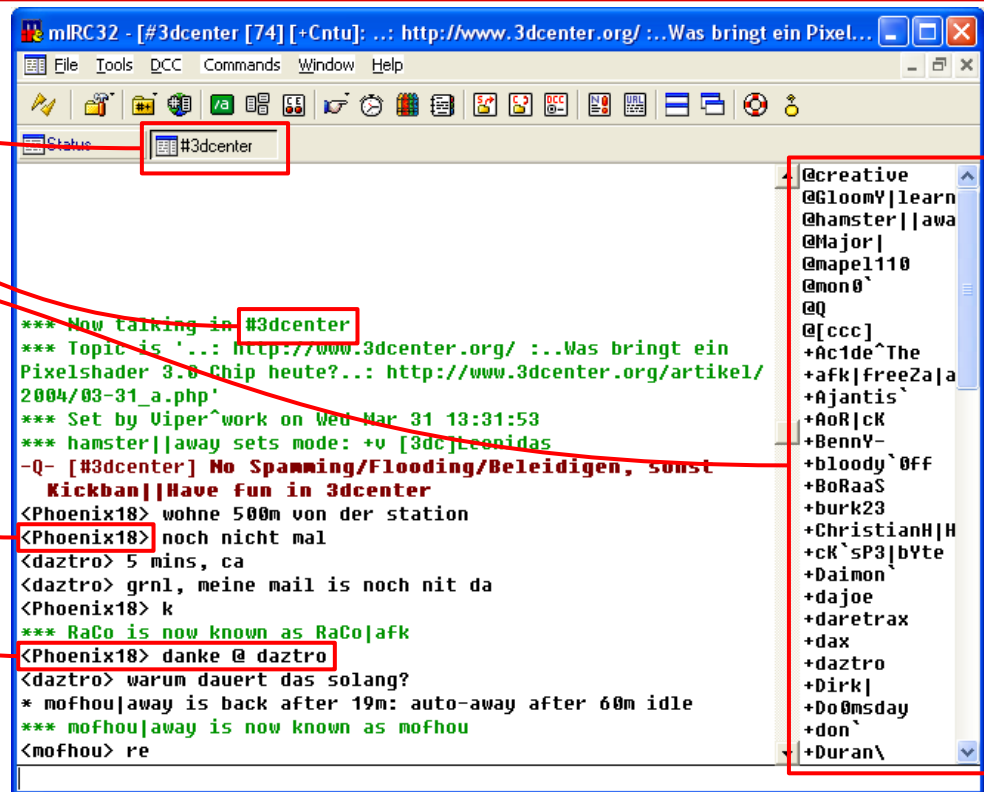
IRC

channel

user list

user

channel message



Client / Server Networking

Client



Internet (the “Cloud”)



Server



Client / Server Networking



Hey, what's up?
Let's talk!



Client / Server Networking



Sure thing, let's do it!



Client / Server Networking



OK cool.



Client / Server Networking



TCP connection



Client / Server Networking



TCP connection



Client / Server Networking

IRC clients



IRC server



IRC Protocol

How do client and server talk to each other?

There's a lot to clarify.

IRC Protocol

What could a **client** want to say?

“I want to choose a username, it should be cool_guy123”

“I want to join channel #coolcrowd”

“I want to write a message to the channel, it should be ‘Hi there!’”

What could a **server** want to say?

“There’s a new message on the channel #rubymonstas by cool_guy123, he says ‘Hi there!’”

“ruby_programmer85 just joined #rubymonstas!”

“nerdinand just left #rubymonstas!”

IRC Protocol: Client

NICK nethad-bot

USER nethad-bot 0 * :nethad-bot

JOIN #rubymonstas

PRIVMSG #rubymonstas :hi all, how are you doing?

IRC Protocol: Server

`:nethad!83.78.226.54 JOIN #rubymonstas`

`:nethad!83.78.226.54 PART #rubymonstas`

`:nethad!83.78.226.54 PRIVMSG #rubymonstas :hey, what's up?`

Bot

So what's a (IRC) bot, then?

From Wikipedia:

“An **IRC bot** is a set of scripts or an independent program that connects to Internet Relay Chat as a client, and so appears to other IRC users as another user. An IRC bot differs from a regular client in that instead of providing interactive access to IRC for a human user, it performs automated functions.”

Things a bot could do

- Let the user play a game
 - Google something and return results
 - Return information about the weather
 - Return random quotes from users
 - ... ? You name it!
-

Time to build a bot!



Let's get to it!
