

# **Regular Expressions in Cold Fusion**

**-Gypsy Rogers**

# What is RegEx?

A Regular Expression (or RegEx) is a way of describing a set of strings without having to list all the strings in your set.

Note: We are talking about strings of characters, not to be confused with guitar strings or balls of strings.

# What is Gypsy Rogers?

- Computer Geek
- Perl Programmer for 10+ years
  - Yeah so? This is ColdFusion not Perl
  - Perl is the Mecca for RegEx silly, trust me
- Your President in 2012
  - See <http://www.Gypsy2012.com>
- The guy who actually read the RegEx chapter in the CF book
- Want more? Check the Resume:  
<http://www.GypsyRogers.com/resume.html>

# Why RegEx?

- To find a sub-string in a string
- To replace a string or sub string
- To Repair a string or sub string
- To boldly go where no Pattern Matching has gone before
- Oh, and Data Validation

# Common Patterns to Match

- Email Addresses
- Phone Numbers
- Order Numbers
- Log File Entries
- Extracting Links from HTML
- Zip Codes
- Finding Partridges in Pair Trees

# Cold Fusion Functions that Make use of RegEx

- REFind
  - REFind(regex,string [,start] [,ReturnSub])
- REFindNoCase
  - REFindNoCase(regex,string [,start] [,ReturnSub])
- REReplace
  - REFind(string, regex, substring [,scope])
- REReplaceNoCase
  - Syntax:REFind(string, regex ,substring [,scope])

# Special Characters

^ -- Begin the line  
\$ -- End the line  
\* -- Match Zero or more  
+ -- Matches 1 or more  
\ -- Escape  
. -- Wildcard  
| -- OR  
() -- Group  
[] -- Set

(?x) -- ignore white space  
(?m) -- multiple lines  
(?i) -- Case insensitive  
(?=...) -- Positive Look Ahead  
(?!...) -- Negative Look Ahead  
(?:...) -- No Back Reference

# Escape Sequences

Escape sequences are special phrases recognized by the RegEx engine and used for matching.

Some Common Ones are:

`\b` – boundary from alpha-num to non-alpha-num

`\n` – newline

`\d` – numeric [`\D` for non]

`\w` – Word (alpha-numeric) [`\W` for non]

`\s` – White Space [`\S` for non]



# Examples... Email...

## Find the Email Address:

```
<cfset phrase =  
"Drop Gypsy a line @ his email address which is  
  regexguy@Gypsy.org just cuz">  
<cfset Index = REFindNoCase(  
"\b[\w-\.\_]+@[ \w\.-]+\.[\w+\.-]+\s*\b",phrase,1)>  
<cfoutput>Gypsy's Email is at Position #Index#<BR></cfoutput>  
--OR-- <BR>  
<cfset myIndex = REFindNoCase(  
"\b[\w-\.\_]+@[ \w\.-]+\.[\w+\.-]+\s*\b",phrase,1,"True")>  
<cfset email = Mid(phrase,myIndex.pos[1],myIndex.len[1])>  
<cfoutput>Gypsy's Email Address is: #email#<BR></cfoutput>
```

## Yields:

```
Gypsy's Email is at Position 48
```

```
--OR--
```

```
Gypsy's Email Address is: regexguy@Gypsy.org
```

# Examples... Phone Number

Code:

```
<cfset phone= "1-612.555.1212">  
<!-- First lets ditch the dashes -->  
<cfset phone = REReplace(phone, "\D", "", "ALL")>  
<cfoutput>Dashless number: #phone#<BR></cfoutput>  
<cfset four = Right(phone,4)>  
<cfset prefix = Left(Right(phone,7),3)>  
<cfset areacode = Left(Right(phone,10),3)>  
<cfoutput>Number Nice: (#areacode#) #prefix#-#four#</cfoutput>
```

Yields:

Dashless number: 16125551212

Number Nice: (612) 555-1212

# Examples... Logfile

```
<cfset logline = '64.243.10.4 - - [01/Feb/2004:18:37:07 -0600] ' &
    '"GET /immortals.jpg HTTP/1.1" 200 47884'>
<cfset ipinfo = REFindNoCase("^\s*\d+\.\d+\.\d+\.\d+\s+", logline, 0, "TRUE")>
<cfset dateinfo = REFindNoCase("\d+\/\w+\/\d\d\d\d:\d\d:\d\d\s+", logline,
    (ipinfo.pos[1] + ipinfo.len[1] - 1), "TRUE")>
<cfset fileinfo = REFindNoCase("(GET|POST)\s+.*?\s+", logline,
    (dateinfo.pos[1] + dateinfo.len[1] - 1), "TRUE")>

<cfoutput>
IP: #RTrim(LTrim(Mid(logline, ipinfo.pos[1], ipinfo.len[1])))#<BR>
Date: #RTrim(LTrim(Mid(logline, dateinfo.pos[1], dateinfo.len[1])))#<BR>
File: #REReplace(RTrim(LTrim(Mid(logline, fileinfo.pos[1], fileinfo.len[1]))),
    "^ (GET|POST) \s+", "", "ONE")#<BR>
</cfoutput>
```

Yields:

```
IP: 64.243.10.4
Date: 01/Feb/2004:18:37:07
File: /immortals.jpg
```

# Examples... Repairing Data

```
<cfset myString = "That gypsy sure does like like regex a lot,
what a weirdo weirdo gypsy is.">
<!--- Gypsy should be capitalized --->
<cfset myString = REReplace(myString, "(gypsy)", "\u\1", "ALL")>
<cfoutput>#mystring#<BR></cfoutput>
<!--- Lets not repeat ourselves --->
<cfset myString = REReplace(myString, "([\w ]+)\1", "\1", "ALL")>
<cfoutput>#mystring#<BR></cfoutput>
```

Yields:

That Gypsy sure does like like regex a lot, what a weirdo weirdo  
Gypsy is.

That Gypsy sure does like regex a lot, what a weirdo Gypsy is.

# Bleahbebleahb....thats all folks

I thank you for your time and hope you've enjoyed this jaunt into Regular Expressions as much as I have.

Hope you got a lot out of it.

Because I've chosen to leave Digital North you will not be able to reach me through them, but I am available to answer questions about this presentation, or RegEx in general by email:

[CFRegEx@Gypsy.org](mailto:CFRegEx@Gypsy.org)