



Regex Practice Problems

▼ Regex Pattern Reference

Pattern	Description
abc...	Letters
123...	Digits
\d	Any Digit
\D	Any Non-digit character
.	Any Character
.	Period
[abc]	Only a, b, or c
[^abc]	Not a, b, nor c
[a-z]	Characters a to z
[0-9]	Numbers 0 to 9
\w	Any Alphanumeric character
\W	Any Non-alphanumeric character
{m}	m Repetitions
{m,n}	m to n Repetitions
*	Zero or more repetitions
+	One or more repetitions
?	Optional character

Pattern	Description
\s	Any Whitespace
\S	Any Non-whitespace character
^...\$	Starts and ends
(...)	Capture Group
(a(bc))	Capture Sub-group
(.*)	Capture all
(abc def)	Matches abc or def

▼ Problem 1: Matching decimal numbers

Exercise 1: Matching Numbers

Task	Text	
Match	3.14529	✓
Match	-255.34	✓
Match	128	✓
Match	1.9e10	✓
Match	123,340.00	✓
Skip	720p	
-?[123]\.?[1-9]e?[1-9]\.?[([0-9] ,)?[234]?[94]?([0.00)?		Continue >

▼ Problem 2: Matching Phone Numbers

Exercise 2: Matching phone numbers

Task	Text	Capture Groups	
Capture	415-555-1234	415	✓
Capture	650-555-2345	650	✓
Capture	(416)555-3456	416	✓
Capture	202 555 4567	202	✓
Capture	4035555678	403	✓
Capture	1 416 555 9292	416	✓
(\d{3})		Continue >	

▼ Problem 3: Matching emails

Exercise 3: Matching E-mails

Task	Text	Capture Groups	
Capture	tom@hogwarts.com	tom	✓
Capture	tom.riddle@hogwarts.com	tom.riddle	✓
Capture	tom.riddle+regexone@hogwarts.com	tom.riddle	✓
Capture	tom@hogwarts.eu.com	tom	✓
Capture	potter@hogwarts.com	potter	✓
Capture	harry@hogwarts.com	harry	✓
Capture	hermione+regexone@hogwarts.com	hermione	✓
<code>^([\w\.\.]*)</code>		Continue >	

1. `^` - Anchors the match to the start of each line
2. `()` - Creates a capture group to extract just the part we want
3. `[\w\.]` - Matches any word character (letters, digits, underscore) OR a period
4. `*` Means "zero or more" of the preceding characters

Why it stops at the right place:

The key insight is that `[\w\.]` only matches alphanumeric characters and periods. When the regex engine encounters either:

- `@` (in regular emails)
- `+` (in emails with plus addressing like `tom.riddle+regexone@hogwarts.com`)

...it stops matching because neither `@` nor `+` are included in the character class `[\w\.]`.

▼ Problem 4: Matching HTML

Exercise 4: Capturing HTML Tags

Task	Text	Capture Groups
Capture	<a>This is a link	a
Capture	Link	a
Capture	<div class='test_style'>Test</div>	div
Capture	<div>Hello world</div>	div
<(\w*)>		Continue >

▼ Problem 5: Matching specific filenames

Exercise 5: Capturing filename data

Task	Text	Capture Groups
Skip	.bash_profile	
Skip	workspace.doc	
Capture	img0912.jpg	img0912 jpg
Capture	updated_img0912.png	updated_img0912 png
Skip	documentation.html	
Capture	favicon.gif	favicon gif
Skip	img0912.jpg.tmp	
Skip	access.lock	
^(\w+)\.(jpg png gif)\$		Continue >

▼ Problem 6: Trimming whitespace from start and end of line

Exercise 6: Matching lines

Task	Text	Capture Groups
Capture	The quick brown fox...	The quick brown fox...
Capture	jumps over the lazy dog.	jumps over the lazy dog.
\s*(.*\s*)		Continue >

▼ Problem 7: Extracting information from a log file

Exercise 7: Extracting data from log entries

Task	Text	Capture Groups
Skip	W/dalvikvm(1553): threadid=1: uncaught exception	
Skip	E/(1553): FATAL EXCEPTION: main	
Skip	E/(1553): java.lang.StringIndexOutOfBoundsException	
Capture	E/(1553): at widget.List.makeView(ListView.java:1727)	<code>makeView</code> <code>ListView.java</code> 1727
		<input checked="" type="checkbox"/>
Capture	E/(1553): at widget.List.fillDown(ListView.java:652)	<code>fillDown</code> <code>ListView.java</code> 652
		<input checked="" type="checkbox"/>
Capture	E/(1553): at widget.List.fillFrom(ListView.java:709)	<code>fillFrom</code> <code>ListView.java</code> 709
		<input checked="" type="checkbox"/>
<code>(\w+)\((([\w\.\.]+):(\d+)\)</code>		Continue >

`(\w+)` - First Capture Group (Method Name)

- `\w+` matches one or more word characters (letters, digits, underscore)
- Parentheses `()` create a capture group
- Captures:** `makeView`, `fillDown`, `fillFrom`

`\(` - Literal Opening Parenthesis

- The backslash escapes the `(` because parentheses have special meaning in regex
- Matches:** the literal `(` character before the filename

`([\w\.\.]+)` - Second Capture Group (Filename)

- `[\w\.\.]+` matches word characters OR periods
- `+` means one or more of these characters
- Captures:** `ListView.java`, `ListView.java`, `ListView.java`

`:` - Literal Colon

- Matches:** the `:` separator between filename and line number

`(\d+)` - Third Capture Group (Line Number)

- `\d+` matches one or more digits

- Captures: 1727, 652, 709

▼ Problem 8: Parsing and extracting data from a URL

Exercise 8: Extracting data from URLs

Task	Text	Capture Groups
Capture	ftp://file_server.com:21/top_secret/life_changing_plans.pdf	ftp file_server.com 21
Capture	https://regexone.com/lesson/introduction#section	https regexone.com
Capture	file://localhost:4040/zip_file	file localhost 4040
Capture	https://s3cur3-server.com:9999/	https s3cur3-server.com 9999
Capture	market://search/angry%20birds	market search
<code>(\w+)://([\w\-\.\.]+)(:(\d+))?</code>		Continue >

▼ Part-1: `(\w+)`

- `(` - Opens capture group #1
- `\w` - Matches any "word character":
 - Letters: `a-z`, `A-Z`
 - Digits: `0-9`
 - Underscore: `_`
- `+` - One or more of the preceding character
- `)` - Closes capture group #1

`::/`

- **Literal text** that must match exactly
- This is the standard separator in URLs

Captures: `ftp`, `https`, `file`, `market`

▼ Part 2: `([\w\-\.\.]+)`

(- Opens capture group #2

[w\-.]+

- ... - Character class (match any ONE of these characters)
- w - Word characters (letters, digits, underscore)
- \- Literal hyphen (escaped to avoid range interpretation)
- \. - Literal period (escaped because . normally means "any character")
- + - One or more characters from this set

) - Closes capture group #2

Captures: file_server.com , regexone.com , localhost , s3cur3-server.com

▼ Part 3: (:(\d+))?

This is the **optional port section** - notice the ? at the very end makes the entire group optional.

(- Opens capture group #3

:

- **Literal colon** - the port separator in URLs

(\d+)

- (- Opens capture group #4 (nested inside group #3)
- \d - Matches any digit (0-9)
- + - One or more digits
-) - Closes capture group #4

) - Closes capture group #3

? - Makes the entire group (:(\d+)) **optional**