Shortcodes, Modifiers, and Delimiters

Juliette Reinders Folmer

@jrf_nl | regexcheatsheets.com

Shortcodes

Common Shortcodes

[0-9]

\d

[A-Za-z0-9_]

\w

[\t\f\r\n]

\s

Negated Shortcodes

[^0-9]

\D

[^A-Za-z0-9_]

\W

[^\t\f\r\n]

\S

PCRE 7.2+ Shortcodes

[\t\f]

\h

[\r\n]

\v

[^\t\f]

\H

[^\r\n]

\V

PCRE 7.2 was released in June 2007

Here

be

dragons

Pitfalls

[^\D\S]

PCRE
vs.
POSIX

PCRE vs. POSIX Syntax

PCRE (common)	POSIX
\s	[:space:]
\s+	[[:space:]]+
[\s\d]+	[[:space:][:digit:]]+

Character classes		PCRE		POSIX	
[0-9]	[^0-9]	\d	\D	[[:digit:]]	[^[:digit:]]
[A-Za-z0-9_]	[^A-Za-z0-9_]	\w	\W	[[:word:]]	[^[:word:]]
[\t\f\r\n \v]	[^\t\f\r\n \v]	\s	\S	[[:space:]]	[^[:space:]]
[\t\f]	[^\t\f]	\h	\H	[[:blank:]]	[^[:blank:]]
[\r\n]	[^\r\n]	\v	\V	-	-

PCRE vs. POSIX

Extra POSIX Shorthand Character Classes

POSIX

- [:alnum:]
- [:alpha:]
- [:ascii:]
- [:cntrl:]
- [:graph:]
- [:lower:]
- [:print:]
- [:punct:]
- [:upper:]
- [:xdigit:]

Hex	Dec	Char		Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char
0x00	0	NULL	null	0x20	32	Space	0x40	64	@	0x60	96	`
0x01	1	SOH	Start of heading	0x21	33	!	0x41	65	A	0x61	97	a
0x02	2	STX	Start of text	0x22	34	"	0x42	66	B	0x62	98	b
0x03	3	ETX	End of text	0x23	35	#	0x43	67	C	0x63	99	c
0x04	4	EOT	End of transmission	0x24	36	$	0x44	68	D	0x64	100	d
0x05	5	ENQ	Enquiry	0x25	37	%	0x45	69	E	0x65	101	e
0x06	6	ACK	Acknowledge	0x26	38	&	0x46	70	F	0x66	102	f
0x07	7	BELL	Bell	0x27	39	'	0x47	71	G	0x67	103	g
0x08	8	BS	Backspace	0x28	40	(	0x48	72	H	0x68	104	h
0x09	9	TAB	Horizontal tab	0x29	41	)	0x49	73	I	0x69	105	i
0x0A	10	LF	New line	0x2A	42	*	0x4A	74	J	0x6A	106	j
0x0B	11	VT	Vertical tab	0x2B	43	+	0x4B	75	K	0x6B	107	k
0x0C	12	FF	Form Feed	0x2C	44	,	0x4C	76	L	0x6C	108	l
0x0D	13	CR	Carriage return	0x2D	45	-	0x4D	77	M	0x6D	109	m
0x0E	14	SO	Shift out	0x2E	46	.	0x4E	78	N	0x6E	110	n
0x0F	15	SI	Shift in	0x2F	47	/	0x4F	79	O	0x6F	111	o
0x10	16	DLE	Data link escape	0x30	48	0	0x50	80	P	0x70	112	p
0x11	17	DC1	Device control 1	0x31	49	1	0x51	81	Q	0x71	113	q
0x12	18	DC2	Device control 2	0x32	50	2	0x52	82	R	0x72	114	r
0x13	19	DC3	Device control 3	0x33	51	3	0x53	83	S	0x73	115	s
0x14	20	DC4	Decive control 4	0x34	52	4	0x54	84	T	0x74	116	t
0x15	21	NAK	Negative ack	0x35	53	5	0x55	85	U	0x75	117	u
0x16	22	SYN	Synchronous idle	0x36	54	6	0x56	86	V	0x76	118	v
0x17	23	ETB	End transmission block	0x37	55	7	0x57	87	W	0x77	119	w
0x18	24	CAN	Cancel	0x38	56	8	0x58	88	X	0x78	120	x
0x19	25	EM	End of medium	0x39	57	9	0x59	89	Y	0x79	121	y
0x1A	26	SUB	Substitute	0x3A	58	:	0x5A	90	Z	0x7A	122	z
0x1B	27	FSC	Escape	0x3B	59	;	0x5B	91	[	0x7B	123	{
0x1C	28	FS	File separator	0x3C	60	<	0x5C	92	\	0x7C	124	|
0x1D	29	GS	Group separator	0x3D	61	=	0x5D	93	]	0x7D	125	}
0x1E	30	RS	Record separator	0x3E	62	>	0x5E	94	^	0x7E	126	~
0x1F	31	US	Unit separator	0x3F	63	?	0x5F	95	_	0x7F	127	DEL

Pitfalls

[^\D\S]

PCRE
vs.
POSIX

Locale

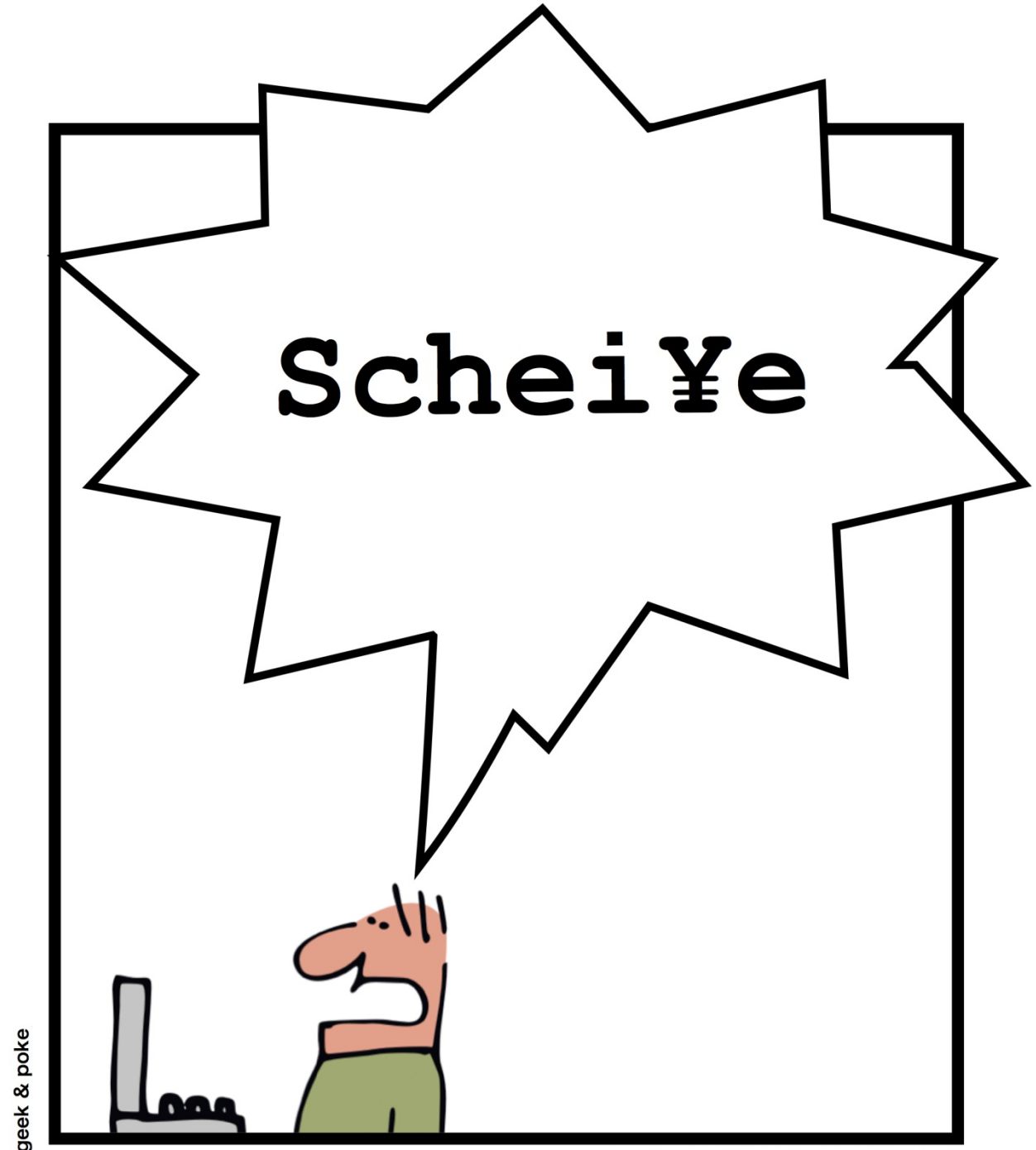
Locale

English (en)

déjà vu [\w]+ 

French (fr)

déjà vu [\w]+ 


Schei¥e
geek & poke
CODERS LOVE UNICODE

Uses \w and \W to find boundaries.

Locale

English (en)

déjà vu [\w]+

French (fr)

déjà vu [\w]+

Pitfalls

`[^\D\S]`	PCRE vs. POSIX	Locale	Engines & Implementations

Inconsistent Implementations

[\f\n\r\t]

[\f\n\r\t\v\x85\p{Z}]

[\f\n\r\t\x0B]

[\f\n\r\t\v\u1680\u180e
\u2000\u2001\u2002
\u2003\u2004\u2005
\u2006\u2007\u2008
\u2009\u200a\u2028
\u2029\u202f\u205f\u3000]

[\f\n\r\t\v]

[\f\n\r\t\p{Z}]

To Use or Not to Use ?

Advantages

- Adjust to locale

Disadvantages

- Adjust to locale
- Inconsistent implementations
- Low portability
- Difficult to unit test

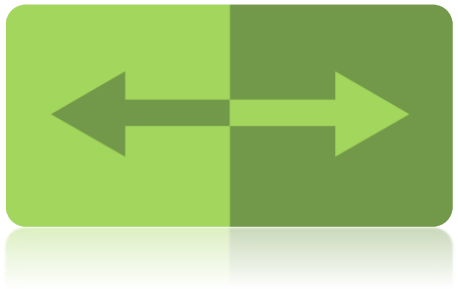
Unicode Shortcodes

Is Unicode Supported ?

Engine support | Compilation | Input encoding

Graphmeme Clusters

à



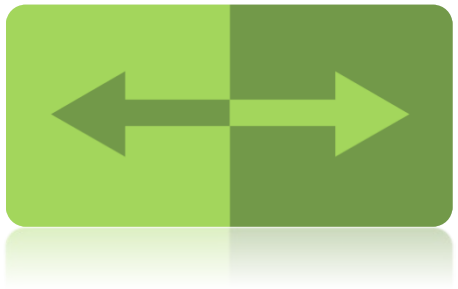
à

U+00E0

U+0061 + U+0300

Codepoints vs. Graphmeme Clusters

à



à

U+00E0

U+0061 + U+0300

Codepoints vs. Graphmeme Clusters

à à

U+00E0 U+0061 + U+0300

Matches:

Unicode Wildcard *

\X

- Matches graphmeme clusters
  - ~ equivalent to \P{M}\p{M}*+
- Matches new line

* Included in the PCRE standard, but not widely supported (yet)

Unicode Range Identifiers

Categories

Scripts

Blocks

Binary properties

Unicode Shortcode Syntax

Positive

```
\p{Identifier}
```

Negative

```
\P{Identifier}
```

Unicode Shortcodes

\p{..}

\P{..}

- {Identifier}
- Mind: Capitalization
- Can be used anywhere

Unicode Range Identifiers

Categories

Scripts

Blocks

Binary properties

Category-based Unicode Shortcodes

Letters	\p{L}
Marks	\p{M}
Numbers	\p{N}
Punctuation	\p{P}
Symbols	\p{S}
Separators	\p{Z}
Other	\p{C}

\p{Ll}	Letter: Lowercase
\p{Lm}	Letter: Mark/modifier
\p{Lo}	Letter: Other
\p{Lt}	Letter: Titlecase
\p{Lu}	Letter: Uppercase

Alternative Syntaxes:

\pX

\p{Category}

\w ≠ \p{L}

Close approximation:

[\p{L}\p{M}\p{Nd}\p{Nl}\p{Pc}\u200c\u200d]

Unicode Range Identifiers

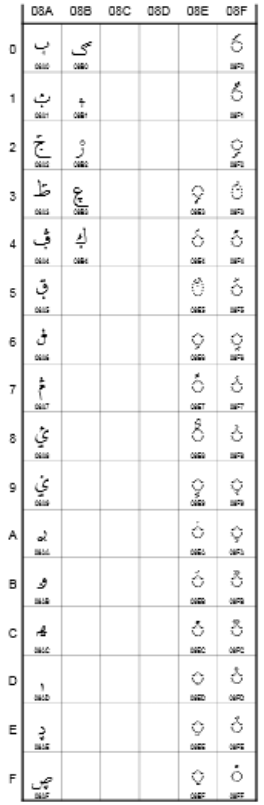
Categories

Scripts

Blocks

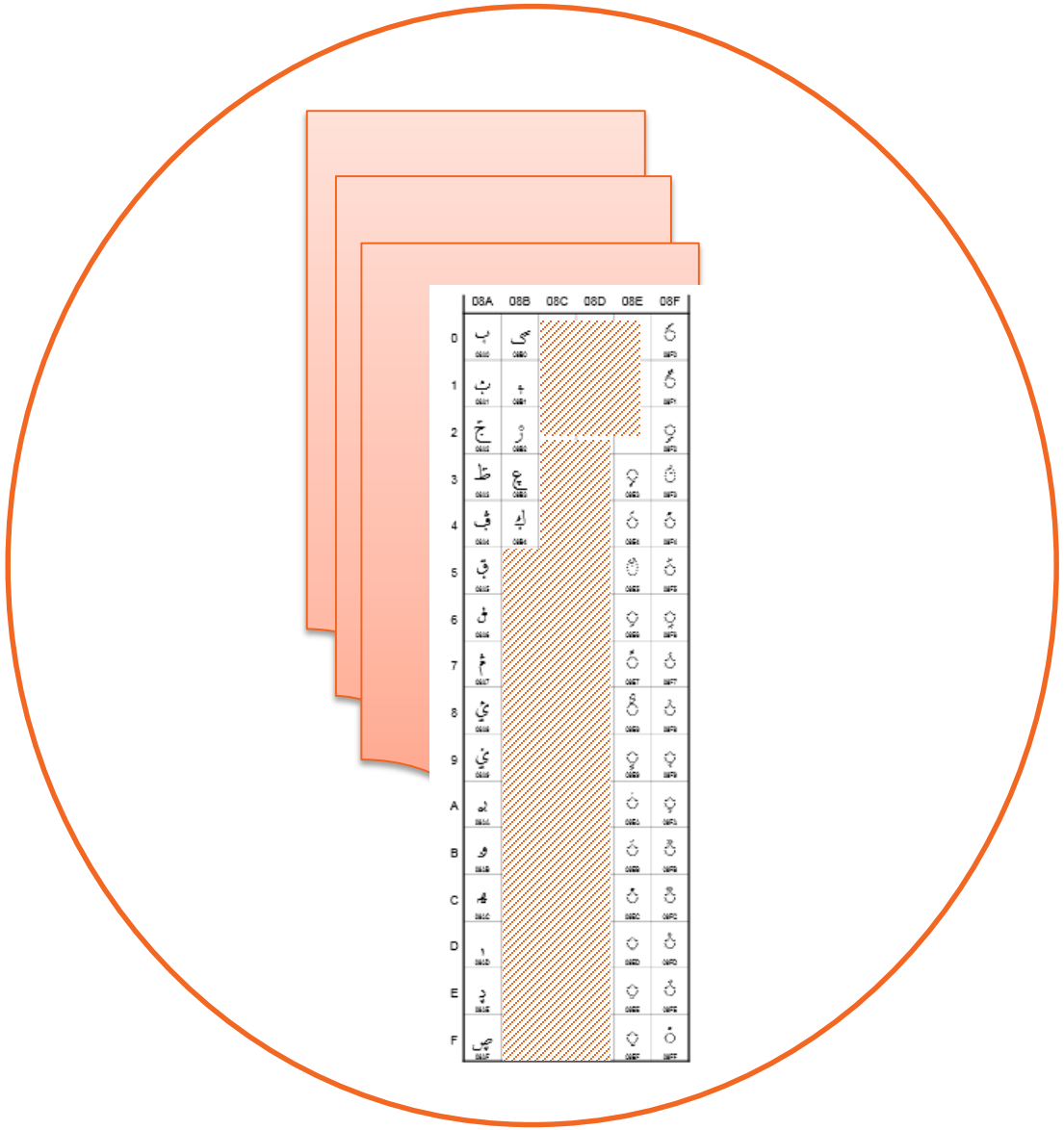
Binary properties

Blocks vs. Scripts



Codepoint Block

Blocks vs. Scripts



Script

Script

- \p{Scriptname}
- \p{IsScriptName}
- \p{script=ScriptName}
- \p{sc=ScriptName}

Block

- \p{Blockname}
- \p{InBlockName}
- \p{IsBlockName}
- \p{block=BlockName}
- \p{blk=BlockName}

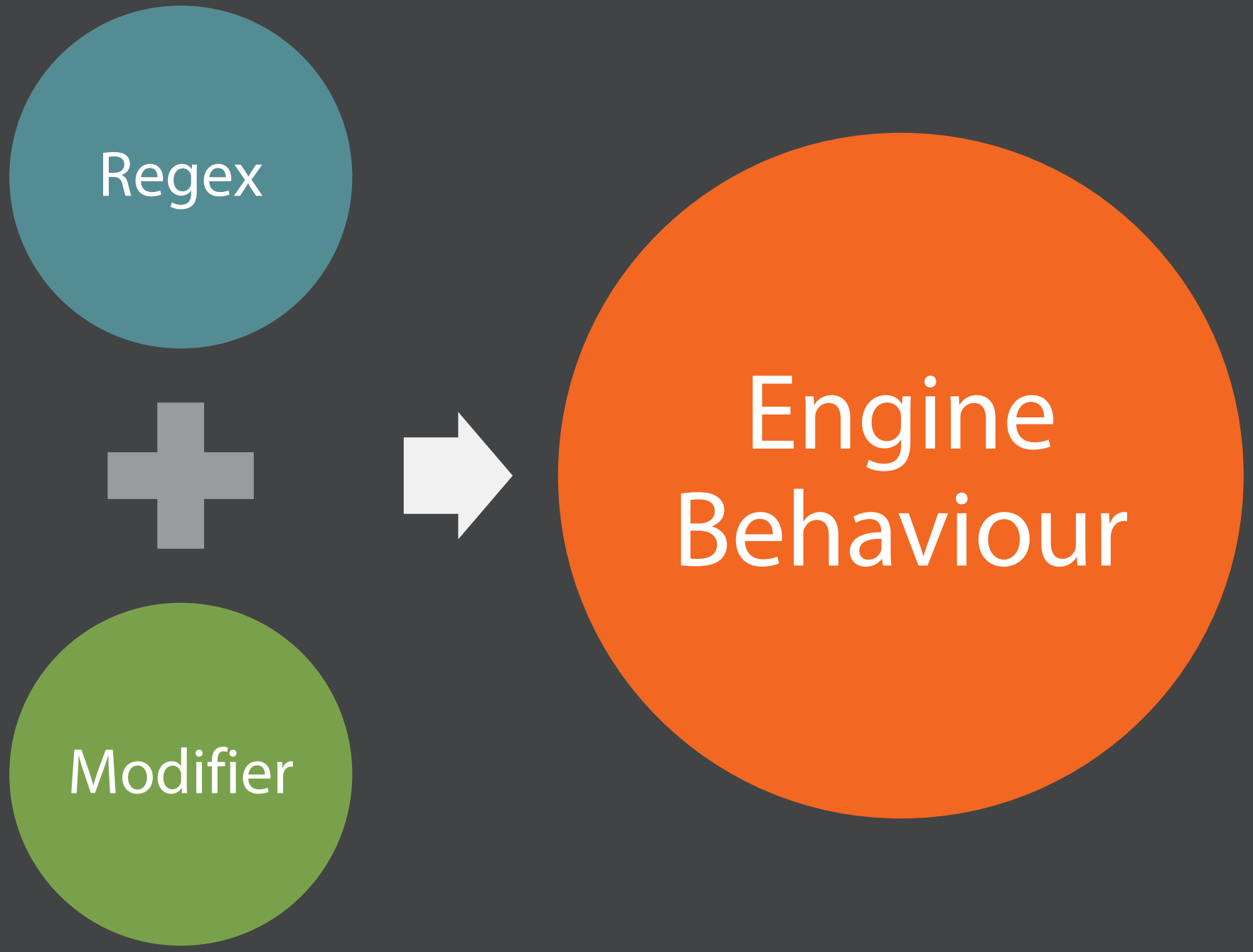
\p{Cyrillic}

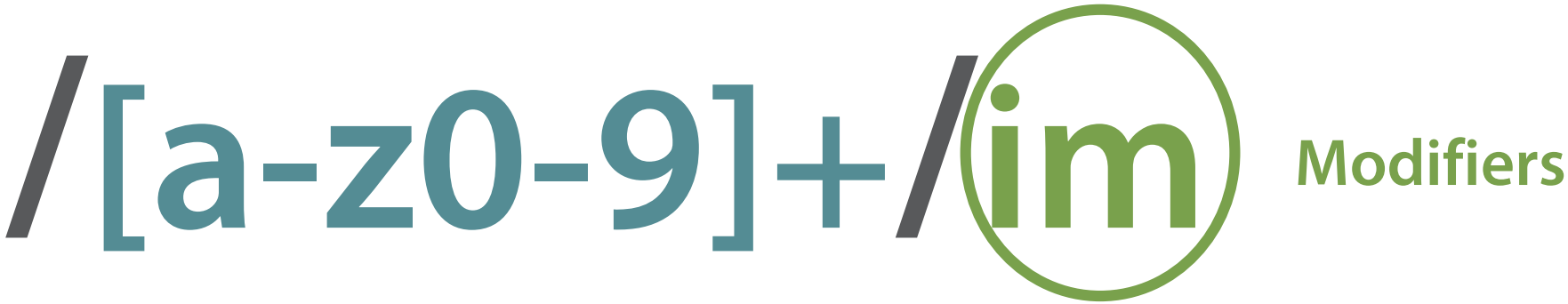
\p{InCyrillic}

\p{InCyrillic_Supplementary}


Photo by Michael Coghlan

Modifiers


Regex
Modifier
Engine
Behaviour


/[a-z0-9]+/im
Modifiers

Applying Modifiers

/regex/m

m/regex/

match('regex', modifiers)

new Re(/regex/, flags)

preg_match()

vs.

preg_match_all()

(?m)

g* i m s x

Modifiers

g*

i

m

s

x

- GLOBAL
- Return all matches vs. first match
- Non-overlapping

Modifiers

g*

i

m

s

x

- CASE-INSENSITIVE
- Mind locales
  - German: FUSSBALL vs. fußball

Modifiers

g*

i

m

s

x

- MULTILINE
- Affects ^ and $ behaviour

Modifiers

g*

i

m

s

x

S

- DOTALL or SINGLELINE
- Affects . (dot) to match \n
- Slow ⁿ

Modifiers

g*

i

m

s

x

- EXTENDED

```
/^((
    25[0-5]|                    # Match 250-255 range
    2[0-4][0-9]|                # Match 200-249 range
    [01]?[0-9]{1,2}             # Match   0-199 range
)\.){3}                         # Repeat 3 times with period
(25[0-5]|2[0-4][0-9]|[01]?[0-9]{1,2}) # and once without
$/x
```

Modifiers

g*

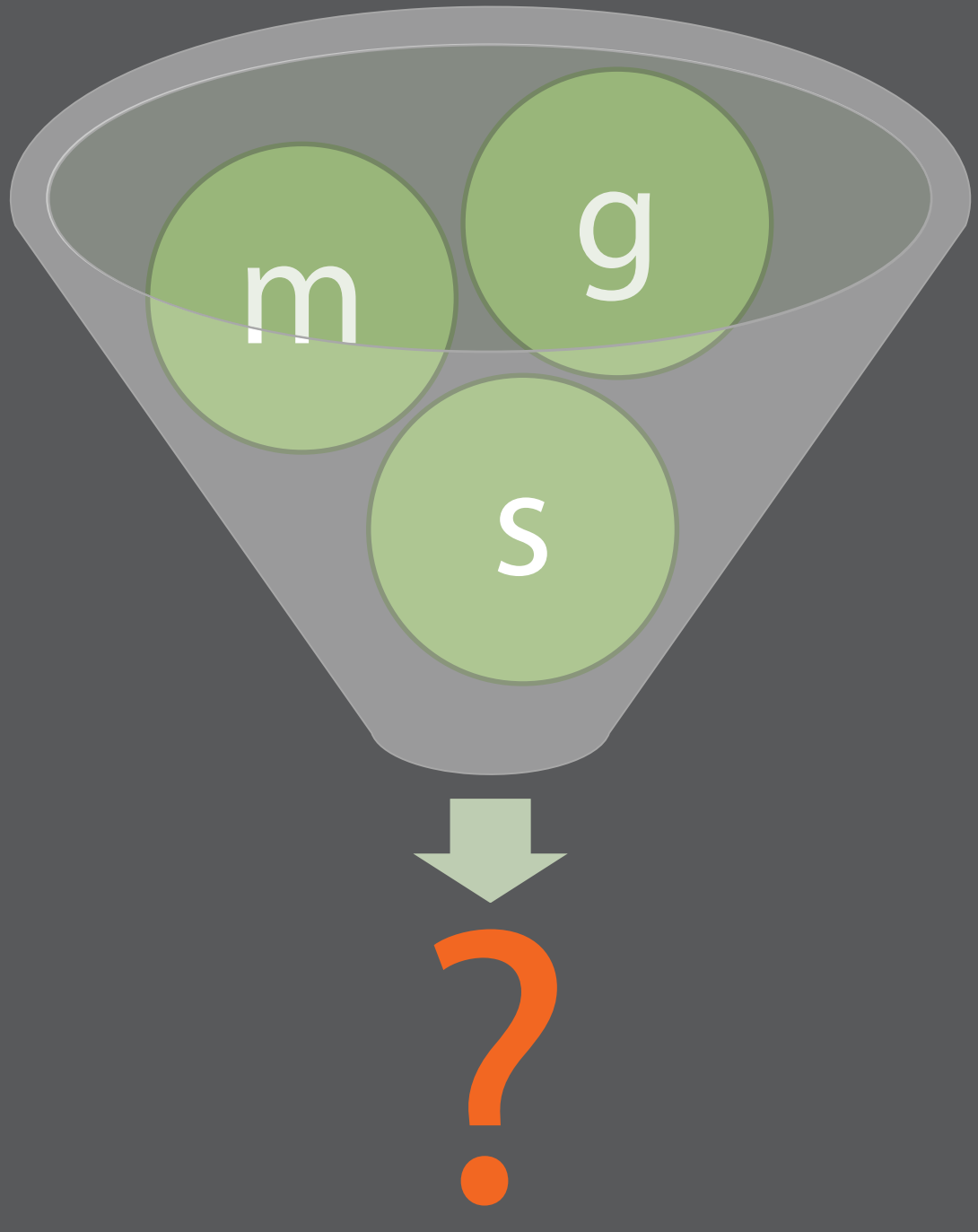
i

m

s

x

- EXTENDED
- Ignore whitespace & # to end of line
- Mind: escaping
  - \x20
  - \#


m
g
s
?

Inline Modifiers

Setting:

(?i)


(?i)caseless(?-i)cased(?i)caseless


cased(case(?i)insen|sitive)cased

Inline Modifiers

Setting:

(?i)

Unsetting:

(?-i)

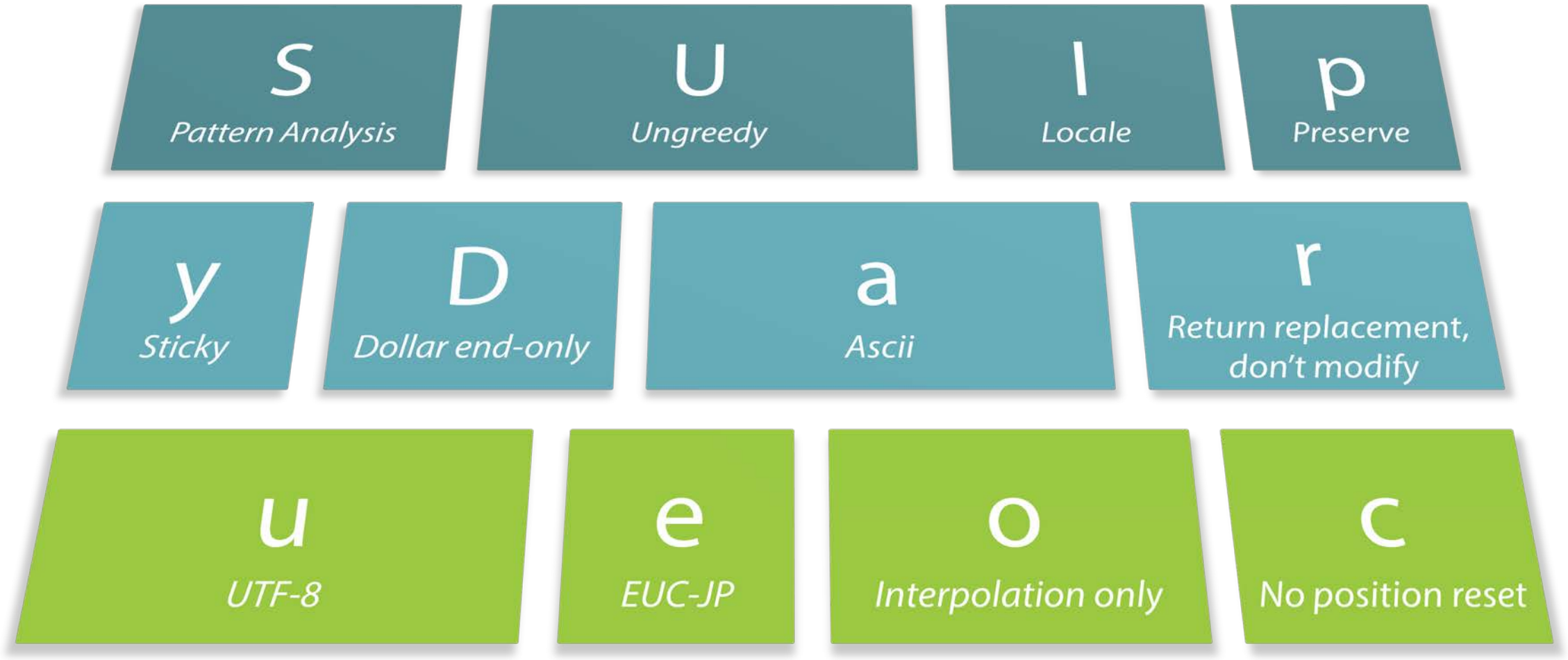
Combined:

(?im-sx)

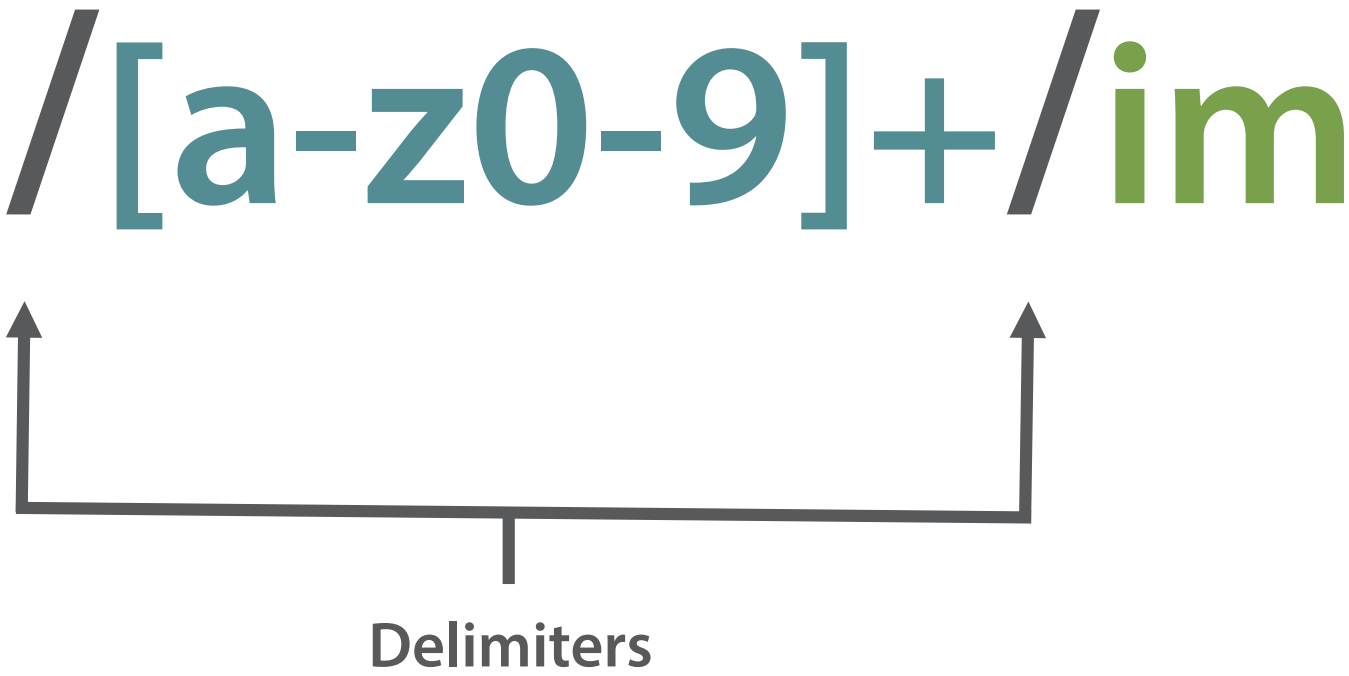
Apply to subpattern (non-capturing):

(?i:subp)

Explore


S
Pattern Analysis
U
Ungreedy
l
Locale
p
Preserve
y
Sticky
D
Dollar end-only
a
Ascii
r
Return replacement, don't modify
u
UTF-8
e
EUC-JP
o
Interpolation only
c
No position reset

Delimiters


/[a-z0-9]+/im
Delimiters

Delimiters

- Enclose the pattern
- Not always needed
- Alternative delimiters

Alternative Delimiter Requirements

Non-alphanumeric | Non-backslash | Non-whitespace

Alternative Delimiters

!\[0-9]+!	#\[0-9]+#	@\[0-9]+@
`\[0-9]+`	~\[0-9]+~	%\[0-9]+%

Did you know ?

You can use brackets as delimiters:

(p[at]{2}te(rn)) {p[at]{2}te(rn)}

[p[at]{2}te(rn)] <p[at]{2}te(rn)>

/http:\/\/\p{L}+\.[a-z]+\//

vs.

`http://\p{L}+\.[a-z]+/`

Choose Wisely

Next up:

Working with Matches

Juliette Reinders Folmer

@jrf_nl | regexcheatsheets.com