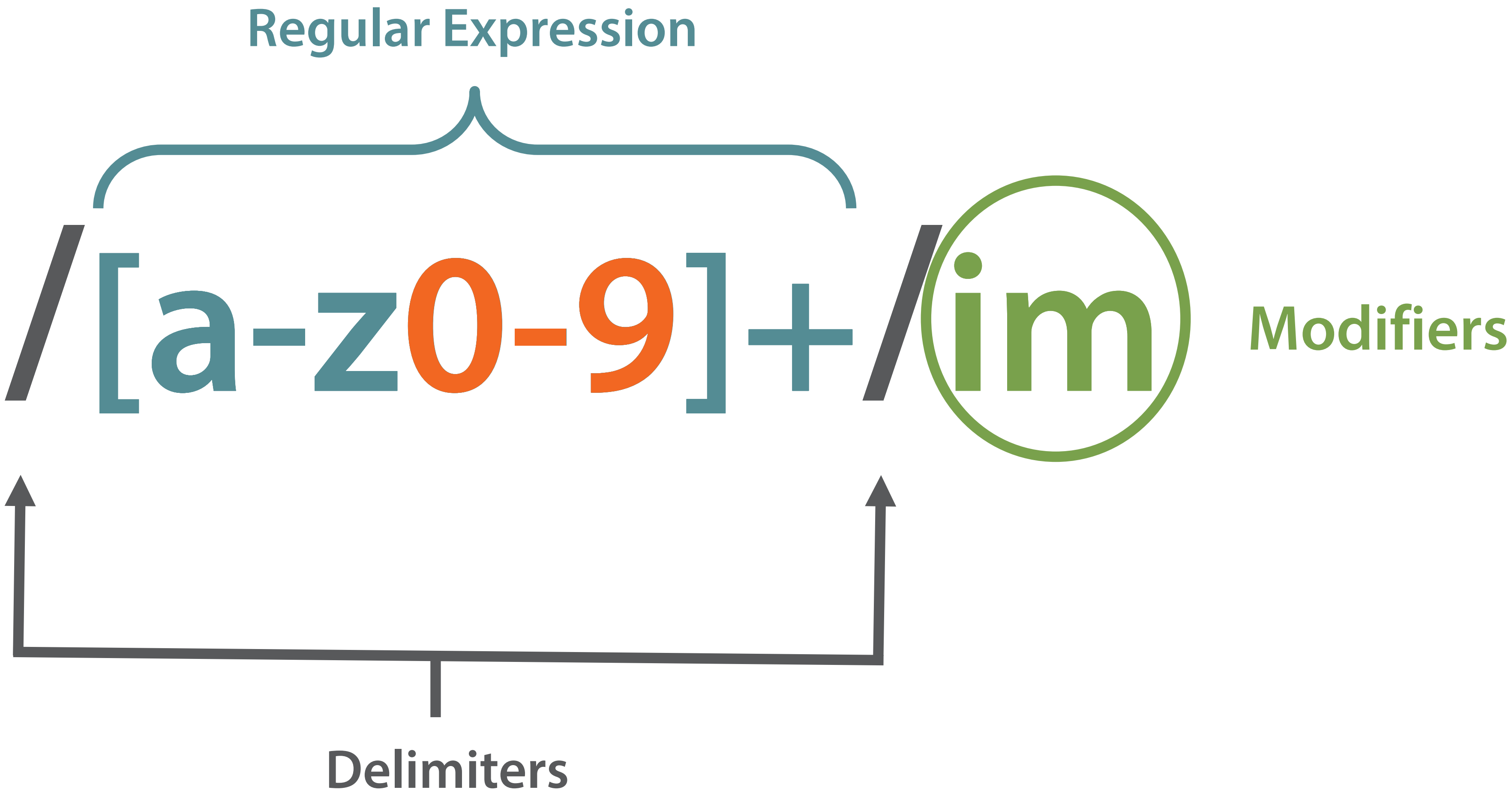
Terminology



Regular Expression

/[a-z0-9]+/im

Modifiers

Delimiters

On Engines, Dialects and Influencers

Evolution

. \

[^ $

*

? +

(|)

{ }

\s \d

(?=...)

(?:...)

(?P<>...)

(?#...)

Ordinary Characters

A a

0

- Match themselves

abc

abcdefghijklmnopqrstuvwxyz

Matching Characters

Control Characters or Non-Printing Characters

Photo by Renato Targa

Control characters

Hex	Escape sequence	Represents
0	\0	Null
7	\a	Bell
8	\b	Backspace
9	\t	Horizontal tab
10	\n	Line feed
11	\v	Vertical tab
12	\f	Form feed
13	\r	Carriage return
27	\e	Escape

Photo by Renato Targa

Control Sequences

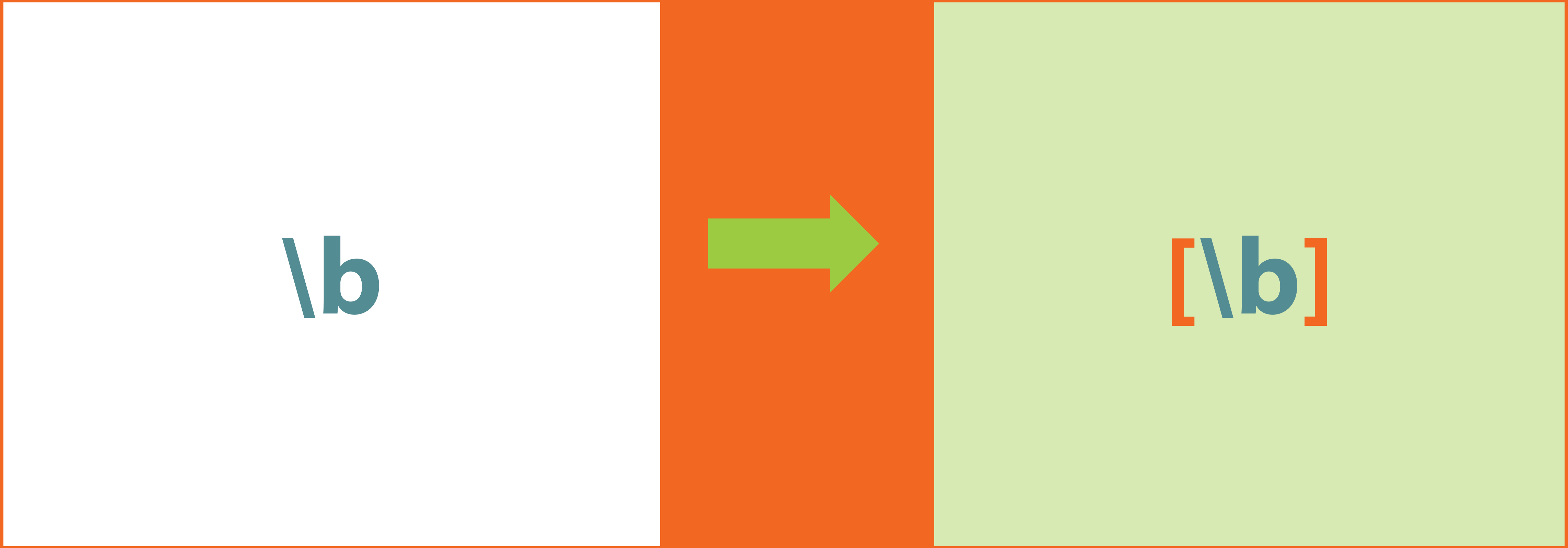
\cX	Control sequences
\XXX	Octals
\0XX	
\xHH	Hex codes
\x{HHHH}	
\uHHHH	Unicode codepoints
\u{H...}	

Did you know ?

Unix, Mac and Windows line endings in one go:

\R

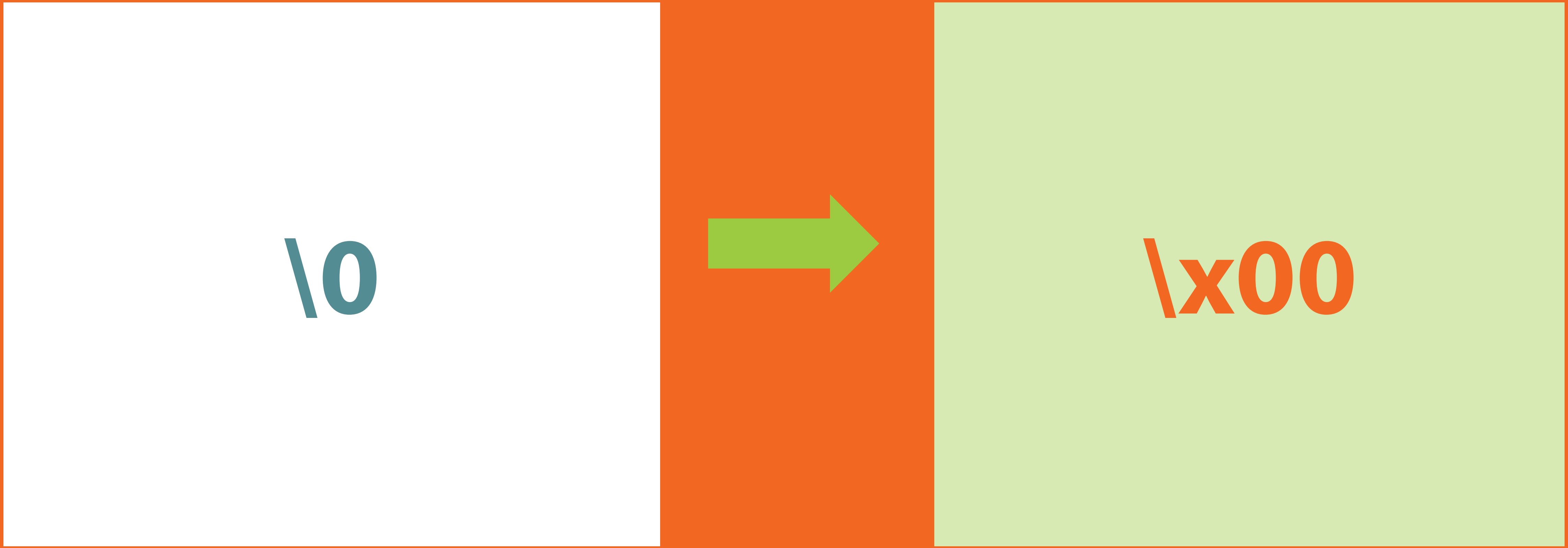
Pitfalls



\b

[\b]

Pitfalls



\0

\x00

Meta Characters

Characters with Special Meaning

Meta Characters

Character Classes

Defining Character Classes

[ ]

Positive

- [abcdef]
- [a-f]
- [a-f_%0-9]

Negative

- [^abcdef]
- [^a-f]
- [^a-f_%0-9]

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

[A-Z]

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

[A-z]

Pitfalls

[A-z]

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

[0-9]

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	[illegible]	[illegible]	&	0x46	70	F	0x66	102	f
0x07	7	BELL	Bell	[illegible]	[illegible]	[illegible]	0x47	71	G	0x67	103	g
0x08	8	BS	Backspace	[illegible]	[illegible]	[illegible]	0x48	72	H	0x68	104	h
0x09	9	TAB	Horizontal tab	0x29	41	[illegible]	0x49	73	I	0x69	105	i
0x0A	10	LF	New line	0x2A	42	[illegible]	0x4A	74	J	0x6A	106	j
0x0B	11	VT	Vertical tab	0x2B	43	[illegible]	0x4B	75	K	0x6B	107	k
0x0C	12	FF	Form Feed	0x2C	[illegible]	[illegible]	0x4C	76	L	0x6C	108	l
0x0D	13	CR	Carriage return	0x2D	[illegible]	-	0x4D	77	M	0x6D	109	m
0x0E	14	SO	Shift out	0x2E	[illegible]	.	0x4E	78	N	0x6E	110	n
0x0F	15	SI	Shift in	0x2F	[illegible]	/	0x4F	79	O	0x6F	111	o
0x10	16	DLE	Data link escape	0x30	[illegible]	0	0x50	80	P	0x70	112	p
0x11	17	DC1	Device control 1	[illegible]	[illegible]	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	[illegible]	[illegible]	4	0x54	84	T	0x74	116	t
0x15	21	NAK	Negative ack	[illegible]	[illegible]	5	0x55	85	U	0x75	117	u
0x16	22	SYN	Synchronous idle	[illegible]	[illegible]	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

[9-0]

Meta Characters

Wildcard

Wildcard

.

- Match any character
  - except new line
  - but matches \n with dotall modifier
- Not special in a class

Meta Characters

Quantifiers and Greediness

Repetition Quantifiers

?

zero or one times

*

zero or more times (unlimited [‡])

+

one or more times (unlimited[‡])

Repetition Quantifiers

{n}

exactly ***n*** times

{n,}

n or **more** times (unlimited‡)

{n,m}

between ***n*** and ***m*** times

{,m}

between ***0*** and ***m*** times
same as: **{0,m}**

Quantifiers Apply to Units

```
abcd+
```

```
[ab]*cd
```

```
a(b|c)?d
```

/ /

We take one step forward, two steps back

/ o /

```
We take one step forward, two steps back
```

/on/

```
We take one step forward, two steps back
```

/ one /

```
We take one step forward, two steps back
```

/ one. /

```
We take one step forward, two steps back
```

/ one.* /

```
We take one step forward, two steps back
```

/ one.*s /

```
We take one step forward, two steps back
```

/ one.*s. /

```
We take one step forward, two steps back
```

/ one.*s.?   /

```
We take one step forward, two steps back
```

```
/ one.*s.?t /
```

```
We take one step forward, two steps back
```

/ one.*s.?t[a-z] /

```
We take one step forward, two steps back
```

```
/ one.*s.?t[a-z]+ /
```

```
We take one step forward, two steps back
```

```
/ one.*s.?t[a-z]+p                /
```

```
We take one step forward, two steps back
```

/ one.*s.?t[a-z]+p /

= space

```
We take one step forward, two steps back
```

```
/ one.*s.?t[a-z]+p .                  /
```

```
We take one step forward, two steps back
```

```
/ one.*s.?t[a-z]+p .{2,}                /
```

```
We take one step forward, two steps back
```

```
/ one.*s.?t[a-z]+p .{2,},                    /
```

```
We take one step forward, two steps back
                        ✓
```

/ o /

```
We take one step back, two steps forward
```

/ on /

```
We take one step back, two steps forward
```

/ one /

```
We take one step back, two steps forward
```

/ one. /

We take one step back, two steps forward

/ one.*? /

```
We take one step back, two steps forward
```

/ one.*?b /

```
We take one step back, two steps forward
```

```
/ one.*?b                                    /
```

```
We take one step back, two steps forward
```

/ one.*?b /

```
We take one step back, two steps forward
```

```
/ one.*?b                                   /
```

```
We take one step back, two steps forward
```

```
/ one.*?b                                   /
```

```
We take one step back, two steps forward
```

```
/ one.*?b                                  /
```

```
We take one step back, two steps forward
```

```
/ one.*?b                               /
```

```
We take one step back, two steps forward
```

```
/ one.*?b[a-z]                                   /
```

```
We take one step back, two steps forward
```

```
/ one.*?b[a-z]+?                                    /
```

```
We take one step back, two steps forward
```

/ one.*?b[a-z]+?k /

```
We take one step back, two steps forward
```

```
/ one.*?b[a-z]+?k                   /
```

```
We take one step back, two steps forward
```

Meta Characters

Alternation

Alternation

|

- Separates branches

```
Branch 1|something else|numbers FTW: [0-9]+|[abc]{2,4}
```

Alternation

|

- Separates branches
- Not special in a class

`[ad|]+`

abcd|abcd|abcd|abcd

Sub-patterns and Grouping

( )

- Create a sub-expression
  - Delimit alternations
  - Repetition

IPv4 Address

0.0.0.0 - 255.255.255.255

Sub-patterns and Grouping

( )

- Create a sub-expression
  - Delimit alternations
  - Repetition
- Remember sub-pattern matches

Match Array

[0] – Complete match

[1] – Match against sub-pattern 1

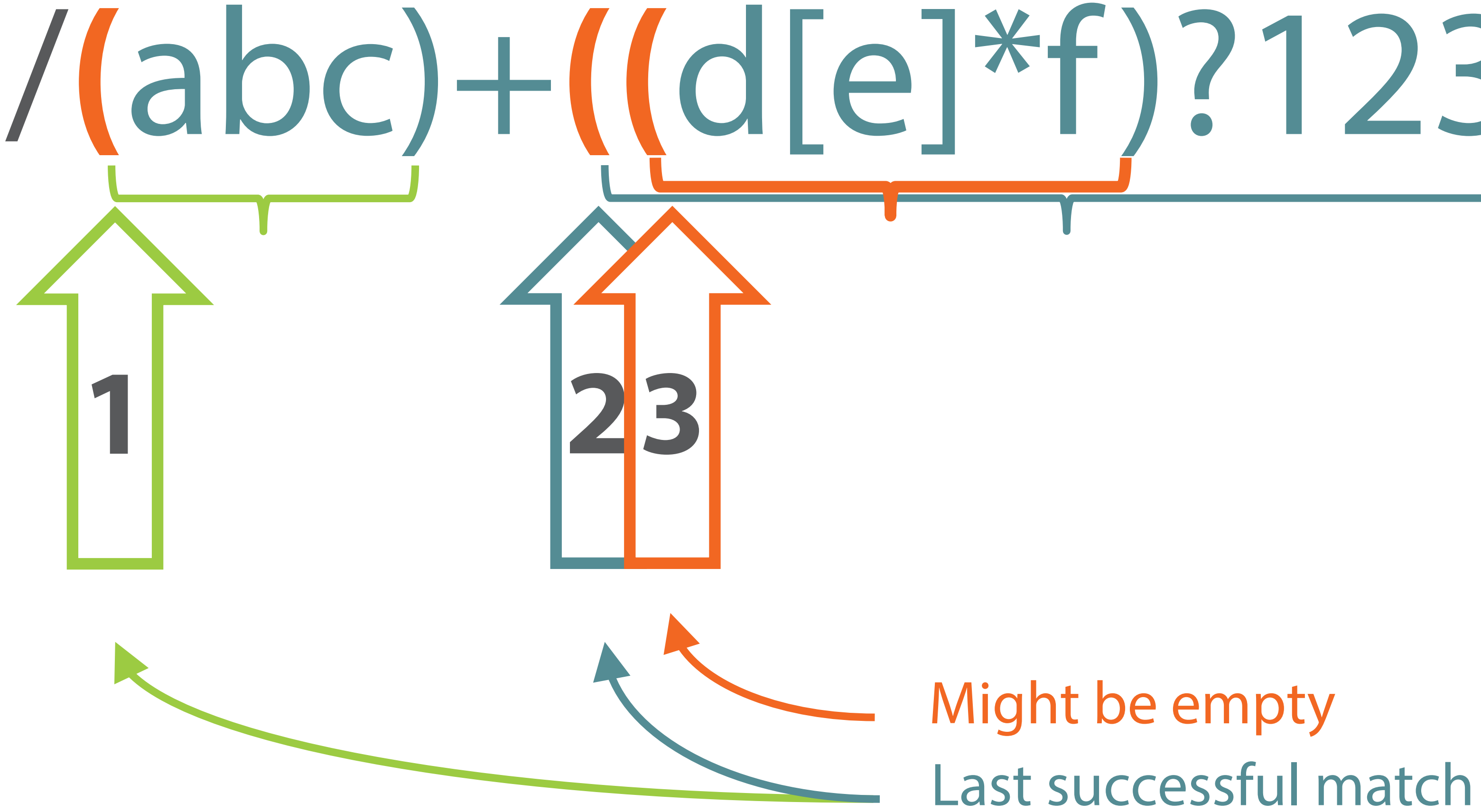
[2] – Match against sub-pattern 2

[3] – Match against sub-pattern 3

…

Photo by Petr Kratochvil

Submatch Order



```
/(abc)+((d[e]*f)?123){2}/
```

Might be empty

Last successful match

Sub-patterns and Grouping

( )

- Create a sub-expression
  - Delimit alternations
  - Repetition
- Remember sub-pattern matches
- Apply advanced features: (?...)

Advanced Features

- Look around
- Named sub-matches
- Conditional sub-patterns
- Recursion
- Inline comments

Sub-patterns and Grouping

( )

- Create a sub-expression
  - Delimit alternations
  - Repetition
- Remember sub-pattern matches
- Apply advanced features: (?...)

Meta Characters

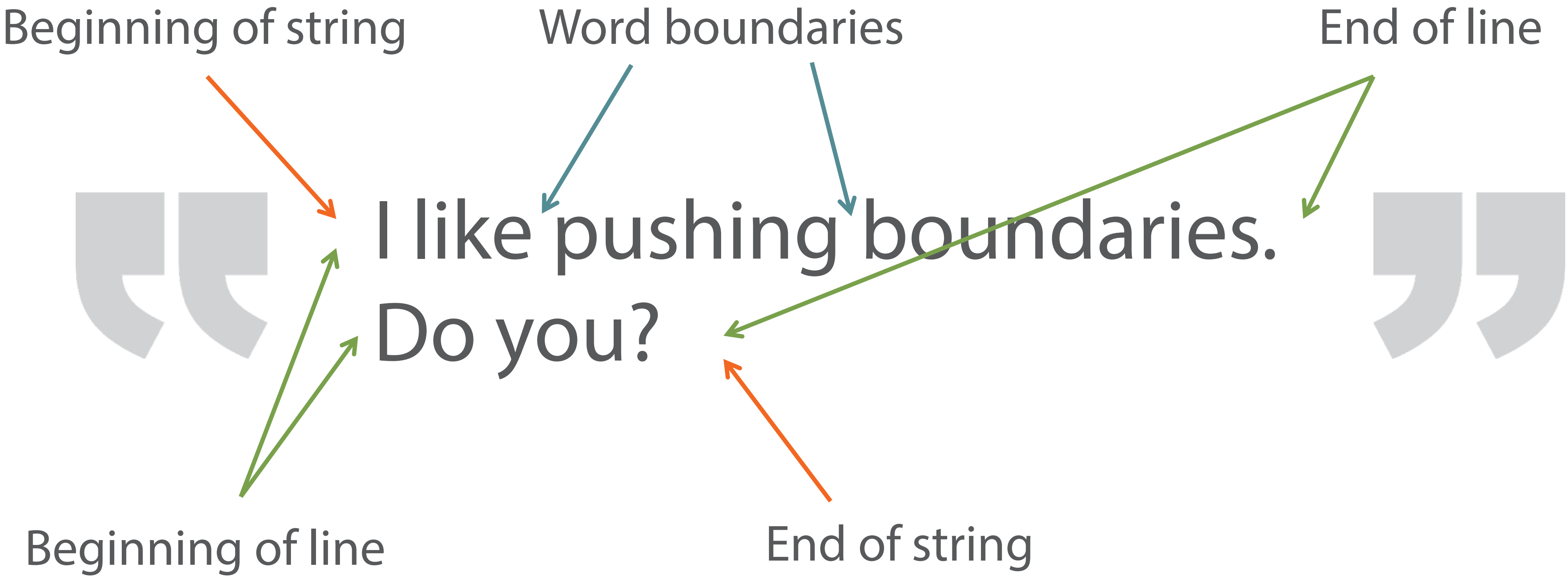
Anchors and Boundaries

Photo by Miguel A.C. Domingo

Know

Your

Boundaries



Beginning of string

Word boundaries

End of line

I like pushing boundaries.
Do you?

Beginning of line

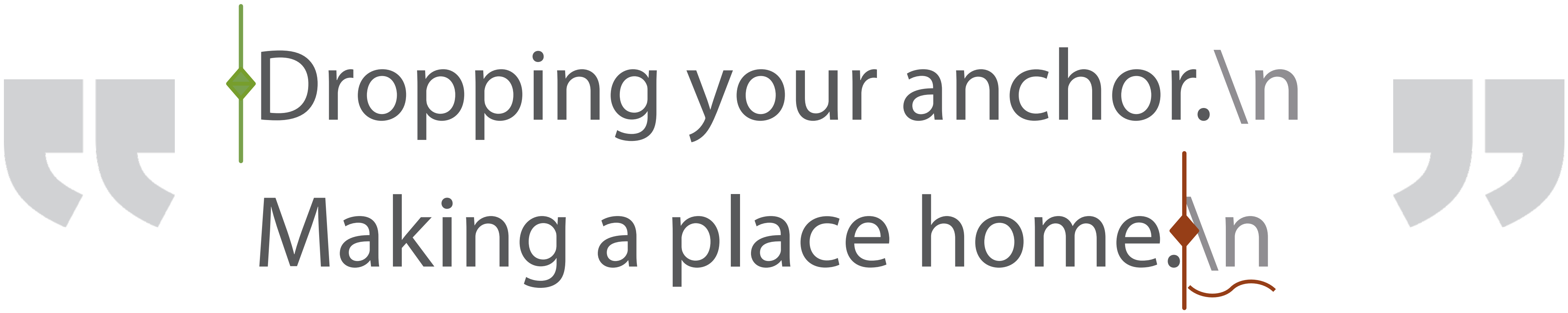
End of string

Zero-width

✗ Capture

✗ Quantify

Anchors



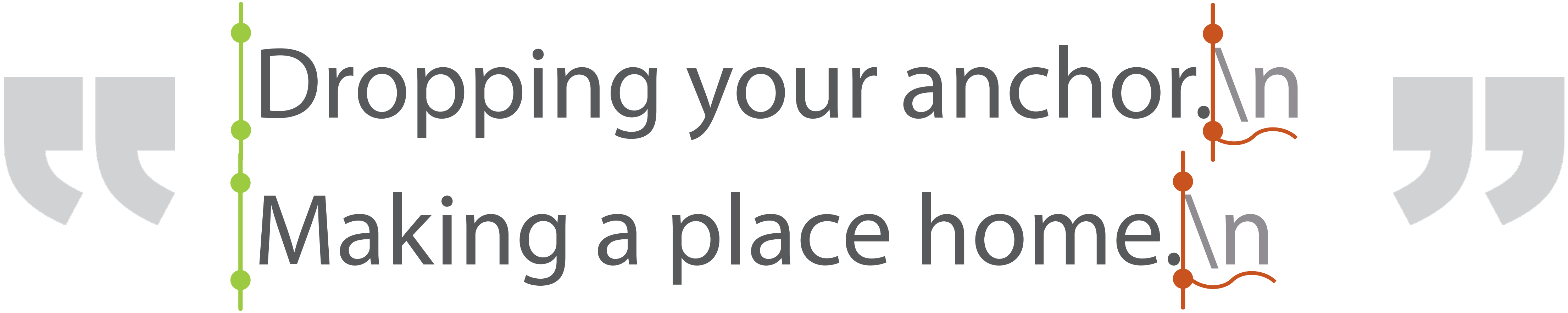
Dropping your anchor.\n

Making a place home.\n

^ - Start of string

$ – End of string *or \n at end of string*

Anchors



Dropping your anchor.\n
Making a place home.\n

^ - Start of string
- Start of line in multiline-mode

$ – End of string *or \n at end of string*
– End of line *or \n at end of line*

Anchors

> Dropping your anchor.\n
> Making a place home.\n

^ - Start of string
  - Start of line in multiline-mode

$ – End of string *or \n at end of string*
  – End of line *or \n at end of line*

\z – End of string

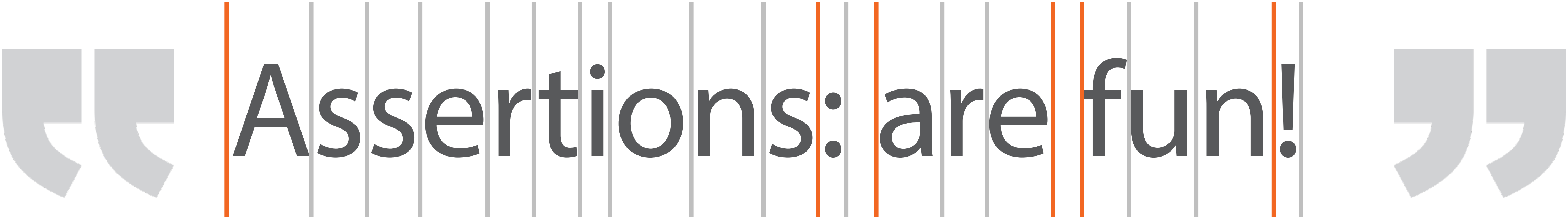
Anchors

^ $

\z

- ^ $ Bound to string *òr line*
- $ Might match \n at end ⚠
- Can be combined or used independently
- Can be part of alternate branch(es)

Word Boundaries



“Assertions: are fun!”

\b - Word boundaries

\B – Non-word boundaries

Meta Characters

Escaping & the Backslash

Backslash Escaping

\

- Remove special meaning from meta-characters
- Give special meaning to ordinary characters

Escaping Meta Characters

Special Meaning

[ ] ( ) | . ? * + { } ^ $ \   / *(delimiter)*

Literals

\[\] \(\) \| \. \? * \+\{ \} \^\$\\   \/

Photo by Renato Targa

Control characters

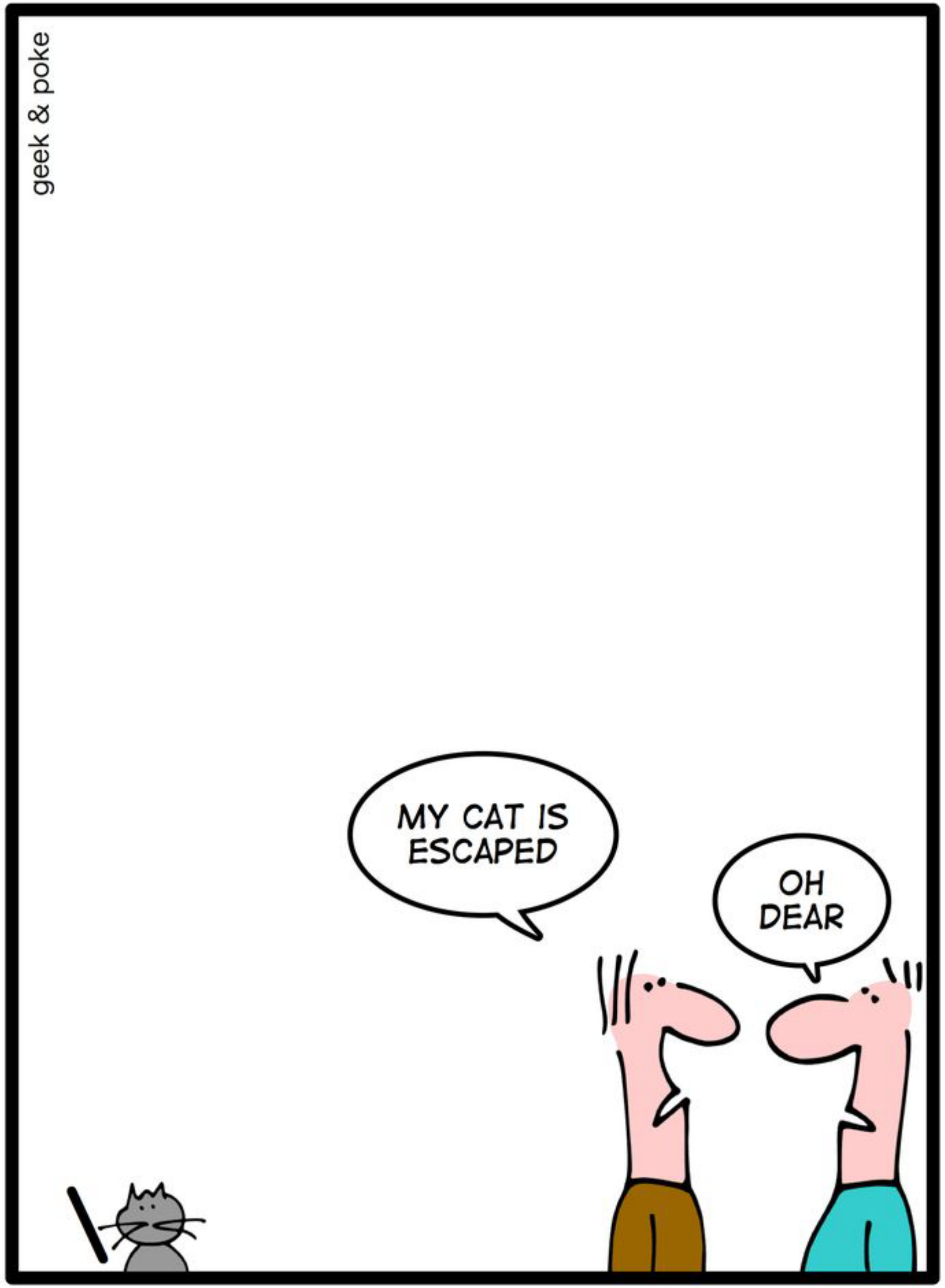
Hex	Escape sequence	Represents
0	\0	Null
7	\a	Bell
8	\b	Backspace
9	\t	Horizontal tab
10	\n	Line feed
11	\v	Vertical tab
12	\f	Form feed
13	\r	Carriage return
27	\e	Escape

Escaping

Programming Languages

vs.

Regular Expressions



Escape and Escape Again

```
$regex = "[\r\n\t]+";

$regex = \([0-9]+\+[0-9]+\)\*[0-9]+;

$regex = /http:\/\//;

$regex = ' attr=["'][^'"]+["']';
```

Escape and Escape Again

```
$regex = "[\\r\\n\\t]+";

$regex = \\([0-9]+\\+[0-9]+\\)\\*[0-9]+;

$regex = /http:\\/\\//;

$regex = ' attr=["\'][^\'"]+["\']';
```