CS 1520: Recitation 3

Regular Expressions

What is a Regex?

- A special text string for describing a search pattern.

- Applications
  - Did the user enter a proper email id, username, password, etc?
  - Anonymize all the phone numbers in a given dataset of users in Pittsburgh

- Cheat Sheet
  - https://www.debuggex.com/cheatsheet/regex/javascript
  - https://www.cheatography.com/savagedesign/cheat-sheets/javascript-regexp/

Examples

- Write a Regex for all the phone numbers in USA
  - (222) 345 4444, 232-446-4444, 212.566.7777, (676).677.8888,….

- \(?\d{3}\)?[-.]\d{3}[-.]\d{4}

- But the above regex accepts phone numbers with non-matching parenthesis too, like (222 333 4444.

- Solution
  - (\(\d{3}\)[-.]\d{3}[-.]\d{4}|\d{3}[-.]\d{3}[-.]\d{4})

Examples

- Write a Regex for passwords that contain between 8-20 characters and can have, alphabets, special characters and digits

- ^[a-zA-Z -~]{8,20}$

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	&#032;	Space	64	40	100	&#064;	@	96	60	140	&#096;	`
1	1	001	SoH	33	21	041	&#033;	!	65	41	101	&#065;	A	97	61	141	&#097;	a
2	2	002	SoTxt	34	22	042	&#034;	"	66	42	102	&#066;	B	98	62	142	&#098;	b
3	3	003	EoTxt	35	23	043	&#035;	#	67	43	103	&#067;	C	99	63	143	&#099;	c
4	4	004	EoT	36	24	044	&#036;	$	68	44	104	&#068;	D	100	64	144	&#100;	d
5	5	005	Enq	37	25	045	&#037;	%	69	45	105	&#069;	E	101	65	145	&#101;	e
6	6	006	Ack	38	26	046	&#038;	&	70	46	106	&#070;	F	102	66	146	&#102;	f
7	7	007	Bell	39	27	047	&#039;	'	71	47	107	&#071;	G	103	67	147	&#103;	g
8	8	010	Bsp	40	28	050	&#040;	(	72	48	110	&#072;	H	104	68	150	&#104;	h
9	9	011	HTab	41	29	051	&#041;	)	73	49	111	&#073;	I	105	69	151	&#105;	i
10	A	012	LFeed	42	2A	052	&#042;	*	74	4A	112	&#074;	J	106	6A	152	&#106;	j
11	B	013	VTab	43	2B	053	&#043;	+	75	4B	113	&#075;	K	107	6B	153	&#107;	k
12	C	014	FFeed	44	2C	054	&#044;	,	76	4C	114	&#076;	L	108	6C	154	&#108;	l
13	D	015	CR	45	2D	055	&#045;	-	77	4D	115	&#077;	M	109	6D	155	&#109;	m
14	E	016	SOut	46	2E	056	&#046;	.	78	4E	116	&#078;	N	110	6E	156	&#110;	n
15	F	017	SIn	47	2F	057	&#047;	/	79	4F	117	&#079;	O	111	6F	157	&#111;	o
16	10	020	DLE	48	30	060	&#048;	0	80	50	120	&#080;	P	112	70	160	&#112;	p
17	11	021	DC1	49	31	061	&#049;	1	81	51	121	&#081;	Q	113	71	161	&#113;	q
18	12	022	DC2	50	32	062	&#050;	2	82	52	122	&#082;	R	114	72	162	&#114;	r
19	13	023	DC3	51	33	063	&#051;	3	83	53	123	&#083;	S	115	73	163	&#115;	s
20	14	024	DC4	52	34	064	&#052;	4	84	54	124	&#084;	T	116	74	164	&#116;	t
21	15	025	NAck	53	35	065	&#053;	5	85	55	125	&#085;	U	117	75	165	&#117;	u
22	16	026	Syn	54	36	066	&#054;	6	86	56	126	&#086;	V	118	76	166	&#118;	v
23	17	027	EoTB	55	37	067	&#055;	7	87	57	127	&#087;	W	119	77	167	&#119;	w
24	18	030	Can	56	38	070	&#056;	8	88	58	130	&#088;	X	120	78	170	&#120;	x
25	19	031	EoM	57	39	071	&#057;	9	89	59	131	&#089;	Y	121	79	171	&#121;	y
26	1A	032	Sub	58	3A	072	&#058;	:	90	5A	132	&#090;	Z	122	7A	172	&#122;	z
27	1B	033	Esc	59	3B	073	&#059;	;	91	5B	133	&#091;	[	123	7B	173	&#123;	{
28	1C	034	FSep	60	3C	074	&#060;	<	92	5C	134	&#092;	\	124	7C	174	&#124;	|
29	1D	035	GSep	61	3D	075	&#061;	=	93	5D	135	&#093;	]	125	7D	175	&#125;	}
30	1E	036	RSep	62	3E	076	&#062;	>	94	5E	136	&#094;	^	126	7E	176	&#126;	~
31	1F	037	USep	63	3F	077	&#063;	?	95	5F	137	&#095;	_	127	7F	177	&#127;	Delete

charstable.com

http://www.charstable.com/

Greedy and Lazy Evaluation

- Write Regex for all html tags in the given code snippet

   This is CS 1520

   This is the last thing I am doing today

- Regex:
- <.+>
- <.+?>

Greedy and Lazy Evaluation

- Write Regex for all html tags in the given code snippet

   This is CS 1520

   This is the last thing I am doing today

- Regex:

- ~~<.+>~~
- **<.+?>**

Regex For A Date

- Write the Regex for a date in the format yyyy/mm/dd or yyyy.mm.dd

- \d{4}([.\/])\d{2}([.\/])\d{2}

- Is this enough? Are all the matched dates valid?
  - No. Because it will also accept invalid dates like 1900/40/40

Regex For A Date

- Write the Regex for valid dates from 1900-01-01 through 2099-12-31

- ^(19|20)\d\d[- /.](0[1-9]|1[012])[- /.](0[1-9]|[12][0-9]|3[01])$

- Is this enough?
  - No
  - Doesn’t test for non-leap year dates, etc
  - Also, allows usage of inconsistent delimiters in the date

Regex Groups

Non-capturing group

- Final Validation of a date
- Capture the year, month and day and check for validation
- ^((?:19|20)\d\d)[- \/.](0[1-9]|1[012])[- \/.](0[1-9]|[12][0-9]|3[01])$
- Here, we have
  - $1 = year, $2 = month, $3 = day of the *yyyy-mm-dd* format date

Regex Groups

- Anonymize all the phone numbers in slide 3, in the format ddd-***-****

- \(?(\d{3})\)?[-.](\d{3})[-.](\d{4})

- Replace $2 = ‘***’ and $3 = ‘****’

Backreferencing

- How to prevent dates with inconsistent delimiters?
  - Prevent dates like 1990/12.01

- ^(19|20)\d\d([- /.])(0[1-9]|1[012])\2(0[1-9]|[12][0-9]|3[01])$

Using Regex in JS – test()

- regex.test(str)

```
const string = "222-111-3344"
const regex = /\(?\d{3}\)?[-. ]\d{3}[-. ]\d{4}/g;
const doesExist = regex.test(string);
console.log(doesExist);
```

- Output: true

Using Regex in JS – match()

- str.match(regex)
- Returns the matched strings

```
> var s = "my phone no are 111-2222 and 111-2233"
< undefined
> var r = /(\d{3})[-.]\d{4}/;
< undefined
> s.match(r)
< ▶ (2) ["111-2222", "111", index: 16, input: "my phone no are 111-2222 and 111
  -2233", groups: undefined]
>
```

Problem with match()

- Does not return the groups with a global flag

```
> var s = "my phone no are 111-2222 and 111-2233"
< undefined
> var r = /(\d{3})[-.]\d{4}/g;
< undefined
> s.match(r)
< ▶ (2) ["111-2222", "111-2233"]
>
```

Using Regex in JS – exec()

- regex.exec(str)
- Unlike match(), returns the groups even with a global flag

```
> var s = "my phone no are 111-2222 and 111-2233"
< undefined
> var r = /(\d{3})[-.]\d{4}/g;
< undefined
> r.exec(s)
< ▶ (2) ["111-2222", "111", index: 16, input: "my phone no are 111-2222 and 111
  -2233", groups: undefined]
> r.exec(s)
< ▶ (2) ["111-2233", "111", index: 29, input: "my phone no are 111-2222 and 111
  -2233", groups: undefined]
> r.exec(s)
< null
```

References

- https://www.youtube.com/watch?v=7DG3kCDx53c&list=PLRqwX-V7Uu6YEypLuls7iidwHMdCM6o2w
- https://www.regular-expressions.info/dates.html
- https://www.talentcookie.com/2015/07/lets-practice-regular-expression/
- https://regex101.com/