Regular Expressions: single characters

Char	Meaning	Example	matches	*non*-matches
`.`	Any one character except newline	`h.t`	hat hst	ht
most chars	Match itself (use \ to suppress any special regexp meaning)	`\(`	(	y
`\d` and `\D`	digit and non-digit, resp.	`\d\d\D`	10l	lol
`\w` and `\W`	word char or non-word char, resp.: letter, digit, underscore	`\w\W\w`	a&7 _-_	-&a
`\s` and `\S`	any (non)whitespace, resp.	a\sb	a b a b	aXYZb ab a XYZ b
`^` and `$`	Start and end of input, resp.	`^h.t$`	hat hst	zhatz huts
`[…]`	Any one character in the range	`[A-Fw-z]`	C z	BCD Bz
`[^…]`	Any one character *not* in the range	`[^A-Fw-z]`	G !	C z

More (including POSIX char classes) on the web.

Regular Expressions: combinations

regexp	Meaning	Example	matches	*non*-matches
α*	0 or more of preceding regex α	`a*b`	aaab b	abb
α+	1 or more of preceding regex α	`a+b`	aaab ab	abb b
α?	0 or 1 of the preceding regex α	`ab?c`	abc ac	abbc c
α{*m*,*n*}	*m* to *n* of preceding regex α If one number is omitted (but comma there), default values are 0,+inf.0	`a{3,5}b`	aaab aaaab aaaaab	aaaaaab abab aab
(…)	Precedence: Group as a single regex	`a(bc)*d`	ad abcbcd	abccd
αβ	Concatenation: a match for α followed by a match for β.	(all the above!)		
α|β	OR: a match for α followed by a match for β. (N.B. Low precedence.)	a*b|c*d	aaab cccd b	abd

Regular Expressions (Common)

simplistic e-mail address	`\S+@\S+\.\S+`	Check for an @ and a . and allow nonwhitespace characters only. (allows "4@5.!" though.)
poor password	`\w+`	Any sequence of word characters. (Disallows puncuation!)
specific-length password	`\w{4,10}`	Any sequence of word characters at least 4 characters long but no more than 10 characters long
advanced password	`[a-zA-Z]\w{3,9}`	A four to ten character password starting with a letter.
another adv. password	`[a-zA-Z]\w*\d+\w*`	A password starting with a letter, containing only letters/digits/underscores, with at least one digit.
limited-length field	`\S{4,10}`	Any set of nonwhitespace characters at least 4 characters long but no more than 10 characters long
SSN	`\d{3}-\d{2}-\d{4}`	Exactly three digits, followed by a dash, followed by exactly two digits, followed by a dash, followed by exactly four digits

Regular Expressions (you try…)

RU SID		Radford University Six digit ID number. The number cannot start with a zero.
Phone number		Phone number in the form (999) 999-9999 (how to have the space be optional?)
Phone number		Allow any of the three common formats: 999-999-9999, or (999)999-9999, or just 999-9999
URL		Universal Resource Locator that limits the protocol to http or https and requires a host name.
Another Email address		An email address that starts with a letter, has a length of at least three word characters to the left of the @ sign, has a domain name that starts with a letter and has a length of at least three word characters, has the requred . after the domain name and has a top level domain of either two or three letters

Regular Expressions: some false starts

For URL: What is wrong with starting out:

- `http|https:\w+…`
- `https*:…`

For a phone number in any of three common forms:
What is wrong with starting out:

- `\(?\d{3}\)?-` (all enclosed in `(…)?` to make it optional)

Telephone sol'n one, step-by-step:

- take the easy part: all three end in `\d{3}-\d{4}`
- that's optionally preceded by area code: `(…)?\d{3}-\d{4}`
- area codes are one of two forms: `((…)|(…))?\d{3}-\d{4}`
- now just spec those two forms:

```
((\d{3}-)|(\(\d{3}\) ?))?\d{3}-\d{4}
```

How does this solution compare to a brute force choice of three:

```
(…)|(…)|(…)
```