

## Regular Expressions

character	description
abcABC123	Themselves literally (any letters or digits)
<b>character classes</b>	
.	anything
\w	letters, numbers, or underscores (opposite is \W)
\d	numbers (opposite is \D)
\s	whitespace (opposite is \S)
<b>repetition</b>	
?	optional
*	zero or more
+	one or more
{n}	exactly n
{n,}	n or more
{n,m}	n to m times
<b>any/none of</b>	
[abc]	a, b, or c
[a-zA-Z]	any of abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ
[^abc]	anything that's not a, b, or c
<b>anchors</b>	
^	start
\$	end
\b	word boundary
<b>groups</b>	
(a)	Capture Group (referenced with \1)

Most anything that's not `a-zA-Z0-9` must be *escaped* to match the character itself literally.

### Functions in the `re` module

- `compile(regex)`: prepare a regex ahead of time
- `search(regex, string)`: test if a regular expression matches
- `findall(regex, string)`: find all the matches
- `sub(regex, replacement, string)`: substitute with regex

### Python Regex Flags<sup>1</sup>

- `re.MULTILINE`: make `^` and `$` match the beginning and end of lines<sup>2</sup>
- `re.IGNORECASE`: treat upper and lowercase the same
- `re.VERBOSE`: ignore whitespace in the regex

<sup>1</sup> Flags may be specified when compiling the regex, or by passing them as the final argument to any of the functions within the `re` module (but not in both places). Flags are combined with the `|` operator. e.g. `re.MULTILINE | re.VERBOSE`.

<sup>2</sup> that is, not just the entire string, but every line in the string. Useful for working with contents read from an external file.