REGEX[Regular Expressions]:

- PHP has 2 "languages" for Regex, POSIX & PCRE [Perl Compatible Regular Expressions]
  ↳ more powerful.
- PRCE functions start with "preg_"
- quick alternatives

  $x = strpos ($string, $findMe);

  x is the position where this appears first or (=== false).
- using a regex needs 2 delimiters ` ` ↼ anything inside will be searched for in the entire given string
- OR is | ex: `ab|cd` is true for abx, acd, xcd, xcdn, etc...
- Flags: added to the end of a regex that add options (can be stacked)

g	i	m
search globally for all occurances	ignore case sensetivity	multiple lines (ignore newline)

- Start/End Limiters: apply as placeholders

^	$	\b	\B
beginning of string	end of string	both beginning or end depending on where it is used	opposite of \b (char inside a word)

- Quantifiers: apply to item to the left

?	+	*	{x}	{x,y}	{x,}
0, 1	1 or more	0, 1, or more	x times	x to y times	x or more times

- Charachter Classes:

[xyz...]	[a-n]	[a-n0-7...]	[^...]
x or y or z or...	a to n	a to n or 0 to 7 or...	^ means <u>not this</u>

- Special Classes:

  [[:xxxx:]]

  ↳

alpha	letter	digit	number	graph	printable
lower	lowercase letter	punct	.?!,;:-[]{}()'"...	cntrl	escape
upper	uppercase letter	space	white space	xdigit	hexadecimal
alnum	alphanumeric	blank	space or tab	print	printable charachter except ctrl+n

- Metacharachters: need \ before them: [ ] ! ( ) { } ^ $? . + * \ #

  Note: doesn't apply inside [ ].

- Short classes:
  - . any charachter
  - \w [a-zA-Z0-9]
  - \W not \w
  - \d [0-9]
  - \D not \d
  - \s \t \n \r
  - \S not \s

- PCRE functions:
  - preg_match(string pattern, string subject, $result); ← optional

    true if pattern matches subject

    $result[0]: part of subject that satisfies pattern

    $result[1]: part that satisfies first ( ).

    etc...
  - preg_match_all(string pattern, string subject, $res); ex: $res[0][1]

    $res[0] is an array of all substrings of ↑ that match the pattern.
  - preg_replace(string p1, string p2, string subject, int limit); ← optional

    replaces occurances of p1 (up to limit) with p2
  - preg_quote(string subject); adds \ before metacharachters.