EXCEL REGEX CHEAT SHEET

RegEx Video & Practice File: https://bit.ly/regex-excel

Step-by-step Guide for Writing Regular Expressions

Writing effective regular expressions involves understanding the specific patterns you need to match.

1. **Identify the Text Pattern:** Determine the specific text or structure you need to find. For example, an email address or a date.

2. **Start Simple**: Begin with a basic pattern and gradually add complexity.

Example: To match a basic email address, start with the pattern for a sequence of word characters **[\w]+**

3. **Use Metacharacters and Classes:** Incorporate special characters and classes to match more complex patterns.

Example: An email address pattern can be:
[\w.-]+@[\w.-]+\.\w+

This pattern matches an email address format, consisting of one or more word characters, dots, or hyphens, followed by an @ symbol, then one or more word characters, dots, or hyphens, a dot, and finally one or more word characters.

4. **Test and Refine:** Use regex testing tools like regex101 to test your patterns against various strings and refine them as needed.

Common RegEx Tokens

Description	Token
A single character of: `a`, `b`, or `c`	[abc]
A character except: `a`, `b`, or `c`	[^abc]
A character in the range: `a-z`	[a-z]
A character not in the range: `a-z`	[^a-z]
A character in the range: `a-z` or `A-Z`	[a-zA-Z]
Any single character	.
Alternate - match either `a` or `b`	a | b
Any whitespace character	\s
Any non-whitespace character	\S
Any digit	\d
Any non-digit	\D
Any word character	\w
Any non-word character	\W
Match everything enclosed	(?:...)
Capture everything enclosed	(...)
Zero or one of `a`	a?
Zero or more of `a`	a*
One or more of `a`	a+
Exactly 3 of `a`	a{3}
3 or more of `a`	a{3,}
Between 3 and 6 of `a`	a{3,6}
Start of string	^
End of string	$
A word boundary	\b
Non-word boundary	\B

Excel REGEX Function Arguments

Text: the input text string.
Pattern: the regular expression pattern to extract.
Return mode: Optional. Determines the return format:
- 0: First match (default)
- 1: Multiple matches as an array
- 2: Groupings for the first match as an array

Ignore case: Optional. If TRUE (default), the match is case-insensitive.
Replacement: The replacement text.
Occurrence: Optional. Determines which occurrences are replaced:
- 0: All occurrences (default)
- n: nth occurrence from the start
- -n: nth occurrence from the end

REGEXREPLACE– Replaces substrings matching a specified pattern with a replacement string.

Syntax: =REGEXREPLACE(text, pattern, replacement, [occurrence], [ignore_case])

Example 1: Redact the first 6 digits of a phone number.

=REGEXREPLACE("My phone number is 123-456-7890", "\d{3}-\d{3}", "XXX-XXX")

Returns: My phone number is XXX-XXX-7890

Example 1: Replace the first 5 digits of a Social Security Number (SSN) with asterisks

=REGEXREPLACE("My SSN is 123-45-6789", "\d{3}-\d{2}", "*-")

Returns: "My SSN is *--6789"

REGEXEXTRACT – Extract substrings that match a specified pattern from the input text.

Syntax: =REGEXEXTRACT(text, pattern, [return_mode], [ignore_case])

Example 1: Extract the domain name from a URL

=REGEXEXTRACT("https://www.MyOnlineTrainingHub.com/blog", "(?<=//)(?:www\.)?([^/]+)")

Returns: www.MyOnlineTrainingHub.com

Example 2: extract the email address from the text in the referenced cell, regardless of where the email exists in the text string.

=REGEXEXTRACT("Contact support@example.com", "[\w.-]+@[\w.-]+\.\w+")

Returns: support@example.com

REGEXTEST – Test if a text string matches a specified pattern

Syntax: =REGEXTEST(text, pattern, [ignore_case])

Example 1: Test if a text string is present

=REGEXTEST("Hello World", "world")

This will return TRUE because the function ignores case by default.

Example 2: Validate whether a string is a correctly formatted email address.

=REGEXTEST("user@example.com", "^[\w.-]+@[\w.-]+\.\w+$")

Returns: TRUE