# ***Regular Expression Cheat Sheet*** *for Paratext and RegEx Pal*

| Reg Ex Function | Description | Alternate expression | sample expression | matching explanation |
|---|---|---|---|---|
| `\` | Escape character— do not know what it is until the next character (Reg Ex metacharacter or actual character). | | `\\`<br>`\s` | means the \ character<br>means any whitespace |
| | **White space** | | | |
| `\r\n` | Carriage return and linefeed (end of line) | | `\r\n` | is both parts of a line break |
| `\s` | Any whitespace character (including nobreak, thin, en, em spaces, etc.) | `[ \r\n\s\t]` | `\s` | match: is the [carriage & linefeed return] |
| | **Range of characters** | | | |
| `[x-y]` | Any one of the characters in the range specified within the brackets. | | `[a-cx-z]` | match: **a**,**b**,**c**,**x**,**y**, or **z** |
| `[^x-y]` | Any one of the character not in the range specified within the brackets. | | `[^a-cx-z]` | match: any thing that's NOT ***a***,***b***,***c***,***x***,***y***, *and* ***z*** |
| | **Character classes** | | | |
| `.` | Any character **except** linefeed in *RegExPal*. Any character in *ParaTExt*. An end of line consists of two parts, the carriage return "\r" and linefeed "\n". | `[^\n]` | | match: **is the** tiger [not the new line] |
| `\w` | Any word building character (letters & digits). | | | match: **Wá 1 sp**? |
| `\W` | Any non-word building character (not a letter and not a digit). | | | match: Wá 1 sp**?** |
| `[\w-[\d]]` | Any word-building character excluding digits. **Note:** "-" in front of embedded [ ] removes digits from the class. | `\p{L}` | | match: **Wá 1 sp**? |
| `\s` | Any whitespace character. | `[ \r\n\s\t]` | | *See \s above under Whitespace* |
| `\S` | Any non-whitespace character | `[^ \r\n\s\t]` | | match: **is the tiger** |
| `\d` | Any digit in any script | `\p{N}` | | match: **24**a১৭ |
| `\D` | Any character other than a digit. | `\P{N` | | match: 24**a**১৭ |
| `[...]` | Any one character between the [ ] | | `[abc]` | match: **abac**us |
| `[^...]` | Any one character not between the [ ] | | `[^abc]` | match: abac**us** |
| | **Environment**—Context, Anchors, Positioning (finds context but does not capture OR anchors at context) | | | |
| `(?=...)` | Followed by … (place expression after matched item) | | `a(?=\s)` | match: **a** when followed by a space h**a** tch, but not hat |
| `(?!...)` | Not followed by … (place expression after matched item) | | `a(?!\s)` | match: **a** when **not** followed by a space h**a**t, but not ha t |
| `(?<=...)` | Preceded by … (place expression before matched item) | | `(?<=\s)c` | match: **c** preceded by a space hat **c**atch, but not hatc |
| `(?<!...)` | Not preceded by … (place expression before matched item) | | `(?<!\s)t` | match: **t** not preceded by a space a**tt**est, tes**t**ing |
| `\b` | Word boundary. Positions to but does not capture the word boundary. | | `\bin\b` | match: word "**in**", but not "in" as part of a word as in: bin, or cinch |
| `\B` | Not a word boundary. Positions to but does not capture other word building characters. | | `\Bin\B` | match: b**in**ary, f**in**e, but not bin, inch, or in |
| | **Anchors** | | | |
| `^` | Start of record | `\A` | | |
| `$` | End of record | `\Z` | | |

a record is a ***chapter*** in *RegExPal*
a record is a ***book*** in *ParaTExt*

**Metacharacters**
\ ( ) { } . ? * + ^ $ |

when **finding** the actual character place a \ before the metacharacter
when **replacing** the actual character only the \ needs to be preceded by a \

In *ParaTExt* to use a regular expression in a *find*, ***press ctrl-f***, then in the find box key in: ***regex:*** immediately followed by the regular expression. Regular expressions **cannot** be used in the *replace*.

# ***Regular Expression Cheat Sheet*** *for Paratext and RegEx Pal*

| Reg Ex Function | Description | Alternate expression | sample expression | matching explanation |
|---|---|---|---|---|
| **Options**—switches | | | | |
| `(?i)` | Ignore case–Match either upper or lower case | | `(?i)a` Matches one **a** at a time | match: lower and uppercase **a** **A**d**a**ms **a**pple |
| `(?s)` | At start of expression dot also matches linefeed. | | `(?s)is.*tiger"` Matches every thing including a newline | match: **is the [carriage return & line feed tiger** |
| **Repetition** | | | | |
| `{n,m}` | Match the previous item at least *n* times but no more than *m* times. | | `xa{2,3}l` | match: **xaal** and **xaaal**, but not xal or xaaaal |
| `{n,}` | Match the previous item at least *n* times. | | `xa{2,}l` | match: **xaal**, **xaaal**, and **xaaal** but not xal |
| `{n}` | Match exactly *n* of the previous item. | | `a{2}` | match: only **aa** |
| `?` | Match 0 or 1 times of previous item (It does not or does exist) | `{0,1}` | `fa?ir` | match: **fir**, **fair**, and a**fir**k, but not faair |
| `*` | Match 0 or more occurrences of previous item until the **last** occurrence of that item. **GREEDY** | `{0,}` | `\\f .*\\f\*` | match: **\f a \fr 1.18 \ft first footnote\f* and more\f b \fr 1.18 \ft 2nd footnote\f*** |
| `*?` | Adding ? matches all occurrences of previous item until **first** occurrence of the next item. **NOT GREEDY** | `{0,}?` | `\\f .*?\\f\*` matches footnote a followed by footnote b | match: **\f a \fr 1.18 \ft first footnote\f*** and more**\f b \fr 1.18 \ft 2nd footnote\f*** |
| `+` | Match 1 or more occurrences of previous item until the **last** occurrence of that item. | `{1,}` **GREEDY** | `b(an)+a` | match: ur**banana** and **banana**na |
| `+?` | Match 1 or more occurrences of previous item until **first** occurrence of that item. | `{1,}?` **NOT GREEDY** | `b(an)+?a` | match: ur**bana**na and **bana**nana |

*Consider the following scripture text with 2 footnotes and with the start and ending footnote markers **underlined**:*

`\v 18 This is some scripture text`**`\f`** `a \fr 1.18 \ft first footnote`**`\f*`** `and more`**`\f`** `b \fr 1.18 \ft second footnote`**`\f*`** `until the end.`

*With 2 footnotes in the verse a greedy match for footnotes `\\f .*\\f\*` would match the start of the 1st all the way thru the end of the 2nd footnote:*

`\v 18 This is some scripture text`**`\f a \fr 1.18 \ft first footnote\f* and more\f b \fr 1.18 \ft second footnote\f*`** `until the end.`

*With 2 footnotes in the verse a non greedy match for footnotes `\\f .*?\\f\*?` would match first on footnote a and then on footnote b*

`\v 18 This is some scripture text`**`\f a \fr 1.18 \ft first footnote\f*`** `and more`**`\f b \fr 1.18 \ft second footnote\f*`** `until the end.`

| | | | |
|---|---|---|---|
| **Groups**—groups are numbered in order of "(" starting from the left. Don't include environment "( " as in "(?"". | | | |
| `(...)` | Match and capture what's in parenthesis **( )**, store in a group for later reference. Groups are numbered **\1–\9** based on sequence from left to right of open **(** . | EXAMPLE GROUP # \1 \2 \3<br>find: (?s)(?<=\\c \d\s+)(\\s .*?)(\s+)(\\r .*)<br>replace: \3\2\1 | Swap order of: \c, \s, \r to to \r, \s, \c.<br>**NOTE**: Parenthesis ( ) followed by a ? as in (?s) are a function and are not assigned a group #. |
| `\|` | Alternation. Match either side of the **\|** | `cat\|dog` | match: **cat**nip or **dog**ma. |
| `\1` | Match text captured in group 1— first set of ( ). You can reference up to 9 groups. | `c.(r\|e)\1_` | match: **carr**ion and **chee**ch, but not caret or cherish. |

A good Regular Expression Web Site — http://www.regular-expressions.info/unicode.html
Try out regular expressions Web Site — https://regex101.com/

**NOTE**: In *ParaTExt*
*//* is used to denote a line break often used in section heads
**~** is used to denote a non-breaking space.

## ***Regular Expression Cheat Sheet*** *for Paratext and RegEx Pal*

| Reg Ex Function | Description | | samples |
|---|---|---|---|
| **Unicode — \p and \P for matching and nonmatching Unicode expressions** | | | |
| \uFFFF | specific Unicode code point | | \u0301 combining acute<br>\u2013 en dash<br>\u201C left double quote |
| \p{L} | any letter (does not include numbers) | alternate expression [\w-[\d]] | not the same as \w, since \w includes numbers |
| \p{Ll} | any lowercase letter | | a-z, à, á, â , è, é, ŋ, ɓ, ɗ, ə, ʉ, etc. |
| \p{Lu} | any uppercase letter | | A-Z, À, Á, Â , È, É, Ɗ, Ɓ, Ɗ, Ə, Ʉ, etc. |
| **White space** | | | |
| \p{Z} | any white space character | | tab (\u, space, carriage return (\r), newline (\n), enspace(\u2002), etc. |
| \p{Zs} | any white space character that does not take up space | | Zero-width space (\u200b), etc. |
| **Numbers** | | | |
| \p{N} | any number in any script | 1 ١ 2 ٢ | includes roman, Arabic-Indic, ideographic, etc. |
| \p{Nd} | any non-ideographic digit | | includes roman, Arabic-Indic (١,٢,٣,…), etc. |
| \p{No} | *superscript* or *subscript* digit, or any digit not 0-9 (excluding ideographic digits) | | |
| **Combining characters** | | | |
| \p{M} | combining characters | | includes both **\p{Mc}** and **\p{Mn}** |
| \p{Mn} | zero width combining | | combining accents, circumflex, etc |
| \p{Mc} | combining characters that | | middle eastern vowels |
| **Punctuation** | | | |
| \p{P} | any punctuation characters | | |
| \p{Pd} | any kind of hyphen or dash | | includes hyphen, nobreak hyphen, en-dash, em-dash, figure-dash |
| \p{Ps} | any kind of open/left bracket | | includes braces **{ }**, square brackets **[ ]**, parenthesis **( )** |
| \p{Pe} | any kind of close/right bracket | | |
| \p{Pi} | any kind of opening quote | | Includes following open quotes: « ‹ ‘ ‚ “ „ ‘ ‘‘ |
| \p{Pf} | any kind of closing quote | | Includes following close quotes: » › ’ ” ’ ’’ |
| \p{Pc} | a punctuation character such as an underscore (low line) that connects words. | _ ‿ | abc_def |
| \p{Po} | any punctuation character that is not a dash, bracket, quote or connector. | | ?¿¡!,:; (to name a few) |

| ::: — Search within a search | | ONLY works in *RegExPal* | |
|---|---|---|---|
| ::: | match on the expression to the left of the :::, then within that match, match on the expression to the right of the ::: | \\xt [^\\]*:::; | Find cross references that contain a book/chapter separator.<br>• first match on **\xt** and its contents<br>• then on semicolon **;** "within **\xt** match" |

# ***Regular Expression Cheat Sheet*** *for Paratext and RegEx Pal*

| Regular Expression | in RegEx Pal, select | sample output | BACKGROUND—What are you doing?<br>INTERPRET EXPRESSION—Whatdoes it mean?<br>ANALYSIS—Interpret results |
|---|---|---|---|
| **RegEx Pal**—Insert Regular Expression via: File, USFM | | | |
| COUNT FOOTNOTE MARKUP<br>\\f .*?\\f* | Tools<br>Count/Extract<br>◉ count<br>◉ sort<br>◉ combine nonmarker text<br>Count marker patterns in footnotes (displays "x" for text. **NOTE**: Be consistent with what precedes \f*. No white space.) | 7: \f x \fr x \ft x \fq x\f*<br>1: \f x \fr x \ft x \fq x\ft x\f*<br>**1: \fr x \ft x\f***<br>9: TOTAL | **Objective:** Count and list footnotes and show the marker patterns collapsing all data in beteen markers into the letter x.<br>**\\f** finds the start of a footnote.<br>**.*?** is a non-greedy match of any character until you find the first occurrence of what follows the "?".<br>**\\f*** matches closing footnote marker (because it follows "*?" it's the first one following the open footnote marker).<br>**Analysis:** 1 footnote starts with **\fr** and is missing the **\f caller id**. |
| EXTRACT SECTION HEADS<br>\\s\d?.* | Tools<br>Count/Extract<br>◉ extract | **will find**<br>\s The Arrival of the Lover<br>\s2 The Adjuration Refrain<br>**will not find**<br>\sp The Beloved to Her…<br>\sc mss\sc* read… | **Objective:** List all section head markers. Include level number when it exists<br>**\\s\d?** finds `\s` marker followed by an optional number.<br>**.*** matches everything up to but not including the line break character.<br>**Analysis:** Without a space following the optional digit **\d?** and the anything character **.** will match **\sp** and **\sc**.<br>***NOTE:*** Insert space after **?** to match only section head levels. |
| COUNT CROSS REFERENCE MARKUP<br>\\x.*?\\x* | Tools<br>Count/Extract<br>◉ count<br>◉ sort<br>◉ combine non-marker text | 254: \x x \xo x \xt x\x*<br>**2: \x x \x***<br>**1: \xo x \xt x \x***<br>257: TOTAL | **Objective:** Count and list cross references and show the marker patterns collapsing all data in beteen markers into the letter x.<br>**\\x** finds a cross reference marker.<br>**.*?** is non-greedy match of any character until first occurrence of "\".<br>**\\x*** matches first closing cross ref.<br>**Analysis:** 2 cross refs are missing **\xo** and **\xt**.<br>1 cross ref is missing the opening cross ref marker **\x**. |
| COUNT ALL USFM<br>\\\w+*? | Tools<br>Count/Extract<br>◉ count<br>◉ sort<br>◉ combine non-marker text | 1: \c<br>1: \h<br>1: \id<br>1: \mt1<br>8: \p<br>14: \v<br>29: TOTAL | **Objective:** List all markers<br>**** find start of a marker \.<br>**\w+** find 1 or more letters/numbers for marker name.<br>***?** find optional end marker indicator.<br>**Analysis:** It's a one chapter book with header, main title, 14 verses, and 8 paragraphs. |