

# Blocks that Form our Language

## Expression Block:

Indicates the start of a statement. For internal input.

- Possible “Start” block - unique start of expression block

## Literal String:

Allows the user to input a string of characters that the regular expression will look for exactly as written (case-sensitive)

## Character Classes:

Block with checkboxes for ranges of characters

- A-Z
- a-z
- 0-9
- Special Characters
- Specify your own range
- Select all
- Negate, all characters except the specified (or a separate negate block)

## All Characters (.\*):

Block to capture a string of any length.

- Determine whether or not multi-line is allowed.
- May implement two (.\* ) blocks, one that includes multi-line and one that does not.

## Whitespace:

Block for /s

Checkboxes for

- All Whitespace
- Tabs
- Spaces
- Newline

## Loops:

Repeat Block. Block asks for a range of times something is allowed to loop. The second number in the range is allowed to be infinity.

- Contains an expression block that will take a statement to repeat.

## Assertion:

Beginning/End of input and/or line.

- If multi-line is enabled, it's end of input.
- If multi-line is disabled then it's the end of a line.

Word boundary - check for beginning or end of word (aka whitespace following). (/b or /B)

## Lookarounds / Disjunctions:

Or blocks. Takes a number of statements (indicated by expression blocks) and checks if any of them match.

## Capture Groups:

Stick exclusively to named capture groups for shorter RegEx if we use any capture groups, not a top priority for implementation.