Cheatography

Lua string patterns Cheat Sheet

by ambi (ambigious) via cheatography.com/195816/cs/41127/

Function Calls

```
string.fi nd( string, pattern, starti ndex, nopattern) # returns indices
string.gm atc h(s tring, pattern, starti ndex) # creates a iterator returning all matches
string.ma tch (st ring, pattern, starti ndex) # returns the matched string
string.gs ub( string, pattern, replace) # replace can be a function or table, can use %n to use n-th
captured element
```

Manual

Pattern Syntax

A *pattern* is a list of consecutive *pattern items*.
Adding a `^` at the start of a pattern anchors it to the start of the string, and `$` at the end. Otherwise, they have no special meaning.

Character Classes

`x`	Represents `x`, where is is not `^$()%.[ ]*+-?`
`%x`	Represents `x` where `x` can be anything
`%l`	Lowercase letters
`%u`	Uppercase characters (same as `%L`)
`%d`	Digits
`%x`	Hexadecimal digits (same as `[a-fA- F0-9]`
`%w`	Alphanumeric characters (same as `[a-Z0-9]`)
`%a`	Everything
`%s`	Space characters
`%p`	Punctuation characters
`%g`	Printable characters (not including space)
`%c`	Control characters
`[set]`	Union of all characters in the set, with possible character classes and RegEx style ranges
`[^set]`	*Inverse* of `[set]`

The uppercase variant represents the *inverse* of the set, so `%L` represents `%u`.
The dash can be included in sets by positioning it as the first or last character, and the square bracket as the first. Or just use a escape.

Pattern Item

A pattern item can be any of the following:

Character Class	**1** of the class
Character Class with *	**≥0** of the class, longest possible
Character Class with -	**≥0** of the class, shortest possible
Character Class with +	**≥1** of the class
Character Class with ?	**1 | 0** of the class
`%n`	Matches the n-th captured string
`%f[set]`	Frontier pattern
`%bxy`	Balanced match

Frontier patterns ::: They match an empty string `" "` that does *not* precede with the set, but *does* succeed with the set.
Balanced match ::: They match until the pair balances out. For example, `%(>` matches entirely `(The quick br((own f>ox jum >ped over the lazy dog>`.

Captures

Captures allow you specify groups. They are defined with parentheses, and they can include any valid pattern within.
Using pattern reference we can match the same character. Note that it is *not* reusing the pattern, it is reusing what the pattern matched. `([abcd])%1` is ***NOT*** the same as `[abcd] [abcd]`.
You can also use captures to return results from `string.find` after the indices, multiple results in `string.gmatch` and `string.match`, and use in the replacement string in `string.gsub`.
The empty capture `()` represents the current index in the string.

By **ambi** (ambigious)
cheatography.com/ambigious/

Published 5th November, 2023.
Last updated 5th November, 2023.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com