Sig

shortcutFoo
(/)

Home (/app/home) Dojos (/app/dojos) Tournaments (/app/tournaments) Achievements (/app/achievements) Profile (/app/profile) Pricing (/app/pricing)

< Learn These Shortcuts (/app/dojos/python-strings)

Python Strings Cheat Sheet

Cases I

s.capitalize()	Capitalize s # 'hello' => 'Hello'
s.lower()	Lowercase s # 'HELLO' => 'hello'
s.swapcase()	Swap cases of all characters in s # 'Hello' => "hELLO"
s.title()	Titlecase s # 'hello world' => 'Hello World'
s.upper()	Uppercase s # 'hello' => 'HELLO'

Sequence Operations I

s2 in s	Return true if s contains s2
s + s2	Concat s and s2
len(s)	Length of s
min(s)	Smallest character of s
max(s)	Largest character of s

Sequence Operations II

s2 not in s	Return true if s does not contain s2
s * integer	Return integer copies of s concatenated # 'hello' => 'hellohellohello'
s[index]	Character at index of s
s[i:j:k]	Slice of s from i to j with step k
s.count(s2)	Count of s2 in s

Whitespace I

s.center(width)	Center s with blank padding of width # 'hi' => ' hi '
s.isspace()	Return true if s only contains whitespace characters
s.ljust(width)	Left justifiy s with total size of width # 'hello' => 'hello '
s.rjust(width)	Right justify s with total size of width # 'hello' => ' hello'
s.strip()	Remove leading and trailing whitespace from s # ' hello ' => 'hello'

Find / Replace I

s.index(s2, i, j)	Index of first occurrence of s2 in s after index i and before index j
s.find(s2)	Find and return lowest index of s2 in s
s.index(s2)	Return lowest index of s2 in s (but raise ValueError if not found)
s.replace(s2, s3)	Replace s2 with s3 in s
s.replace(s2, s3, count)	Replace s2 with s3 in s at most count times
s.rfind(s2)	Return highest index of s2 in s
s.rindex(s2)	Return highest index of s2 in s (raise ValueError if not found)

Cases II

s.casefold()	Casefold s (aggressive lowercasing for caseless matching) # 'ßorat' => 'ssorat'
s.islower()	Return true if s is lowercase
s.istitle()	Return true if s is titlecased # 'Hello World' => true
s.isupper()	Return true if s is uppercase

Inspection I

s.endswith(s2)	Return true if s ends with s2
s.isalnum()	Return true if s is alphanumeric
s.isalpha()	Return true if s is alphabetic
s.isdecimal()	Return true if s is decimal
s.isnumeric()	Return true if s is numeric
s.startswith(s2)	Return true is s starts with s2

Splitting I

s.join('123')	Return s joined by iterable '123' # 'hello' => '1hello2hello3'
s.partition(sep)	Partition string at sep and return 3-tuple with part before, the sep itself, and part after # 'hello' => ('he', 'l', 'lo')
s.rpartition(sep)	Partition string at last occurrence of sep, return 3-tuple with part before, the sep, and part after # 'hello' => ('hel', 'l', 'o')
s.rsplit(sep, maxsplit)	Return list of s split by sep with rightmost maxsplits performed
s.split(sep, maxsplit)	Return list of s split by sep with leftmost maxsplits performed
s.splitlines()	Return a list of lines in s # 'hello\nworld' => ['hello', 'world']

Inspection II

s[i:j]	Slice of s from i to j
s.endswith((s1, s2, s3))	Return true if s ends with any of string tuple s1, s2, and s3
s.isdigit()	Return true if s is digit

s.isidentifier()	Return true if s is a valid identifier
s.isprintable()	Return true is s is printable

Whitespace II

s.center(width, pad)	Center s with padding pad of width # 'hi' => 'padpadhipadpad'
s.expandtabs(integer)	Replace all tabs with spaces of tabsize integer # 'hello\tworld' => 'hello world'
s.lstrip()	Remove leading whitespace from s # ' hello ' => 'hello '
s.rstrip()	Remove trailing whitespace from s # ' hello ' => ' hello'
s.zfill(width)	Left fill s with ASCII '0' digits with total length width # '42' => '00042'

< Learn These Shortcuts (/app/dojos/python-strings)

Blog (/blog) About (/about) Privacy Policy (/privacy) Terms of Service (/terms) Pricing (/app/pricing)

(https://www.facebook.com/shortcutfoo) (https://www.twitter.com/shortcutfoo)