Nota 3.4

`.lowercase`	lettere minuscole
`.uppercase`	lettere maiuscole
`.letters`	minuscole e maiuscole
`.digits`	`0123456789`
`.hexdigits`	`0123456789abcdefABCDEF`
`.punctuation`	`!"#$%&'()*+,-./:;<=>?@ [\]^_‘{|}~`
`.whitespace`	ASCII 9-13 e 32

Nota 3.6

`a.find(x)`	Indice della prima posizione in cui `x` appare in `a`; restituisce -1 se `x` non fa parte di `a`.
`a.find(x,i)`	Come `find`, ma con ricerca in `a[i:]`. L'indice trovato si riferisce ad `a`.
`a.find(x,i,j)`	Come `find`, ma con ricerca in `a[i:j]`. L'indice trovato si riferisce ad `a`.
`a.rfind(x)`	Come `find`, ma la ricerca inizia a destra, con l'indice comunque sempre contato dall'inizio della stringa.
`a.rfind(x,i)`	Come per `find`.
`a.rfind(x,i,j)`	Come per `find`.
`a.count(x)`	Indica quante volte `x` appare in `a`.
`a.count(x,i)`	Indica quante volte `x` appare in `a[i:]`.
`a.count(x,i,j)`	Indica quante volte `x` appare in `a[i:j]`.
`a.startswith(x)`	Vero se `a` inizia con `x`.
`a.startswith(x,i)`	Vero, se `a[i:]` inizia con `x`.
`a.startswith(x,i,j)`	Vero, se `a[i:j]` inizia con `x`.
`a.endswith(x)`	Vero, se `a` termina con `x`.
`a.endswith(x,i)`	Vero, se `a[i:]` termina con `x`.
`a.endswith(x,i,j)`	Vero, se `a[i:j]` termina con `x`

Nota 3.7

`a.strip()`	Si ottiene da `a`, eliminando spazi bianchi (caratteri appartenenti a `string.whitespace`) iniziali e finali.
`a.strip(s)`	Si ottiene da `a`, eliminando i caratteri iniziali e finali che appartengono ad `s`.
`a.lstrip()`	Come `strip`, eliminando però solo caratteri iniziali.
`a.lstrip(s)`	Come `strip`, eliminando solo caratteri iniziali.
`a.rstrip()`	Come `strip`, eliminando solo caratteri finali.
`a.rstrip(s)`	Come `strip`, eliminando solo caratteri finali.
`a.ljust(n)`	Se la lunghezza di `a` è minore di `n`, vengono aggiunti `n-len(a)` spazi alla *fine* di `a`.
`a.rjust(n)`	Se la lunghezza di `a` è minore di `n`, vengono aggiunti `n-len(a)` spazi all'*inizio* di `a`.

Nota 3.8

`a.split()`	Lista di stringhe che si ottiene spezzando `a`, utilizzando come stringhe separatrici le stringhe consistenti di spazi bianchi.
`a.split(s)`	Lista di stringhe che si ottiene spezzando `a`, usando `s` come separatrice.
`a.split(s,n)`	Al massimo `n` separazioni.

Nota 3.9

`a.isalpha()`	Vero, se ogni carattere di `a` è una lettera, fa cioè parte di `string.letters`.
`a.isdigit()`	Vero, se ogni carattere di `a` è una delle cifre 0,...,9.
`a.isalnum()`	Vero, se ogni carattere di `a` è una lettera o una delle cifre 0,...,9.
`a.islower()`	Vero, se tutte le lettere tra i caratteri di `a` sono minuscole.
`a.isupper()`	Vero, se tutte le lettere tra i caratteri di `a` sono maiuscole.
`a.isspace()`	Vero, se tutti i caratteri di `a` appartengono a `string.whitespace`.