Universidad Nacional de Jujuy
Facultad de Ingeniería

Introducción a la Programación
Programación I

Cadenas - *strings*

Contenido

- Introducción
- Concepto de cadena de caracteres
- Operaciones
- Rebanadas - *slicing*
- Caracteres de escape
- Métodos de las cadenas
- Expresiones regulares
- Librerías

Concepto de cadenas (*strings*)

Una cadena es una secuencia inmutable de caracteres Unicode, delimitada por comillas y se accede a cada caracter mediante un subíndice entre corchetes.

```
# Comillas simples
a = 'Aquí me pongo a cantar, al compás de la vigüela'
# Comillas dobles
b = "Los hermanos sean unidos porque ésa es la ley primera"
# Comillas triples
c = '''Yo no tengo en el amor
Quien me venga con querellas;
Como esas aves tan bellas
Que saltan de rama en rama
Yo hago en el trébol mi cama
Y me cubren las estrellas.'''
```


Operaciones

```
b = ' '
h = 'hOlA'
m = 'mUNdo'
c = input('Ingrese cadena: ') #cRuEL
s = h + b + m + b + c + b*2

print('Longitud de s:', len(s)) #18

#Inmutabilidad
s[4] = 'M'
#TypeError: 'str' object does not support item assignment
```


Recorrido

```
#Contar blancos
cont = 0
for car in s:
    if car == b:
        cont += 1
print(f'espacios: {cont}') #4

#Contar blancos bis
cont = 0
for i in range(len(s)):
    if s[i] == b:
        cont += 1
print(f'espacios: {cont}') #4
```


Método format()

El método format() formatea los valores especificados y los inserta dentro del marcador de posición de la cadena. Los marcadores de posición se pueden identificar mediante índices con nombre {precio}, índices numerados {0} o incluso marcadores de posición vacíos {}

```
txt = 'Mi nombre es {nombre}, tengo {edad} años'
print(txt.format(nombre = 'Juan', edad = 19))
txt = 'Me gusta {0} y {1}'
print(txt.format('viajar','nadar'))
txt = 'Soy estudiante {} de {}'
print(txt.format('estudiante','Ingeniería'))
txt = 'El binario de {0} es {0:b}'
print(txt.format(5))
precio = 49.0999
txt = "El precio es {:.2f} pesos"
print(txt.format(precio))
```

Caracteres de escape

print('\\hola\\')	\hola\
\'	Comilla simple
\\	Backslash
\n	Nueva linea
\r	Carriage Return
\t	Tab
\b	Backspace
\v	Vertical tab
\ooo	Octal value
\xhh	Hex value

Índices de un *string*

Los caracteres en un string están numerados con índices que inician en 0

```
nombre = 'Juan Cruz'
print(len(nombre)) # 9

#Para acceder a un caracter individual de un string:
print(nombre[0]) # J
print (nombre[-1]) # z
```

nombre	J	u	a	n		C	r	u	z
índice	0	1	2	3	4	5	6	7	8
índice	-9	-8	-7	-6	-5	-4	-3	-2	-1

Rebanadas - *Slicing*

Las rebanadas pueden tener 1, 2 o 3 parámetros str[i:f:p], donde i es inicio, f fin (-1) y p es el paso

```
nombre = 'Juan Cruz'
```

```
print(nombre[1]) # u
print(nombre[-4]) # C
print(nombre[1:3]) # ua
print(nombre[1:-1]) # uan Cru
print(nombre[:3]) # Jua
```

```
print(nombre[2:]) # an Cruz
print(nombre[:-1]) # Juan Cru
print(nombre[::2]) # Ja rz
print(nombre[1::2]) # unCu
print(nombre[::-1]) # zurC nauJ
```

nombre	J	u	a	n		C	r	u	z
índice	0	1	2	3	4	5	6	7	8
índice	-9	-8	-7	-6	-5	-4	-3	-2	-1

Métodos

En Python un método es una función que es definida con respecto a un objeto en particular. Para una cadena el objeto es un *string*
La sintaxis es:

<objeto>.<metodo>(<parametros>)

```
adn = 'ACGT'
pos = adn.find('T')
print(pos) #3
```


Métodos

s	'hOlA mUNdo cRuEL '
s.find('m')	5
s.rfind('u')	11
s.find('el')	-1
s.count(' ')	4
s.capitalize()	'Hola mundo cruel '
s.lower()	'hola mundo cruel '
s.upper()	'HOLA MUNDO CRUEL '
s.swapcase()	'HoLa MunDO CrUel '
s.title()	'Hola Mundo Cruel '

Métodos

s	'hOlA mUNdo cRuEL '
s.center(30, '.')	'.....hOlA mUNdo cRuEL .....'
s.ljust(30, '.')	'hOlA mUNdo cRuEL .....'
s.rjust(30, '.')	'.....hOlA mUNdo cRuEL '
s.zfill(30)	'000000000000hOlA mUNdo cRuEL '
t = s.replace('hOlA', ' Chau')	' Chau mUNdo cRuEL '
t.lstrip()	'Chau mUNdo cRuEL '
t.rstrip()	' Chau mUNdo cRuEL'
t.strip()	'Chau mUNdo cRuEL'

Métodos: separar - unir

```
a = '192.168.0.1'.split('.')
print(a) # ['192', '168', '0', '1']

b = ['red', 'green', 'blue']
print(' '.join(b)) # red green blue
print(''.join(b)) # redgreenblue
print('***'.join(b))# red***green***blue
```


Otros métodos

cad.isdigit()	True si cad es todo números
cad.isalnum()	True si cad es todo números o carácteres alfabéticos
cad.isalpha()	True si cad es todo carácteres alfabéticos
cad.islower()	True si cad es todo minúsculas
cad.istitle()	True si la primera letra de cada palabra es mayúscula
cad.isspace()	True si cad es todo espacios
cad.startswith(sub)	True si cad empieza con una (sub)cadena

Conversiones

- **ord**(texto) - convierte un string a un número.

  Ejemplo: **ord**("a") es 97, **ord**("b") es 98, …

Los caracteres mapean a números usando el estándar *ASCII* y *Unicode*.

- **chr**(número) - convierte a número a un string.

  Ejemplo: **chr**(99) es "c"

Expresiones regulares

Una limitación de las operaciones básicas de las cadenas es que no ofrecen ningún tipo de transformación usando patrones más sofisticados. Para eso vas a tener que usar el módulo **re** y aprender a usar expresiones regulares. El manejo de estas expresiones es un tema en sí mismo.

```
# Encontrar las fechas en el texto
import re
texto = 'Hoy es 22/05/2023. Mañana será 23/05/2023.'
lista = re.findall(r'\d+/\d+/\d+', texto)
print(lista) #['22/05/2023', '23/05/2023']

# Cambiar el formato
texto2 = re.sub(r'(\d+)/(\d+)/(\d+)', r'\3-\2-\1', texto)
print(texto2) #'Hoy es 2023-05-22. Mañana será 2023-05-23.'
```


Procesamiento del lenguaje natural

"El lenguaje sirve no sólo para expresar el pensamiento, sino para hacer posibles pensamientos que no podrían existir sin él." Bertrand Russell

El Procesamiento natural del lenguaje (NLP) mezcla Ciencia de la computación, inteligencia artificial y lingüística dedicándose a la generación y comprensión automática del lenguaje natural.

Librerías de Python para Procesamiento del Lenguaje Natural: NLTK, TextBlob, Stanford CoreNLP, Spacy, Textacy, Gensim, pyLDAvis,

Bibliografía

- https://www.w3schools.com/python/ref_string_format.asp
- https://docs.python.org/es/3/library/stdtypes.html#string-methods
- https://www.digitalocean.com/community/tutorials/python-string-functions
- https://realpython.com/python-strings/#built-in-string-methods
- https://docs.python.org/es/3/library/stdtypes.html#textseq
- https://docs.python.org/3/library/re.html
- https://www.aprendemachinelearning.com/procesamiento-del-lenguaje-natural-nlp/