

Strings

June 22, 2018

1 Resumen IIC-1103

Por: Nicolás Quiroz, Daniel Leal, Alfoso Irarrázaval.

2 Strings

Strings es un contenido base en programación. Básicamente, un string es una forma de guardar cualquier cosa, generalmente palabras, en python. Se crea con "" al rededor de el valor

```
In [36]: numero = '1'
         palabra = 'Frase'
         frase = 'Esto es una palabra'
         parrafo = """ Esto es un string multilinea por las tres comillas al comienzo y al fin
         commillas dobles y usando \n (ojo que si imprimen esto asi, se a crear un salto de li
         Se va a mostrar separado en lineas. Notarán que
         en algunos strings usamos comillas simples, y otras dobles. Para efectos de python, es
         se recomienda usar simples.
         """
```

Puedes hacer varias cosas con los strings, como por ejemplo, sacar un caracter

```
In [3]: print(palabra[0])
```

F

O varios

```
In [5]: print(frase[2:8])
```

to es

Divirlo por algun simbolo, como el espacio:

```
In [7]: print(frase.split(' '))
```

```
['Esto', 'es', 'una', 'palabra']
```

O por lineas

```
In [12]: print(parrafo.splitlines())
```

```
[' Esto es un string multilinea por las tres comillas al comienzo y al final. Tambien se puede
```

Tambien se pueden recorrer strings con for y while

```
In [15]: for caracter in palabra:
          print(caracter)
```

```
F
r
a
s
e
```

```
In [37]: vieja = palabra
          while palabra: # Equivalente a while len(palabra) != 0
              print(palabra[0])
              palabra = palabra[1:] # Las strings son inmutables, no puedo cambiarlas solo reasignar
          palabra = vieja # Para que vuelva a ser un string no-vacio
```

```
F
r
a
s
e
```

Y por ultimo podemos crear plantillas y formatearlas con valores.

```
In [22]: plantilla = '{} <- ahi se agrega algo >>{}<< aca tambien, ojo que no tiene por que ser string'
          print(plantilla.format(1,2,3,4))
```

```
1 <- ahi se agrega algo >>2<< aca tambien, ojo que no tiene por que ser strings, puede ser cualquier cosa
```

Algunos *slices* utiles:

```
In [38]: palabra_invertida = palabra[::-1]
          palabra_completa = palabra[:]
          palabra_menos_ultima = palabra[:-1]
          palabra_menos_pares = palabra[1::2]
          print(palabra_invertida)
          print(palabra_completa)
          print(palabra_menos_ultima)
          print(palabra_menos_pares)
```

```
esarF
Frase
Fras
rs
```

Algunos comandos utiles con strings

```
In [43]: print(numero.isdigit(), palabra.isdigit())
          print(numero.isalpha(), palabra.isalpha())
          print(palabra.lower(), frase.upper(), palabra.capitalize())
```

True False

False True

frase ESTO ES UNA PALABRA Frase