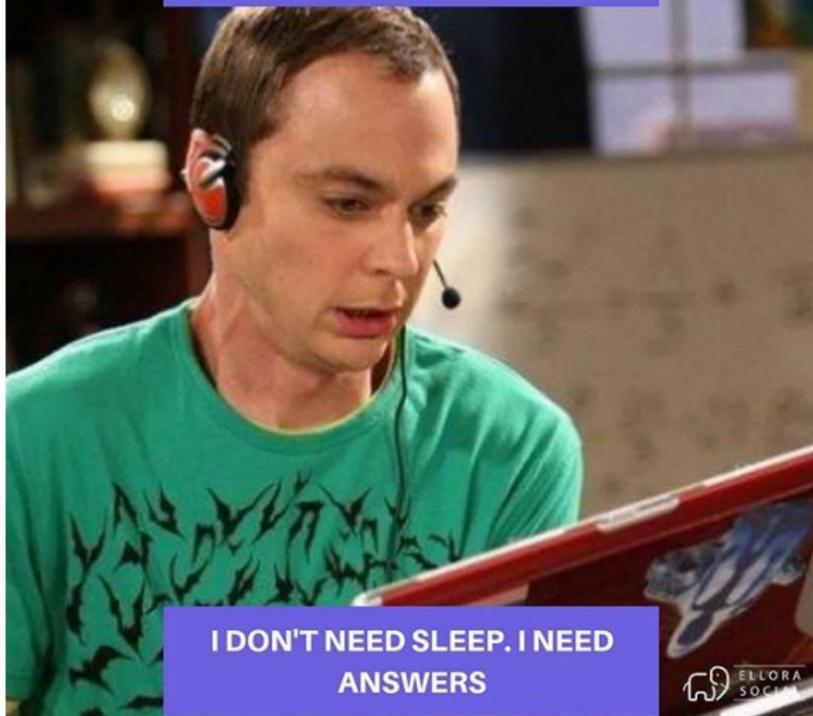


Unidad IV

Introducción a Python - Parte 2

27 de abril del 2020

**WHEN IT'S 3 A.M. AND YOUR
CODE STILL ISN'T WORKING**



**I DON'T NEED SLEEP. I NEED
ANSWERS**



Agenda

Strings

- String de una línea y multilinea
- strings como listas
- Largo de la cadena
- Remove espacios en blanco del comienzo y del final

- Mayúscula y minúsculas
- Reemplazo
- Split
- Union de cadenas
- Concatenación
- Contiene?
- Formato

Agenda

Strings

- String de una línea y multilinea
- strings como listas
- Largo de la cadena
- Remove espacios en blanco del comienzo y del final

Mayúscula y minúsculas

Reemplazo

Split

Union de cadenas

Concatenación

Contiene?

Formato

Strings

- ▶ Los strings, son cadenas de caracteres, y siempre se escriben con doble comillas, y comillas simple.
- ▶ Una cadena de caracteres la podemos visualizar a traves de la función `print()`

Agenda

Strings

String de una línea y
multilinea

strings como listas

Largo de la cadena

Remover espacios en blanco
del comienzo y del final

Mayúscula y minúsculas

Reemplazo

Split

Union de cadenas

Concatenación

Contiene?

Formato

String de una línea y multilinea

```
cadena = "esto es un string"  
print(cadena)
```

```
multilinea_caracter = """Esto es una cadena  
de caracteres que tiene  
multiples lineas. Estas se escriben  
con triple commilla.  
Muy parecido a los comentarios.  
"""
```

Agenda

Strings

String de una línea y
multilínea

strings como listas

Largo de la cadena

Remover espacios en blanco
del comienzo y del final

Mayúscula y minúsculas

Reemplazo

Split

Unión de cadenas

Concatenación

Contiene?

Formato

strings como listas

- ▶ Los strings son (de manera conceptual) listas.
- ▶ Al ser listas podemos hacer muchas operatorias interesantes

strings como listas

```
cadena = "Esto es una cadena"  
print(cadena[0])  
print(cadena[4])  
print(cadena[3:5]) # slice  
print(cadena[-5:-2]) # slice
```

Agenda

Strings

String de una línea y
multilinea

strings como listas

Largo de la cadena

Remove espacios en blanco
del comienzo y del final

Mayúscula y minúsculas

Reemplazo

Split

Union de cadenas

Concatenación

Contiene?

Formato

Largo de la cadena

- ▶ Para “contar” cuantos caracteres tiene un string, podemos usar la función `len()`

Largo de la cadena

```
cadena = "Esto es una cadena"  
print(len(cadena))
```

Agenda

Strings

String de una línea y
multilinea

strings como listas

Largo de la cadena

**Remove espacios en blanco
del comienzo y del final**

Mayúscula y minúsculas

Reemplazo

Split

Union de cadenas

Concatenación

Contiene?

Formato

Remover espacios en blanco del comienzo y del final

- ▶ Para eliminar los espacios en blancos del comienzo y final usamos el método *strip* de la clase `str`

Remover espacios en blanco del comienzo y del final

```
cadena = "    Esto es una cadena    "  
cadena.strip()
```

Agenda

Strings

- String de una línea y multilinea
- strings como listas
- Largo de la cadena
- Remover espacios en blanco del comienzo y del final

Mayúscula y minúsculas

- Reemplazo
- Split
- Union de cadenas
- Concatenación
- Contiene?
- Formato

Mayúscula y minúsculas

- ▶ Para convertir en mayúscula o minúscula a un string podemos usar `lower()` y `upper()` respectivamente.

Mayúscula y minúsculas

```
cadena = "Esto es una cadena"  
cadena.lower()  
cadena.upper()
```

Agenda

Strings

- String de una línea y multilinea
- strings como listas
- Largo de la cadena
- Remove espacios en blanco del comienzo y del final

Mayúscula y minúsculas

Reemplazo

Split

Union de cadenas

Concatenación

Contiene?

Formato

Reemplazo

- ▶ Para reemplazar un caracter o fragmento del string usamos *replace*.

Reemplazo

```
cadena = "Esto es una cadena"  
cadena.replace('a', 'e')  
cadena.replace('Esto es una cadena', 'hola mundo')
```

Agenda

Strings

- String de una línea y multilinea
- strings como listas
- Largo de la cadena
- Remover espacios en blanco del comienzo y del final

Mayúscula y minúsculas

Reemplazo

Split

Union de cadenas

Concatenación

Contiene?

Formato

Split

- ▶ Para separar en fragmentos la cadena podemos usar *split()* mandando como argumento, el caracter (o fragmento) que queremos utilizar para separar
- ▶ Por defecto va a intentar cortar con los espacios (' ')

Split

```
cadena = "Esto es una cadena"  
cadena.split()  
cadena2 = "Esto, es, una, cadena"  
cadena.split(',')
```

Agenda

Strings

String de una línea y
multilinea
strings como listas
Largo de la cadena
Remover espacios en blanco
del comienzo y del final

Mayúscula y minúsculas

Reemplazo

Split

Union de cadenas

Concatenación

Contiene?

Formato

Union de cadenas

- ▶ Podemos unir listas de string (por ejemplo las del resultado del slide anterior) y usar un caracter (o fragmento) para unir
- ▶ se llama escribiendo el caracter (o string) que se utilizará para unir y llamar el método *join()*

Union de cadenas

```
lista_string = ['Esto', 'es', 'una', 'cadena']  
,'.join(lista_string)
```

Agenda

Strings

String de una línea y
multilinea
strings como listas
Largo de la cadena
Remover espacios en blanco
del comienzo y del final

Mayúscula y minúsculas

Reemplazo

Split

Union de cadenas

Concatenación

Contiene?

Formato

Concatenación

- ▶ Podemos concatenar strings usando el operador `+`.

Union de cadenas

```
cadena1 = "hola"  
cadena2 = " mundo!"  
  
print(cadena1 + cadena2)
```

Agenda

Strings

- String de una línea y multilinea
- strings como listas
- Largo de la cadena
- Remover espacios en blanco del comienzo y del final

Mayúscula y minúsculas

Reemplazo

Split

Union de cadenas

Concatenación

Contiene?

Formato

Contiene?

- ▶ Podemos consultar si dentro de un string se encuentra un substring.

Contiene?

```
cadena = "Hola Mundo"  
if 'ola' in cadena:  
    print("si, ola esta en el string")
```

Agenda

Strings

- String de una línea y multilinea
- strings como listas
- Largo de la cadena
- Remover espacios en blanco del comienzo y del final

Mayúscula y minúsculas

Reemplazo

Split

Union de cadenas

Concatenación

Contiene?

Formato

Formato

- ▶ Podemos hacer más sencillo ingresar variables dentro de un string, haciendo uso del método *format()*

Formato

```
nombre = "Emmanuel"  
apellido = "Arias"  
print("Hola yo soy {}, mi  
apellido es {}".format(nombre, apellido))  
print("Hola yo soy {name}, mi apellido  
es {surname}".format(name=nombre,  
surname=apellido))
```