

Clase 1_1_Cadenas

March 17, 2023

1 Seminario de Lenguajes - Python

1.1 Temario

- Cadenas de caracteres
- Función `range()`

2 Tipos de datos

2.1 ¿Qué nos indica un tipo de datos?

- El tipo de datos nos indica **qué valores** y **qué operaciones** podemos hacer con una determinada variable.

3 ¿Qué tipos de datos vimos en la clase?

- Números: **int** y **float**
- Booleanos: **bool** (que mencionamos que eran números también)
- Cadenas de caracteres: **str**

La función **`type()`** nos permite saber de qué tipo es un determinado objeto referenciado por una variable.

```
[39]: x = True  
      type(x)
```

```
[39]: bool
```

4 También vimos que hay algunas conversiones de tipo implícitas y otras explícitas

```
[40]: mitad = 10 / 2  
      type(mitad)
```

```
[40]: float
```

```
[41]: mitad = int(10 / 2)
      type(mitad)
```

```
[41]: int
```

5 Las cadenas de caracteres

- Secuencia de caracteres encerrados entre comillas simples ' ' o comillas dobles " ".
- También se pueden definir con """ """.

```
[42]: mensaje_de_error = "ATENCION: la opción ingresada no es correcta."
      mensaje_de_error
```

```
[42]: 'ATENCION: la opción ingresada no es correcta.'
```

```
[44]: menu = """ Menú de opciones:
                1.- Jugar
                2.- Configurar el juego
                3.- Salir
            """
      print(menu)
```

```
Menú de opciones:
    1.- Jugar
    2.- Configurar el juego
    3.- Salir
```

6 Operaciones con cadenas de caracteres

- Concatenación: +
- Repetición: *
- Longitud de la cadena: len()

```
[45]: cadena = "Python "
      otra_cadena = "es lo más!"
      print(cadena + otra_cadena)
      print(cadena * 5)
      print(len(cadena))
```

```
Python es lo más!
Python Python Python Python Python
7
```

7 Algo más sobre cadenas de caracteres

- Cada elemento de la cadena se accede mediante un índice entre []

```
[48]: cadena = "Python"
      cadena[-2]
```

```
[48]: 'o'
```

- El índice puede ser negativo.

8 Subcadenas - slicing

```
[53]: #cadena[3:]
      cadena[:]
```

```
[53]: 'Python'
```

- El operador `:` permite obtener subcadenas. Esto se denomina **slicing**.
- El formato es **cadena[inicio:fin]**
- NO incluye al elemento cuyo índice es **fin**.
- `[:]` devuelve toda la cadena.
- Si los índices son negativos, se recorre de derecha a izquierda.

9 Probemos esto:

```
[54]: cadena[1] = 'm'
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[54], line 1
----> 1 cadena[1] = 'm'

TypeError: 'str' object does not support item assignment
```

- Las cadenas son INMUTABLES.

TypeError: 'str' object does not support item assignment

9.0.1 Tenemos que acostumbrarnos a leer los mensajes de error.

10 Algo más sobre cadenas de caracteres

- Ya mencionamos que en Python, todos son objetos.
- Si bien retornaremos a esto más adelante, podemos mencionar que los objetos tienen **propiedades y métodos**.
 - objeto.propiedad
 - objeto.metodo()
- Volviendo a las cadenas, algunos métodos que podemos utilizar son:

```
[56]: cadena = "Python es lo más!"  
      #cadena.upper()  
      cadena.lower()
```

```
[56]: 'python es lo más!'
```

```
[57]: cadena.islower()  
      #cadena.isupper()
```

```
[57]: False
```

10.1 Algo un poco más interesante:

```
[58]: cadena = "Somos campeones del mundo!!!!!"  
      cadena.count("!")
```

```
[58]: 6
```

```
[61]: cadena.center(70, "*")
```

```
[61]: '*****Somos campeones del mundo!!!!*****'
```

```
[66]: "  Somos campeones del mundo!!!!  ".strip()
```

```
[66]: 'Somos campeones del mundo!!!!'
```

11 Y un poco más...

```
[73]: cadena = "_camino"  
      #cadena.startswith("2")  
      cadena.endswith(("ar", "er", "ir"))
```

```
[73]: False
```

```
[76]: "Somos campeones del mundo!!!".split()
```

```
[76]: ['Somos', 'campeones', 'del', 'mundo!!!']
```

Probar: ¿de qué tipo es el objeto retornado por split?

- [+Info](#)

12 El operador in

- Este operador retorna True o False de acuerdo a si un elemento está en una colección o no.

- Las cadenas de caracteres son **secuencias de caracteres** por lo que puede utilizarse este operador.

```
[78]: palabra = input("Ingresá una palabra: ")
if "a" in palabra:
    print("Hay letras a.")
else:
    print("No hay letras a. ")
```

Ingresá una palabra: sol
No hay letras a.

13 El módulo string

- Python tiene un módulo denominado `string` que contiene mucha funcionalidad para la manipulación de cadenas.
- Para acceder a esta funcionalidad hay que **importarla**. Esto lo veremos en detalle más adelante.

```
[80]: import string
letras = string.ascii_letters
minusculas = string.ascii_lowercase
digitos = string.digits

digitos
```

```
[80]: '0123456789'
```

Ahora podemos tener otra solución al desafío que planteamos en clase: > Dado una letra ingresada por el teclado, queremos saber si es mayúscula o minúscula.

```
[83]: import string
minusculas = string.ascii_lowercase
mayusculas = string.ascii_uppercase

letra = input("Ingresar una letra: ")
if letra in minusculas:
    print("Es minúscula.")
elif letra in mayusculas:
    print("Es mayúscula.")
else:
    print("No es una letra.")
```

Ingresar una letra: 2
No es una letra.

14 Cadenas con formato

- Es posible definir cadenas con determinados formatos utilizando el método **format**.
- La forma general es:

`cadena.format(argumentos)`

- Observemos los siguientes ejemplos:

```
[84]: intentos = 5
      print('Hola {} !!! Ganaste! y necesitaste {} intentos!!!'.format("Lionel",
      ↪ intentos))
```

Hola Lionel !!! Ganaste! y necesitaste 5 intentos!!!

```
[85]: for num in "123":
      x = int(num)
      print("{0:2d} {1:3d} {2:4d}".format(x, x*x, x*x*x))
```

```
1   1   1
2   4   8
3   9  27
```

15 Los f-String

- Fueron introducidos a partir de la versión 3.6.
- Ver la [PEP 498](#)
- [+Info](#) en la documentación oficial
- Es una forma más sencilla de usar el format.

16 Un ejemplo

```
[ ]: intentos = 5
      nombre = "Lionel"
      print(f'Hola {nombre} !!! Ganaste! y necesitaste {intentos} intentos!!!')
      x = 4
      print(f"{x:2d} {x*x:3d} {x*x*x:4d}")
```

17 Algunas cosas interesantes

```
[86]: cad1 = "En Argentina nació"
      cad2 = "Tierra del Diego y Lionel"
      cad3 = "De los pibes de Malvinas"
      cad4 = "Que jamás olvidaré."

      print(cad1)
      print(cad2)
```

```
print(cad3)
print(cad4)
```

En Argentina nací
Tierra del Diego y Lionel
De los pibes de Malvinas
Que jamás olvidaré.

```
[89]: print(f"La mejor canción de todas:\n{cad1:<30}\n{cad2:>50}")
      print(f"\n{cad3:^30}")
      print(f"\n{cad4:*^50}")
```

La mejor canción de todas:
En Argentina nací
Tierra del Diego y Lionel
De los pibes de Malvinas
*****Que jamás olvidaré.*****

18 Un artículo sobre sistemas de codificación

[-Unicode & Character Encodings in Python: A Painless Guide](#)

19 Un desafío

Escribir un programa que ingrese 4 palabras desde el teclado e imprima aquellas que contienen la letra “r”. - **Pensar:** ¿podemos usar la instrucción **for** tal como la usamos hasta ahora para generar las 4 iteraciones?

- La sentencia **for** permite iterar sobre una **secuencia**.

```
for variable in secuencia:
    instrucción
    instrucción
    ...
    instrucción
```

```
[90]: cadena = "0123"
      for elem in cadena:
          print(elem)
```

0
1
2
3

20 Alguien podría pensar en plantear esto:

```
[91]: for i in "1234":
        cadena = input("Ingresá una palabra: ")
        if "r" in cadena:
            print(cadena)
```

```
Ingresá una palabra: arbol
arbol
Ingresá una palabra: casa
Ingresá una palabra: sol
Ingresá una palabra: r
r
```

Pero.. ¿sería una solución correcta? ¿Qué pasa si queremos ingresar 200 palabras? ¿O 2000?

21 La función range()

- Esta función devuelve una secuencia de números enteros.
- Puede tener de 1 a 3 argumentos:

```
range(valor_inicial, valor_final, paso)
```

- Es posible invocarla con uno, dos o los tres argumentos.

```
[98]: for i in range(4, 23, 3):
        print(i, end="-")
```

```
4-7-10-13-16-19-22-
```

22 Entonces, una mejor forma sería:

```
[100]: for i in range(4):
        cadena = input("Ingresa una palabra: ")
        if "r" in cadena:
            print(cadena)
```

```
Ingresa una palabra: csaa
Ingresa una palabra: arbol
arbol
Ingresa una palabra: sol
Ingresa una palabra: rosa
rosa
```