

# CADENAS DE CARACTERES - STRINGS

CS1100 - Introducción a Ciencia de la Computación  
UTEC

## Descifra los refranes

1. 🐘 📦 🚫 👁️ 😊
2. + 📄 👁️ 🖐️ ➡️ 100 ✈️
3. + 💡 😡 🖐️ 😊 ➡️ 😡
4. 🐒 👗 🖐️ ➡️ 🐒 ..
5. 😊 ? 🔪 🐱
6. ⚡ ☁️ ☔ 🖐️ 🌈
7. 🤝 zzz 🖐️ 🌊
8. 🍼 🐱 🖐️ 🔪 🐭
9. 🏠 🛠️ 🔪 🖐️ 🚪
10. 🚫 🔫 🖐️ 🍰 🐟
11. 😊 ⌚ ☀️ ☁️ 🙏 👍
12. 👤 👤 👤 👤 🖐️ 💪
13. 🖐️ 💋 🔪 🐟
14. 🚫 😞 📖 🖐️ 📖
15. 🚫 ✨ 🖐️ 💰
16. 🐶 🐶 🚫 😊

## Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Realizar programas que utilizan cadenas de caracteres.

# CADENAS DE CARACTERES



## CADENA (STRING)

**Un texto consiste de caracteres: letras, números, signos de puntuación, espacios en blanco, etc. Una cadena (string) es una colección ordenada de caracteres.**

# ¿Cómo se representa?

## CADENA (STRING)

**En Python, un string literal se denota encerrándolo entre un par de paréntesis simples o dobles.**

```
1 print ("Esto es un string")  
2 mensaje = 'Nos dijo: "cambien la pintura"'
```

**¿cuál será el resultado?**

# "Cadenas" : Operadores

## CADENA (STRING)

Las cadenas se pueden delimitar utilizando comilla simple, comilla doble o triple comilla:

```
1 cadena_simple = 'Ingrese nombre:'  
2 print (cadena_simple)
```



# "Cadenas" : Operadores

## CADENA (STRING)

**Las cadenas se pueden delimitar utilizando comilla simple, comilla doble o triple comilla:**

```
1 cadena_simple = "Mi nombre es: Carlos D'Alessio"
```

# “Cadenas”: operaciones

## CONCATENACION

- Dados 2 strings, podemos concatenarlos para formar un solo string.
- El resultado será todos los caracteres del primer string seguidos de todos los caracteres del segundo string.
- En Python utilizamos el operador +

```
1 primeraCadena = "hola"  
2 segundaCadena = "amigo"  
3 resultado = primeraCadena + " " + segundaCadena  
4 print(resultado)
```

## Da como resultado:

```
1 hola amigo
```



# "Cadenas": operaciones

## REPETICION

Podemos producir un string que es el resultado de repetir un mismo string varias veces.

### Ejemplo 1 :

```
1 asteriscos = "*" * 20
```

### El resultado es:

```
1 *****
```

### Ejemplo 2:

```
1 asteriscos = "Hola..." * 5
```

### El resultado es:

```
1 Hola...Hola...Hola...Hola...
```

# "Cadenas": uso de índices para acceder a una posición

0	1	2	3	4	5	6	7	8	9	10	11
M	o	n	t	y		P	y	t	h	o	n

[6:10]

```
1 s = 'Monty Python'
2 print (s[2])           ## n
3 print (s[:6]+s[6:8]+s[8:]) ## Monty Python
```

**NOTA:** El primer elemento de la cadena tiene índice 0.

# "Cadenas": Inmutables

Los string en Python son inmutables. No pueden ser cambiados.  
**NO** es válida una asignación:

```
1 word = 'Hola a todos'
2 word [2] = "x"
```

El resultado es:

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

**Ejemplo 2:** Si necesitamos un string distinto, debemos crear uno nuevo:

```
1 asteriscos = "Hola..." * 5
2
3 nuevo = "Escuchen: " + word[:4]
```

El resultado es:

```
1 Escuchen: Hola
```

# Ejemplo 1

Imprima todas las letras de la palabra 'saludos'

```
1 s = "saludos"  
2 for x in s:  
3     print(x)
```

Da como resultado:

```
1 s  
2 a  
3 l  
4 u  
5 d  
6 o  
7 s
```

Un string es iterable elemento por elemento.

## Ejemplo 2

**Imprima las veces en que la letra 's', aparece en la frase "Estamos desarrollando un código en Python"**

```
1 frase="Estamos desarrollando un código en python"
2 letra='s'
3 k=0
4 for i in frase:
5     if i==letra:
6         k=k+1
7 print("La letra ",letra,"aparece ",k," veces en la frase '", frase,"' ")
```

**Da como resultado:**

```
1 La letra  s aparece  3  veces en la frase ' Estamos desarrollando un
   código en python '
```

## Ejemplo 3

**Elige un refrán, forma una cadena con el refrán y luego:**

- a) Imprime las letras del refrán una por línea**
- b) Imprime las letras del refrán empezando desde la última letra hacia la primera letra, igual una por línea**
- c) ¿Cuántas letras tiene el refrán?**



## Ejemplo 3 - Solución

### a) Imprime las letras del refrán una por línea

```
1 refran = "No por mucho madrugar se amanece más temprano"  
2 for i in refran:  
3     print(i)
```

## Ejemplo 3 - Solución

**b) Imprime las letras del refrán empezando desde la última letra hacia la primera letra, igual una por línea**

```
1 refran = "No por mucho madrugar se amanece más temprano"
2 for i in range(len(refran)-1,-1,-1):
3     print(refran[i])
```

## Ejemplo 3 - Solución

### c) ¿Cuántas letras tiene el refrán?

```
1 refran = "No por mucho madrugar se amanece más temprano"
2 letras = 0
3 for i in refran:
4     if i != " ":
5         letras = letras + 1
6 print (f"Total de letras : {letras}")
```

# "Cadenas" : Funciones

## CADENA (STRING)

**Cuando queremos utilizar un número como cadena debemos utilizar la función `str()` que convierte el valor numérico en una cadena de caracteres:**

```
1 pi = 3.14
2 ##text = 'El valor de pi es: ' + pi
3 ## NO, no funciona
4 ## Debe ser:
5 text = 'El valor de pi es: ' + str(pi) ## SI
```

# "Cadenas" : Funciones

## LEN

Algunos de las funciones que se pueden aplicar a un string son:  
**len:** para hallar el largo de un string o cantidad de caracteres

```
1 >>> len('La vida es mucho mejor con Python.')
```

```
2 >>> 34
```

## "Cadenas" : Funciones

### FIND

Para determinar si un string está contenido en otro string.  
Devuelve el índice donde comienza el string hallado.

```
1 >>> "La vida es mucho mejor con Python".find("Python")
```

Da como resultado:

```
1 27
```

El resultado es 27 porque a partir de esa posición se encuentra el string "Python"



# "Cadenas" : Funciones

## FIND

Devuelve -1 si no halla el string.

```
1  
2 >>> "La vida es mucho mejor con Python.".find('Perl')  
3 -1
```

## "Cadenas" : Funciones

### REPLACE

Reemplaza el string a buscar por el indicado dentro del string grande

```
1  
2 >>> 'Mejor con Python'.replace('Python', 'Jython')  
3 'Mejor con Jython'
```

## "Cadenas" : Funciones

Dada la cadena: "soy un hacker de CS1100":

- a) Capitaliza el primer caracter de la cadena
- b) Convierta la cadena en minúscula
- c) Convierta la cadena en mayúscula
- d) Elimina los espacios en blanco de los costados
- e) Insertar tabulado



## "Cadenas" : Funciones

Dada la cadena: "soy un hacker de CS1100":

**a) Capitaliza el primer carácter de la cadena**

```
1 capitalize()
```

**b) Convierta la cadena en minúscula**

```
1 lower()
```

**c) Convierta la cadena en mayúscula**

```
1 upper()
```

**d) Elimina los espacios en blanco de los costados**

```
1 strip()
```

**e) Insertar tabulado**

```
1 \t
```



## "Cadenas" : Funciones

Dada la cadena: "soy un hacker de CS1100":

- a) Operación salto de línea
- b) Alinear a la izquierda
- c) Alinear al centro
- d) Alinear a la derecha



## "Cadenas" : Funciones

Dada la cadena: "soy un hacker de CS1100":

a) Operación salto de línea

```
1 \n
```

b) Alinear a la izquierda

```
1 '{:100}'.format(s)
```

c) Alinear al centro

```
1 '{:~100}'.format(s)
```

d) Alinear a la derecha

```
1 '{:>100}'.format(s)
```





## "Cadenas" : Funciones

### Ejemplo 4

**Elige un refrán, forma una cadena con el refrán y luego:**

**1. Imprime:**

**a) Todas las letras en mayúsculas**

**b) Todas las letras en minúsculas**

**2. Considerando un ancho de 100 lugares:**

**a) Escribe alineado a la izquierda**

**b) Escribe centrado**

**c) Escribe alineado a la derecha**

# "Cadenas" : Funciones

## Ejemplo 4 - Solución

### a) Todas las letras en mayúsculas

```
1 refran = "A quien madruga, Dios le ayuda"  
2 print (refran.upper())
```

#### Da como resultado:

```
1 A QUIEN MADRUGA, DIOS LE AYUDA
```

### b) Todas las letras en minúsculas

```
1 refran = "A quien madruga, Dios le ayuda"  
2 print (refran.lower())
```

#### Da como resultado:

```
1 a quien madruga, dios le ayuda
```

# "Cadenas" : Funciones

## Ejemplo 4 - Solución

### 2. Considerando un ancho de 100 lugares:

#### a)Escribe alineado a la izquierda

```
1 refran = "A quien madruga, Dios le ayuda"  
2 print('{:100}'.format(refran))
```

#### Da como resultado:

```
1 A quien madruga, Dios le ayuda
```

#### b)Escribe centrado

```
1 print('{:^100}'.format(refran))
```

#### Da como resultado:

```
1 A quien madruga, Dios le ayuda
```

#### c)Escribe alineado a la derecha

```
1 print('{:>100}'.format(refran))
```

#### Da como resultado:

```
1 A quien madruga, Dios le ayuda
```

# "Cadenas" : Funciones

## JOIN

Devuelve una cadena de texto, en la cual, los valores de la cadena original aparecen separados por un caracter que fue enviado a `join()` como argumento. Ejemplo 1:

```
1 s = "soy un hacker de CS1100"  
2 >>> print(" ".join(s))
```

**Da como resultado:**

```
1 s o y   u n   h a c k e r   d e   C S 1 1 0 0
```

**Ejemplo 2:**

```
1 >>> print(", ".join(s))
```

**Da como resultado:**

```
1 s,o,y, ,u,n, ,h,a,c,k,e,r, ,d,e, ,C,S,1,1,0,0
```

# "Cadenas" : Funciones

## FIND

busca una cadena dentro de otra y devuelve la posición donde la encuentra ó -1 si no la encuentra.

```
1 s = "soy un hacker de CS1100"  
2  
3 print(s.find("hack"))      #7  
4 print(s.find("1"))        #19
```

# "Cadenas" : Funciones

## REPLACE

Busca un cadena dentro de otra y reemplaza todas sus ocurrencias.

```
1 s = "soy un hacker de CS1100"  
2 print(s.replace("hacker", "developer"))  
3 print(s.replace("CS1100", "CS"))
```

## Da como resultado:

```
1 soy un developer de CS1100  
2 soy un hacker de CS
```



## Ejemplo 5

**Escriba un programa que obtenga un solo string de dos strings diferentes dados por el usuario, separados por un espacio y donde se intercambien los primeros 2 caracteres de los respectivos strings.**

### Ejemplo

**Si la entrada es:**

```
1 abc
2 xyz
```

**La salida es:**

```
1 xyc abz
```

## Ejemplo 5 -Solución

Escriba un programa que obtenga un solo string de dos strings diferentes dados por el usuario, separados por un espacio y donde se intercambien los primeros 2 caracteres de los respectivos strings.

```
1 string1 = input("Ingrese string 1 ")
2 string2 = input("Ingrese string 2 ")
3 new_a = string2[:2] + string1[2:]
4 new_b = string1[:2] + string2[2:]
5 print( new_a + ' ' + new_b)
```

## Ejercicio 1

Escriba un programa que agregue "ing" al final de cualquier string (de longitud mínimo 3 caracteres). Si el string ya termina en "ing" entonces agregar ly. Si el string tiene menos de 3 caracteres, devuelve el mismo string.

**Ejemplos:**

**Si la entrada es:**

1 Runn

**La salida es:**

1 Running

**Si la entrada es:**

1 Dying

**La salida es:**

1 Dyingly

## Ejercicio 1 - Solución

Escriba un programa que agregue "ing" al final de cualquier string (de longitud mínimo 3 caracteres). Si el string ya termina en "ing" entonces agregar ly. Si el string tiene menos de 3 caracteres, devuelve el mismo string.

```
1 palabra = input("Ingrese la palabra: ")
2 length = len(palabra)
3 if length > 2:
4     if palabra[-3:] == 'ing':
5         palabra += 'ly'
6     else:
7         palabra += 'ing'
8 print(palabra)
```

## Ejercicio 2

Escriba un programa que tome los últimos 2 caracteres de un string que ingrese el usuario y cree un nuevo string que repita 4 veces esa secuencia de 2 caracteres. Si la longitud del string original es menor que 2, devolver "no cumple".

**Ejemplos:**

**Si la entrada es:**

1 la

**La salida es:**

1 lalalala

**Si la entrada es:**

1 y

**La salida es:**

1 no cumple

## Ejercicio 2 - Solución

Escriba un programa que tome los últimos 2 caracteres de un string que ingrese el usuario y cree un nuevo string que repita 4 veces esa secuencia de 2 caracteres. Si la longitud del string original es menor que 2, devolver "no cumple".

```
1 string = input("ingrese el string: ")
2 if len(string)> 2:
3     sub_str = string[-2:]
4     print (sub_str * 4)
5 else:
6     print ("no cumple")
```



## Ejercicio 3

**Escriba un programa que permita ingresar un texto y luego muestre el mismo en el format(MayusculaMinuscula...)**

**Ejemplo:**

**Si la entrada es:**

1 Universidad

**La salida es:**

1 UnIvErSiDaD

## Ejercicio 3 - Solución

Escriba un programa que permita ingresar un texto y luego muestre el mismo en el format(MayusculaMinuscula...)

```
1 x=0
2 cadena= input ("Ingrese texto: ")
3 for letra in cadena:
4     if x % 2 ==0:
5         print(letra.upper(),end=" ")
6     else:
7         print(letra, end=" ")
8     x=x+1
```

Your Turn!!!



## "Cadenas" : Evaluación

### CADENA (STRING)

- a) ¿Con qué caracter se expresa un string en Python?
- b) ¿Con qué operadores realizamos concatenación y repetición?
- c) ¿Qué realiza cada una de las siguientes funciones: join, replace, find, upper, lower, len, split?
- f) ¿Se puede modificar un String?

## "Cadenas" : Resumen

### CADENA (STRING)

**En la sesión de hoy, aprendimos qué son STRINGS ó CADENAS DE TEXTO. Vimos:**

- a) Cómo se expresa un string en Python ( comilla simple, doble, triple)**
- b) Vimos los operadores: + (concatenación) \* (repetición)**
- c) Conocimos las funciones: join, replace, find, upper(), lower(), len, split**
- d) Aprendimos a obtener substrings: [:2 ]**
- e) Aprendimos a iterar a través de un string con FOR**
- f) Entendimos que los String son inmutables, no pueden cambiar.**

Luego de finalizar esta sesión, ya te encuentras en la capacidad de:

- Realizar programas que utilizan cadenas de caracteres.



**See you next week!**

