# CADENAS DE CARACTERES - STRINGS

Computer Science

**CS1100 - Introducción a Ciencia de la Computación**

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

1. 
2. 
3. 
4. 
5. 
6. 
7. 
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.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# CADENAS DE CARACTERES


BLOQUES (figuras)

## 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.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# ¿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"'
3 ## £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)
3  ##Ingrese nombre
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# "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"
2 ## Mi nombre es: Carlos D'Alessio
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# 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)
5
6 "hola amigo"
```

# "Cadenas": operaciones

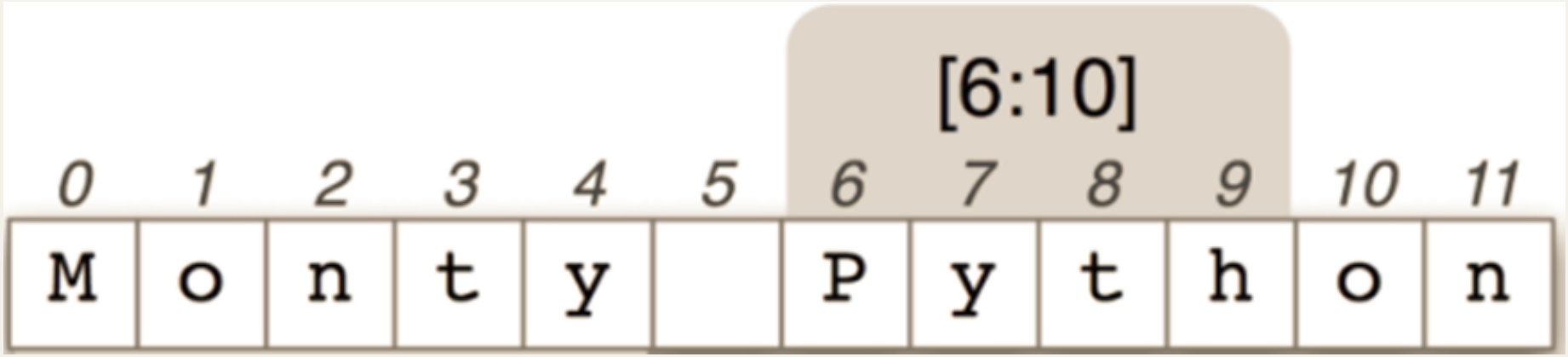
## REPETICION

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

```
1 Ejemplo 1 :
2
3 asteriscos = "*" * 20
4
5 #El resultado es:
6 ********************
```

```
1 Ejemplo 2:
2
3 asteriscos = "Hola..." * 5
4
5 #El resultado es:
```

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




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

# "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"
3
4 #El resultado es:
5 TypeError: 'str' object does not support
     item assignment
6 **********************
```

```
1 Ejemplo 2:
2 Si necesitamos un string distinto, debemos
      crear uno nuevo
3 asteriscos = "Hola..." * 5
4
```

# Ejemplo 1

Imprima todas las letras de la palabra "saludos"

ej02.py | ej01.py | ej03.py | ej0

```
1  s = "saludos"
2  for x in s:
3      print(x)
4
```

Run ej03

```
/Library/Frameworks/Python.framework/Ve
s
a
l
u
d
o
s

Process finished with exit code 0
```

# Ejemplo 2

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

ej02.py | ej01.py | ej03.py | ej04.py | ej06.py | ej08.py

```
frase="Estamos desarrollando un codigo en python"
letra="s"
k=0
for i in frase:
    if i==letra:
        k=k+1
print("la letra ",letra,"aparece ",k," veces en la frase '",frase,"' ")
```

Run ej04

```
/Library/Frameworks/Python.framework/Versions/3.5/bin/python3.5 /Users/fiestas/PycharmProj
la letra  s aparece  3  veces en la frase ' Estamos desarrollando un codigo en python '

Process finished with exit code 0
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# 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?

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# Ejemplo 3 - Solución

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

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

```
1 refran = "No por mucho madrugar se amanece
      más temprano"
2 letras = 0
3 for i in refran:
4     if i !=" ":
```

# "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
2  pi = 3.14
3  ##text = 'El valor de pi es: ' + pi
4  ## NO, no funciona
5  ## Debe ser:
6  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
2 >>> len('La vida es mucho mejor con Python
      .')
3 >>> 34
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# "Cadenas" : Funciones

## FIND

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

```
1
2  >>> "La vida es mucho mejor con Python".
       find("Python")
3  27
```

# "Cadenas" : Funciones

## FIND

Devuelve -1 si no halla el string.

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

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# "Cadenas" : Funciones

## REPLACE

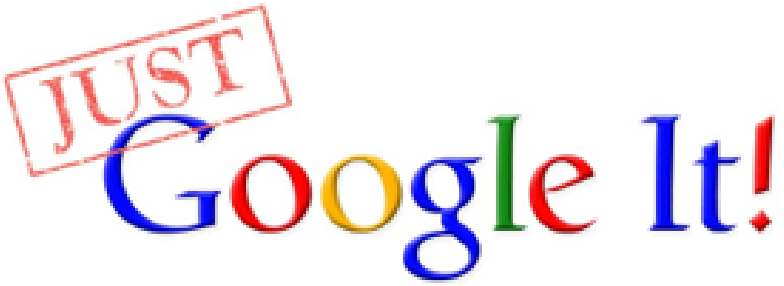
Reemplaza el string a buscar por el indicado dentro del string grande

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

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# "Cadenas" : Funciones

Dada la cadena: "soy un hacker de CS1100":
a) Capitaliza el primer carácter 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




UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# "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()
```

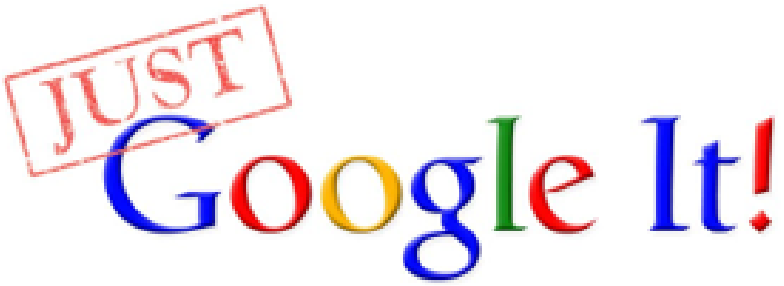
d) Elimina los espacios en blanco de los costados

```
1 strip()
```

e) Insertar tabulado

# "Cadenas" : Funciones

Dada la cadena: "soy un hacker de CS1100":
a) Operación retorno de línea
b) Alinear a la izquierda
c) Alinear al centro
d) Alinear a la derecha




UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# "Cadenas" : Funciones

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

a) Operación retorno 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 centrado
b)Escribe alineado a la izquierda
c)Escribe alineado a la derecha

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# "Cadenas" : Funciones

## Ejemplo 4 - Solución

a)Todas las letras en mayúsculas

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

b)Todas las letras en minúsculas

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

2. Considerando un ancho de 100 lugares:

a)Escribe centrado

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

b)Escribe alineado a la izquierda

# "Cadenas" : Funciones

## JOIN

devuelve una cadena de texto donde los valores de la cadena original que llama el join() aparecen separados por un caracter que fue pasado al join() como argumento:

```
1
2 s = "soy un hacker de CS1100"
3 >>> print(" ".join(s))
4 s o y   u n   h a c k e r   d e   C S 1 1
    0 0
5 >>> print(",".join(s))
6 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 posicion donde la encuentra o -1 si no está.

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

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# "Cadenas" : Funciones

## REPLACE

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

```
1 s = "soy un hacker de CS1100"
2
3 #replace
4 print(s.replace("hacker", "developer"))
5 print(s.replace("CS1100", "CS"))
6 #resultado: soy un developer de CS1100
7 #resultado: soy un hacker de CS
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# 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: input: "abc" y "xyz" output: "xyc abz"

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# Ejemplo 5 -Solución

```
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)
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# 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.
Ejemplo: input: "Runn"
output: "Running"
input: "Dying"
output: "Dyingly"

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# Ejercicio 1 - Solución

```
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)
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# 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"
Ejemplo: input: "la" output: "lalalala"
input: "y" output: "no cumple"

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# Ejercicio 2 - Solución

```
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")
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# Ejercicio 3

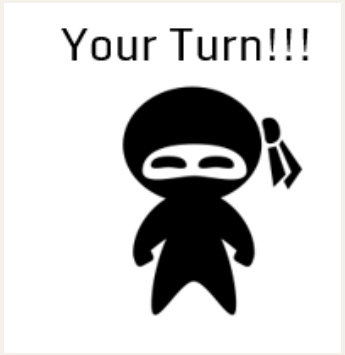
Escriba un programa que permita ingresar un texto y luego muestre el mismo en el format(MayusculaMinuscula) Ej: Universidad UnIvErSi-DaD

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# Ejercicio 3 - Solución

```
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
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA


Your Turn!!!

## de:

- Realizar programas que utilizan cadenas de caracteres.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# "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 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.
g) Vimos varios ejemplos
h) Resolvimos ejercicios en HackerRank

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# See you next week!

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA