UNIDAD 2: VARIABLES Y TIPOS DE DATOS

CONTENIDOS:

2.7 Cadena de caracteres

2.8 Operaciones con cadena de caracteres.

OBJETIVOS ESPECÍFICOS

- Usar funciones de lenguaje de programación para manipular cadenas de caracteres.
- Utilizar métodos de cadenas para manipular texto y resolver problemas.

CADENAS DE CARACTERES

- Se considera un **tipo de dato compuesto** porque están formadas de elementos más pequeños llamados **caracteres.**
- Se pueden crear utilizando comillas **simples** o **dobles**.

Tipo	Nombre	Descripción	Ejemplo
Cadenas	str	Expresiones (texto) formadas por caracteres. Se pueden representar con comillas simples o dobles.	‘Hola’ “Mundo”

CADENAS DE CARACTERES

- Se puede acceder a los caracteres de una cadena individualmente, utilizando un índice (proceso conocido como ***indexar***).
- Los índices inician en cero y terminan en la longitud de la cadena, menos uno.

ciudad = 'Manta'

indice	0	1	2	3	4
caracter	M	a	n	t	a

indice	-5	-4	-3	-2	-1
caracter	M	a	n	t	a

CADENAS DE CARACTERES

- El **operador corchete** [] selecciona solo un caracter de una cadena: **cadena[índice]**

```
>>> fruta = "banana"
>>> letra = fruta[2]
>>> print (letra)
```

– **Muestra el caracter en la posición tres (3)**

TIP:

Las posiciones inician desde **1**, los índices desde **0**.

OPERACIONES CON CADENAS

Operación	Acción
x **in** s	Indica si el texto de la cadena **x** está contenido en la cadena **s**
x **not in** s	Indica si el texto de la cadena **x** no está contenido en la cadena **s**
s **+** t	Concatena dos cadenas
s ***** n, n ***** s	Multiplica (repite) una cadena **n** veces
len(s)	Indica la longitud de la cadena
min(s)	Indica el menor caracter dentro de la cadena (de acuerdo a su código ASCII)
max(s)	Indica el mayor caracter dentro de la cadena (de acuerdo a su código ASCII)

OPERACIONES CON CADENAS

- La función **`len()`** devuelve el número de caracteres de una cadena:

```
>>> fruta = "banana"
>>> len(fruta)
6
```

- Si deseamos obtener la última letra de la cadena de caracteres:

```
longitud = len(fruta)
ultima = fruta[longitud-1]
```

```
longitud = len(fruta)
ultima = fruta[-1]
```

OPERACIONES CON CADENAS

Slicing (recorte) cadena[*start:stop:step*]

Operación	Acción
s[i]	Muestra el caracter indicado por el índice i en la cadena s
s[i:j]	Muestra caracteres en el rango entre el índice i (incluido) y el índice j (no incluido)
s[i:j:k]	Muestra caracteres en el rango entre el índice i (incluido) y el índice j (no incluido), con saltos indicados por k. Si el **step** es negativo, se recorre la cadena desde la derecha.

```
a = "abcdefghi"
a[:]  = a[0:len(a):1]= 'abcbdefghi' # a +1 step is the default
a[::2]   = a[0:len(a):2] = 'acegi'      # posiciones pares
a[1::2]  = 'bdfh'
a[::-1]  = 'ihgfedcba'
```

OPERACIONES CON CADENAS

- Dividir la cadena de caracteres s="Fundamentos"

Code	Result	Descripción
s[2:5]	nda	Caracteres en los indices 2,3,4.
s[:5]	Funda	Los primero 5 caracteres.
s[5:]	mentos	Caracteres desde el índice 5 al final.
s[-2:]	os	Los últimos dos caracteres.
s[:]	Fundamentos	Toda la cadena de caracteres.
s[1 : 7 : 2]	udm	Caracteres desde el índice 1 al 6, step 2.
s[: : -1]	sotnemadnuF	Un step negativo muestra al revese el string.

TIP:

Hay que tener cuidado con salirse del rango de una cadena (tamaño)

OPERACIONES CON CADENAS

- Una cadena de caracteres es **inmutable**, es decir sus elementos no se pueden modificar.
- Si se requieren modificaciones, se debe construir una cadena nueva (muchas veces esto lo hace automáticamente el intérprete de Python).

```
saludo = "¡Hola a todo el mundo!"
saludo[2] = 'L'              # ¡ERROR!
print (saludo)
```

```
#ERROR
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment
```

MÉTODOS DE CADENAS

Operación	Acción
str.isalnum()	Indica si la cadena (completa) es alfanumérica
str.isalpha()	Indica si la cadena (completa) es alfabética
str.isdigit()	Indica si la cadena (completa) está formada por dígitos
str.islower()	Indica si la cadena está formada por minúsculas
str.isupper()	Indica si la cadena está formada por mayúsculas
str.lower()	Muestra la cadena convertida en minúsculas
str.upper()	Muestra la cadena convertida en mayúsculas

MÉTODOS DE CADENAS

s.lower(), s.upper()

- Retorna el texto almacenado en s en minúsculas o mayúsculas respectivamente

```
 name = "Espol @ Ecuador"
length = len(name)
nameUp = name.upper()
print nameUp, "tiene", length, "caracteres"
```

Salida:

```
ESPOL @ ECUADOR tiene 15 caracteres
```

MÉTODOS DE CADENAS

Operación	Acción
str.capitalize()	
str.count()	
str.endswith()	
str.startswith()	
str.find()	
str.index()	
str.strip()	
str.replace()	
str.split()	

- Llenar la tabla con las acciones que realiza cada método

MÉTODOS DE CADENAS

Buscar caracter o subcadena utilizando la función `index()`

```
>>> cadena = “hola”
>>> cadena.index('a')
3
>>> cadena.index('ha')
Traceback (most recent call last):
  File "", line 1, in
ValueError: substring not found
```

- Utilizando la función `find()`

```
>>> cadena = "hola”
>>> cadena.find('a')
3
>>> cadena.find('ha')
-1
```

MÉTODOS DE CADENAS

Reemplazar texto

- En Python esto lo hacemos con el método `replace`
- **Función**: `replace("subcadena a buscar", "subcadena por la cual reemplazar")`
- **Retorna**: la cadena reemplazada.

```
>>> buscar = "nombre apellido"
>>> reemplazar_por = "Juan Pérez"
>>> print ("Estimado Sr. nombre apellido:".replace(buscar,
reemplazar_por))
Estimado Sr. Juan Pérez:
```

MÉTODOS DE CADENAS

- **Convertir mayúsculas a minúsculas y viceversa**
- **Método**: swapcase()
- **Retorna**: una copia de la cadena convertidas las mayúsculas en minúsculas y viceversa.

```
>>> cadena = "Hola Mundo"
>>> print (cadena.swapcase())
hOLA mUNDO
```

- **Convertir una cadena en Formato Título**
- **Método**: title()
- **Retorna**: una copia de la cadena convertida.

```
>>> cadena = "hola mundo"
>>> print (cadena.title())
Hola Mundo
```

MÉTODOS DE CADENAS

- s.split('delim')
  - retorna una lista de subcadenas separadas por el delimitador dado. Por ejemplo:
  - 'aaa,bbb,ccc'.split(',') -> ['aaa', 'bbb', 'ccc'].
  - Si se usa sin argumento el delimitador son los "whitespace characters" (enter, espacio en blanco)
- s.join(list)
  - opuesto a split(), une los elementos de una lista usando el delimitador dado.
  - '---'.join(['aaa', 'bbb', 'ccc']) -> aaa---bbb---ccc

PRÁCTICA AUTÓNOMA

- Escriba un progama que pida una cadena de caracteres. Deberá mostrar por pantalla lo siguiente:

- El número total de caracteres
- La cadena repetida 5 veces separada por un enter
- Los tres primeros caracteres de la cadena
- Los tres últimos caracteres de la cadena
- La cadena escrita al reves (Hola → aloH)
- La cadena en mayúsculas
- La cadena con cada letra “a” remplazada por una “e”