Python

Cadenas de Caracteres

Cadenas de Caracteres

```
>>>nombre = 'Juan'
>>>apellido = "Perez"
```

Pueden estar encerradas entre comillas simples ó dobles

```
>>>nombre + apellido
'JuanPerez'

>>>apellido + ',  ' + nombre
'Perez, Juan'
```

Se pueden concatenar mediante el signo +

Cadenas de Caracteres

Si las comillas deben formar parte de la cadena, entonces se deben combinar comillas simples y dobles, ó anteponer el carácter de exclusión "\"

```
>>>linea = 'Ella dijo: "No fue mi culpa"'
```

ó

```
>>>linea = "Ella dijo: \"No fue mi culpa\""
```

Carácter de Exclusión

```
>>>linea
'Ella dijo: "No fue mi culpa"'
```


Cadenas de Caracteres

Si queremos almacenar varias líneas de texto, debemos encerrarlas entre triples comillas, ya sea simples, ó dobles

```
>>>texto = """Bello es mejor que feo.
Explícito es mejor que implícito.
Simple es mejor que complejo.
Complejo es mejor que complicado."""
```

```
>>>texto
'Bello es mejor que feo. \nExplícito
es mejor que implícito. \nSimple es
mejor que complejo. \nComplejo es
mejor que complicado.
```

Carácter de Nueva Línea

```
>>>print(texto)
Bello es mejor que feo.
Explícito es mejor que implícito.
Simple es mejor que complejo.
Complejo es mejor que complicado.
```

Cadenas de Caracteres

Otra alternativa cuando queremos un texto compuesto por varias oraciones, es encerrar el texto entre paréntesis y cada cadena encerrada entre comillas. En este caso, el resultado será una concatenación de todas las cadenas.

```
>>>texto = (”Texto de la oración 1.”
            “Texto de la oración 2.”
            “Texto de la oración 3.”)

>>>texto
‘Texto de la oración 1.Texto de la oración 2.Texto de la oración 3.’

>>>print(texto)
Texto de la oración 1.Texto de la oración 2.Texto de la oración 3.
```


Cadenas de Caracteres

Se puede acceder a cada uno de los caracteres de la cadena, mediante un índice.

```
>>>palabra = ”Algoritmo”

>>>palabra[0]
‘A’
>>>palabra[2]
‘g’
>>>palabra[-1]
‘o’
>>>palabra[-3]
‘t’
```

El primer carácter ocupa la posición 0

También puedo acceder desde la derecha de la cadena, siendo la posición del último carácter, -1

Cadenas de Caracteres

Podemos acceder a "rebanadas" de una cadena

Posición Incluida

```
>>>palabra[0:4]
'Algo'
```

Posición Excluida

```
>>>palabra[:4]
'Algo'
>>>palabra[4:]
'ritmo'
```

La ausencia de posición, implica desde la posición inicial ó hasta la posición final (incluídas)

```
>>>palabra[-4:-1]
'itm'
```


Cadenas de Caracteres

```
>>>len(palabra)
9
```

La función len, nos devuelve la cantidad de caracteres que hay en la cadena

```
>>>palabra.upper()
'ALGORITMO'
```

Nos devuelve la cadena convertida a mayúsculas

```
>>>palabra.count("o")
2
```

Nos devuelve la cantidad de letras "o" que hay en la cadena

```
>>>palabra.index("A")
0
```

Nos devuelve la posición de la letra "A" en la cadena

```
>>>palabra.find("ori")
3
```

Nos devuelve la posición en la que comienza la subcadena "ori", ó -1 si no está incluída

Cadenas de Caracteres

1. Mostrar el contenido de la variable palabra, imprimiendo un carácter por línea.

```
for caracter in palabra:
    print(caracter)
```

2. Mostrar la cadena resultante de extraer las vocales de la variable palabra.

```
for caracter in palabra:
    if (caracter.capitalize() not in "AEIOU"):
        print(caracter, end="")
```