

EIN413B

Programación



UVA 4 Strings

| | | | | | | | | | |
|---|---|---|---|--|---|---|---|---|---|
| H | O | L | A | | M | U | N | D | O |
|---|---|---|---|--|---|---|---|---|---|

Objetivos de Aprendizaje



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

Contenido:

- Resolver problemas usando strings.
- Comparar strings lexicográficamente.
- Procesar strings manualmente.
- Usar ciclos for y while para iterar sobre strings.



¿Dónde estamos?



EIN413B Programación / ELI109_A Introducción a la Programación
Sede Viña del Mar
Planificación Asignatura 2025-1

| SEMANA | LUNES | MARTES | MIERCOLES | JUEVES | VIERNES | SABADO | DOMINGO |
|--------|------------|------------|------------|------------|------------|------------|------------|
| 1 | 10-03-2025 | 11-03-2025 | 12-03-2025 | 13-03-2025 | 14-03-2025 | 15-03-2025 | 16-03-2025 |
| 2 | 17-03-2025 | 18-03-2025 | 19-03-2025 | 20-03-2025 | 21-03-2025 | 22-03-2025 | 23-03-2025 |
| 3 | 24-03-2025 | 25-03-2025 | 26-03-2025 | 27-03-2025 | 28-03-2025 | 29-03-2025 | 30-03-2025 |
| 4 | 31-03-2025 | 01-04-2025 | 02-04-2025 | 03-04-2025 | 04-04-2025 | 05-04-2025 | 06-04-2025 |
| 5 | 07-04-2025 | 08-04-2025 | 09-04-2025 | 10-04-2025 | 11-04-2025 | 12-04-2025 | 13-04-2025 |
| 6 | 14-04-2025 | 15-04-2025 | 16-04-2025 | 17-04-2025 | 18-04-2025 | 19-04-2025 | 20-04-2025 |
| 7 | 21-04-2025 | 22-04-2025 | 23-04-2025 | 24-04-2025 | 25-04-2025 | 26-04-2025 | 27-04-2025 |
| 8 | 28-04-2025 | 29-04-2025 | 30-04-2025 | 01-05-2025 | 02-05-2025 | 03-05-2025 | 04-05-2025 |
| 9 | 05-05-2025 | 06-05-2025 | 07-05-2025 | 08-05-2025 | 09-05-2025 | 10-05-2025 | 11-05-2025 |
| 10 | 12-05-2025 | 13-05-2025 | 14-05-2025 | 15-05-2025 | 16-05-2025 | 17-05-2025 | 18-05-2025 |
| 11 | 19-05-2025 | 20-05-2025 | 21-05-2025 | 22-05-2025 | 23-05-2025 | 24-05-2025 | 25-05-2025 |
| 12 | 26-05-2025 | 27-05-2025 | 28-05-2025 | 29-05-2025 | 30-05-2025 | 31-05-2025 | 01-06-2025 |
| 13 | 02-06-2025 | 03-06-2025 | 04-06-2025 | 05-06-2025 | 06-06-2025 | 07-06-2025 | 08-06-2025 |
| 14 | 09-06-2025 | 10-06-2025 | 11-06-2025 | 12-06-2025 | 13-06-2025 | 14-06-2025 | 15-06-2025 |
| 15 | 16-06-2025 | 17-06-2025 | 18-06-2025 | 19-06-2025 | 20-06-2025 | 21-06-2025 | 22-06-2025 |
| 16 | 23-06-2025 | 24-06-2025 | 25-06-2025 | 26-06-2025 | 27-06-2025 | 28-06-2025 | 29-06-2025 |
| 17 | 30-06-2025 | 01-07-2025 | 02-07-2025 | 03-07-2025 | 04-07-2025 | 05-07-2025 | 06-07-2025 |
| 18 | 07-07-2025 | 08-07-2025 | 09-07-2025 | 10-07-2025 | 11-07-2025 | 12-07-2025 | 13-07-2025 |

| Semana | Tema | Control | SMOJ | Tarea |
|--------|---|---------|------|-------|
| 1 | UVA 1: Python - Programas Secuenciales | - | - | - |
| 2 | UVA 1: Python - Programas Secuenciales | C1 | UVA1 | |
| 3 | UVA 2: Condicionales | C2 | UVA2 | TA1 |
| 4 | UVA 3: Ciclos | | | |
| 5 | UVA 3: Ciclos | C3 | UVA3 | |
| 6 | UVA 4: Strings | C4 | UVA4 | |
| 7 | UVA 5: Funciones | C5 | UVA5 | TA2 |
| 8 | Preparación Certamen 1 | | | |
| 9 | Certamen 1 : Jueves 08/05 17:30 hrs. | | | |
| 10 | UVA 6: Listas | C6 | UVA6 | |
| 11 | Vacaciones Estudiantes | | | |
| 12 | UVA 6: Listas | | | TA3 |
| 13 | UVA 7: Diccionarios | C7 | UVA7 | |
| 14 | UVA 8: Procesamiento de Texto y Archivos | C8 | UVA8 | TA4 |
| 15 | Preparación Certamen 2 | | | |
| 16 | Certamen 2 : Jueves 26/06 17:30 hrs. | | | |
| 17 | No hay más clases ni ayudantías | | | |
| 18 | Certamen Recuperativo : martes 08/07 17:30 hrs. | | | |

¿Qué es un String?



- Un string en Python es una secuencia **inmutable** de caracteres utilizada para almacenar y manipular texto.
- Los strings en Python se pueden definir utilizando comillas simples ('...') o dobles ("...").
- Cada carácter en un string está indexado, comenzando desde 0 para el primer carácter, y pueden ser accedidos utilizando índices positivos o negativos.
- Los strings son objetos de la clase str y admiten una variedad de operaciones básicas como concatenación, repetición, y verificación de pertenencia.
- Debido a su inmutabilidad, cualquier operación que modifique un string realmente crea un nuevo string.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| H | O | L | A | | M | U | N | D | O |

- **Concatenación (+):** `'Hello' + 'World' → 'HelloWorld'`
- **Repetición (*):** `'A' * 3 → 'AAA'`
- **Pertenencia (in, not in):** `'a' in 'apple' → True`
- **Longitud (len()):** `len('Python') → 6`

- **Índices Positivos:** $s[0]$ para el primer carácter.
- **Índices Negativos:** $s[-1]$ para el último carácter.
- **Substrings:** $s[1:4]$ → Subcadena desde el índice 1 hasta el 3.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| H | O | L | A | | M | U | N | D | O |

Inmutabilidad de Strings



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

Cuando decimos que un string en Python es inmutable, significa que una vez que un string ha sido creado, no puede ser modificado.

Por ejemplo, si tienes un string 'hello' y quieres cambiar la primera letra a mayúscula, no puedes hacer algo como esto:

```
1 s = "hello"  
2 s[0] = "H" # Error  
3
```

Intentar modificar el string de esta manera resultará en un error, porque Python no permite alterar los valores en las posiciones individuales de un string ya existente

Inmutabilidad de Strings



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

En lugar de modificar un string existente, lo que puedes hacer es crear un nuevo string a partir de las modificaciones que desees:

```
1 s = 'hello'
2 s = 'H' + s[1:] # Esto crea un nuevo string 'Hello'
3
```

Aquí, `s[1:]` obtiene una subcadena de 'ello', y luego se concatena con 'H' para formar un nuevo string 'Hello'. Así, aunque parece que hemos "modificado" el string, en realidad hemos creado uno nuevo.



La inmutabilidad es una característica importante porque garantiza que los strings son seguros para su uso en estructuras de datos complejas, ya que no pueden ser modificados por error.

Métodos upper() y lower()



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

`upper()`: Convierte todos los caracteres de un string a mayúsculas.

```
1 s = 'hello'
2 print(s.upper()) # 'HELLO'
3
```

`lower()`: Convierte todos los caracteres de un string a minúsculas.

```
1 s = 'HELLO'
2 print(s.lower()) # 'hello'
3
```

Iteración con for y while



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

En programación, iterar se refiere a usar un bucle, como for o while, para pasar por cada elemento de una secuencia, como un string, una lista o una tupla.

Por ejemplo, cuando iteras sobre un string en Python, estás procesando cada carácter del string uno por uno.

```
1 i = 0
2 while i < len('hello'):
3     print('hello'[i])
4     i += 1
5
```

```
1 for char in 'hello':
2     print(char)
3
```

Iteración con for y while



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

En programación, iterar se refiere a usar un bucle, como for o while, para pasar por cada elemento de una secuencia, como un string, una lista o una tupla.

Por ejemplo, cuando iteras sobre un string en Python, estás procesando cada carácter del string uno por uno.

```
1 i = 0
2 while i < len('hello'):
3     print('hello'[i])
4     i += 1
5
```

```
1 for char in 'hello':
2     print(char)
3
```

Comparaciones Lexicográficas



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

Orden basado en el código ASCII.

Ejemplos:

'apple' < 'banana' → True

'a' < 'B' → False (debido a que 'a' > 'B' en ASCII).

| | | | | | | |
|----|-----|-----|-----|-----|----|----|
| 65 | 112 | 112 | 108 | 101 | 58 | 41 |
| A | p | p | l | e | : |) |

For the machine everything is number.

| Dec | Hex | Oct | Chr | Dec | Hex | Oct | HTML | Chr | Dec | Hex | Oct | HTML | Chr | Dec | Hex | Oct | HTML | Chr |
|-----|-----|-----|---------------------|-----|-----|-----|--------|-------|-----|-----|-----|--------|-----|-----|-----|-----|--------|-----|
| 0 | 0 | 000 | NULL | 32 | 20 | 040 | | Space | 64 | 40 | 100 | @ | @ | 96 | 60 | 140 | ` | ` |
| 1 | 1 | 001 | Start of Header | 33 | 21 | 041 | ! | ! | 65 | 41 | 101 | A | A | 97 | 61 | 141 | a | a |
| 2 | 2 | 002 | Start of Text | 34 | 22 | 042 | " | " | 66 | 42 | 102 | B | B | 98 | 62 | 142 | b | b |
| 3 | 3 | 003 | End of Text | 35 | 23 | 043 | # | # | 67 | 43 | 103 | C | C | 99 | 63 | 143 | c | c |
| 4 | 4 | 004 | End of Transmission | 36 | 24 | 044 | $ | \$ | 68 | 44 | 104 | D | D | 100 | 64 | 144 | d | d |
| 5 | 5 | 005 | Enquiry | 37 | 25 | 045 | % | % | 69 | 45 | 105 | E | E | 101 | 65 | 145 | e | e |
| 6 | 6 | 006 | Acknowledgment | 38 | 26 | 046 | & | & | 70 | 46 | 106 | F | F | 102 | 66 | 146 | f | f |
| 7 | 7 | 007 | Bell | 39 | 27 | 047 | ' | ' | 71 | 47 | 107 | G | G | 103 | 67 | 147 | g | g |
| 8 | 8 | 010 | Backspace | 40 | 28 | 050 | (| (| 72 | 48 | 110 | H | H | 104 | 68 | 150 | h | h |
| 9 | 9 | 011 | Horizontal Tab | 41 | 29 | 051 |) |) | 73 | 49 | 111 | I | I | 105 | 69 | 151 | i | i |
| 10 | A | 012 | Line feed | 42 | 2A | 052 | * | * | 74 | 4A | 112 | J | J | 106 | 6A | 152 | j | j |
| 11 | B | 013 | Vertical Tab | 43 | 2B | 053 | + | + | 75 | 4B | 113 | K | K | 107 | 6B | 153 | k | k |
| 12 | C | 014 | Form feed | 44 | 2C | 054 | , | , | 76 | 4C | 114 | L | L | 108 | 6C | 154 | l | l |
| 13 | D | 015 | Carriage return | 45 | 2D | 055 | - | - | 77 | 4D | 115 | M | M | 109 | 6D | 155 | m | m |
| 14 | E | 016 | Shift Out | 46 | 2E | 056 | . | . | 78 | 4E | 116 | N | N | 110 | 6E | 156 | n | n |
| 15 | F | 017 | Shift In | 47 | 2F | 057 | / | / | 79 | 4F | 117 | O | O | 111 | 6F | 157 | o | o |
| 16 | 10 | 020 | Data Link Escape | 48 | 30 | 060 | 0 | 0 | 80 | 50 | 120 | P | P | 112 | 70 | 160 | p | p |
| 17 | 11 | 021 | Device Control 1 | 49 | 31 | 061 | 1 | 1 | 81 | 51 | 121 | Q | Q | 113 | 71 | 161 | q | q |
| 18 | 12 | 022 | Device Control 2 | 50 | 32 | 062 | 2 | 2 | 82 | 52 | 122 | R | R | 114 | 72 | 162 | r | r |
| 19 | 13 | 023 | Device Control 3 | 51 | 33 | 063 | 3 | 3 | 83 | 53 | 123 | S | S | 115 | 73 | 163 | s | s |
| 20 | 14 | 024 | Device Control 4 | 52 | 34 | 064 | 4 | 4 | 84 | 54 | 124 | T | T | 116 | 74 | 164 | t | t |
| 21 | 15 | 025 | Negative Ack. | 53 | 35 | 065 | 5 | 5 | 85 | 55 | 125 | U | U | 117 | 75 | 165 | u | u |
| 22 | 16 | 026 | Synchronous idle | 54 | 36 | 066 | 6 | 6 | 86 | 56 | 126 | V | V | 118 | 76 | 166 | v | v |
| 23 | 17 | 027 | End of Trans. Block | 55 | 37 | 067 | 7 | 7 | 87 | 57 | 127 | W | W | 119 | 77 | 167 | w | w |
| 24 | 18 | 030 | Cancel | 56 | 38 | 070 | 8 | 8 | 88 | 58 | 130 | X | X | 120 | 78 | 170 | x | x |
| 25 | 19 | 031 | End of Medium | 57 | 39 | 071 | 9 | 9 | 89 | 59 | 131 | Y | Y | 121 | 79 | 171 | y | y |
| 26 | 1A | 032 | Substitute | 58 | 3A | 072 | : | : | 90 | 5A | 132 | Z | Z | 122 | 7A | 172 | z | z |
| 27 | 1B | 033 | Escape | 59 | 3B | 073 | ; | ; | 91 | 5B | 133 | [| [| 123 | 7B | 173 | { | { |
| 28 | 1C | 034 | File Separator | 60 | 3C | 074 | < | < | 92 | 5C | 134 | \ | \ | 124 | 7C | 174 | | | |
| 29 | 1D | 035 | Group Separator | 61 | 3D | 075 | = | = | 93 | 5D | 135 |] |] | 125 | 7D | 175 | } | } |
| 30 | 1E | 036 | Record Separator | 62 | 3E | 076 | > | > | 94 | 5E | 136 | ^ | ^ | 126 | 7E | 176 | ~ | ~ |
| 31 | 1F | 037 | Unit Separator | 63 | 3F | 077 | ? | ? | 95 | 5F | 137 | _ | _ | 127 | 7F | 177 | | Del |

asciicharstable.com

Ejercicio de Aplicación



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

Problema: Dado un string, conviértelo todo a minúsculas y cuenta cuántas veces aparece la letra 'a'.

```
1 s = "Abracadabra"
2 s = s.lower()
3 count = 0
4 for char in s:
5     if char == 'a':
6         count += 1
7 print(count)
8
9
```


UVA 4

Strings



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

- Resolver problemas usando strings.
- Comparar strings lexicográficamente.
- Procesar strings manualmente.
- Usar ciclos for y while para iterar sobre strings.

