

Paradigmas de Programación

Sesión 3 Laboratorio



Cristian Tejedor García
Departamento de Informática
Universidad de Valladolid

Curso 2025-26

Grado en Ingeniería Informática
Grado en Estadística
INDAT





Contenido

1. Objetivos
2. Cadenas de caracteres en Python
3. Reto (ayuda)



1. Objetivos

- Resolver dudas de la teoría y ejercicios de la sesión 2.
- Explicar las funciones de cadenas de caracteres en Python.
- Realizar ejercicios de repaso.





2. Cadenas de caracteres (I)

1. Longitud

```
print(len("cadena")) # 6
```

2. Comparación (cuidado con mayúsculas y minúsculas)

```
print("cadena" == "cadena") # True
```

```
print("cadena" != "Cadena") # True
```

```
print("CADENA".lower() == "cadena".upper()) # False
```

3. División de elementos

```
a = "AMPLI;451;PARAD;729"
```

```
b = a.split(';') # Delimitador, puede ser cualquier elemento
```

```
print(b) # ['AMPLI', '451', 'PARAD', '729'])
```



2. Cadenas de caracteres (II)

4. Búsqueda

find() devuelve -1 si no encuentra, index() devuelve una excepción

```
print("cadena".find('e')) # 3
print("cadena".find('una')) # -1
print('i' in "cadena") # False
```

5. Acceso

```
print("cadena"[0]) # c
print("cadena"[-1]) # a
print("cadena"[-2]) # n
```

6. Concatenación

```
print("cadena" + "s largas") # cadenas largas
```



2. Cadenas de caracteres (III)

7. Slicing (subcadenas)

```
print("cadena"[0:]) # cadena  
print("cadena"[:-2]) # cade  
print("cadena"[2:-2]) # de  
# Empieza:Acaba-1:Número_saltos  
print("cadena muy larga"[2:-2:3]) # daul
```

8. Reemplazamiento

```
# Devuelve una copia, porque String es inmutable  
print("cadena larga".replace("a", "W", 3)) # cWdenW lWrga
```

9. Conteo

```
print("cabeza".count('a')) # 2
```



2. Cadenas de caracteres (IV)

10. Última ocurrencia de un carácter (fórmula) - *lastIndexOf()* en Java

cadena = "Paradigmas es un grupo ejemplar"

```
print(len(cadena)) # 31
```

```
print(len(cadena) - cadena[::-1].find('e') - 1) # 25, cadena[::-1] invierte cadena
```

11. Eliminación de todos los espacios al inicio y fin

```
print('      Hello      '.strip()) # Hello
```

12. Comienzo/fin por

```
print("Casa del campo".startswith("Casa")) # True
```

```
print("Casa del campo".endswith("ampa")) # False
```



2. Cadenas de caracteres (V)

13. Unión de elementos

```
a = ['this', 'is', 'a', 'string']
print("-".join(a)) # this-is-a-string
```

14. Repetición de elementos

```
a = '^'
print(a * 45) # ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```



2. Cadenas de caracteres (VI)

15. Expresiones regulares (subcadenas con cierta forma)

<https://platzi.com/blog/expresiones-regulares-python/>

```
import re

# Coincide con una letra 'a', seguida de al menos (+) 1 dígito entre 2 y 6 [2-6]
patron = re.compile('a[2-6]+')

cadena = 'ba3425 a6' # La coincidencia del patrón no está al principio!

# findall() devuelve todas las ocurrencias en forma de lista
print(patron.findall(cadena)) # ['a3425', 'a6']

# search() busca en la cadena alguna ocurrencia del patrón
print(patron.search(cadena)) # <_sre.SRE_Match object at 0x0311C3D8>

# match() igual que search() pero coincidencia sólo al comienzo de la cadena
print(patron.match(cadena)) # None
```



3. Reto (ayuda)

Disponemos de un **fichero** de texto que contiene la información de la matrícula de alumnos de primer curso. **Cada línea** del fichero representa un par **asignatura-alumno** (la asignatura se representa por una abreviatura de 2, 3 o 4 letras, le sigue un espacio y el NIA del alumno): [Enlace de descarga](#)

Crear una **función reto**, que lea el fichero y lo procese de forma que devuelva un diccionario cuyo índice sea una tupla de dos asignaturas y que almacene el número de alumnos que están matriculados simultáneamente de ambas asignaturas.

Aproximadamente: 15 líneas en Python ☺ 100 líneas en Java ☹

Ejemplo de uso: `dic = reto("reto.txt")`

```
dic[("PAR","AMAT")] → 113 # Devuelve el número de alumnos matriculados  
# simultáneamente en Paradigmas y Ampliación de Matemáticas
```

```
# Paso 1: Lista de pares ("ASIG",'NIA') del fichero  
# Paso 2: Diccionario con entradas: "ASIG" : Conjunto de NIAs  
# Paso 3: Conjunto de asignaturas (sin repetir)  
# Paso 4: Diccionario con entradas: ("ASIG1","ASIG2") : Longitud intersección conj.
```



Para afianzar...

- Ejercicios de la asignatura (sesión 2) ~2horas:
 - <https://www.infor.uva.es/~cvaca/asigs/docpar/sesion2.pdf>
- Ejercicios de la asignatura (sesión 6) ~2.5horas:
 - <https://www.infor.uva.es/~cvaca/asigs/docpar/sesion6.pdf>
- Reto propuesto sesión 2.
- “Python para todos”, Raúl González Duque
 - http://www.utic.edu.py/citil/images/Manuales/Python_para_todos.pdf
 - **Capítulos:** Revisitando objetos y repaso de todos los anteriores.