
UVA 5

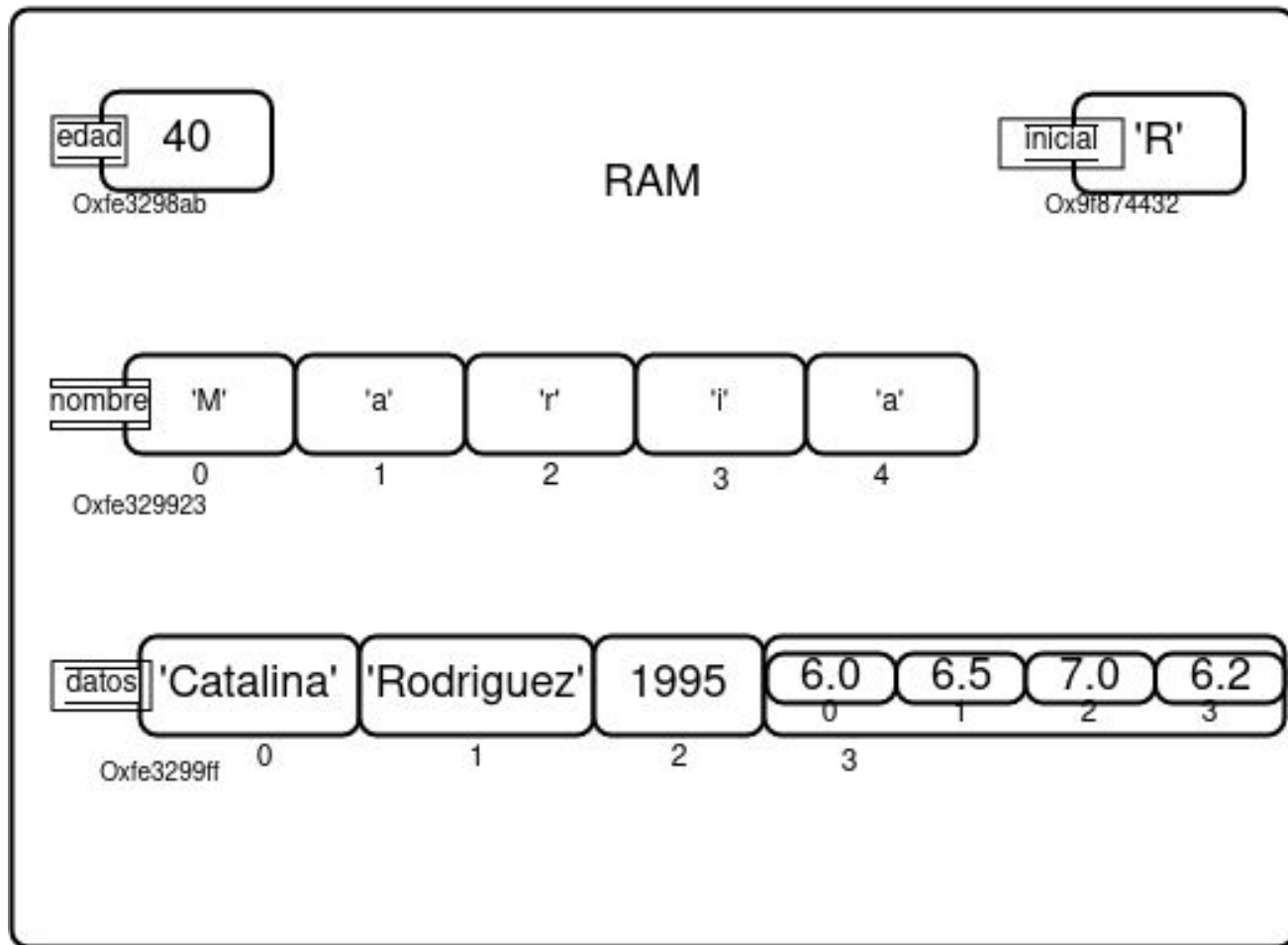
Strings y ciclo for

Roberto Asín

Sesión en Aula

Roberto Asín

Strings



Métodos, operaciones y funciones sobre strings

- $S1 + S2$ -> crea un nuevo string que es la concatenación de $S1$ y $S2$
- sub in S - > verifica si sub es un substring de S
- $num * S$ -> crea un nuevo string que es la concatenación de S consigo mismo, num veces.
- $S[pos]$ -> crea un nuevo string que incluye únicamente el caracter de la posición pos
- $S[i:f]$ -> crea un nuevo string con la subcadena de S que inicia en i y termina en f (sin incluir la posición f)
- $len(S)$ -> devuelve el tamaño de S
- $S.find(sub)$ -> devuelve la posición de sub dentro de S o -1 (si no es substring)
- $S.lower()$, $S.upper()$ -> crea un nuevo string con solo minúsculas/mayúsculas
- $S.replace(from, to)$ -> crea un nuevo string en el que las ocurrencias de $from$, son reemplazadas por ocurrencias de to

Ciclo for

1. Leer la entrada
2. Bloque 1
3. **Para cada** valor **v** en **Colección**:
 - 3.1 Bloque 2
4. Imprimir resultado



```
entrada()  
operaciones1()  
for v in Coleccion:  
    operaciones2()  
impresion_resultado()
```

Ejemplos

```
▶ for letra in 'Una furtiva lagrima':  
    print('iteracion '+letra)
```

```
↳ iteracion U  
    iteracion n  
    iteracion a  
    iteracion  
    iteracion f  
    iteracion u  
    iteracion r  
    iteracion t  
    iteracion i  
    iteracion v  
    iteracion a  
    iteracion  
    iteracion l  
    iteracion a  
    iteracion g  
    iteracion r  
    iteracion i  
    iteracion m  
    iteracion a
```

Break y continue

```
▶ for letra in 'Una furtiva lagrima':  
    if (letra=='v'):  
        break  
    print('Iteracion '+letra)  
  
print('FIN')
```

```
↳ Iteracion U  
Iteracion n  
Iteracion a  
Iteracion  
Iteracion f  
Iteracion u  
Iteracion r  
Iteracion t  
Iteracion i  
FIN
```

```
▶ for letra in 'Una furtiva lagrima':  
    if (letra=='v'):  
        continue  
    print('Iteracion '+letra)  
  
print('FIN')
```

```
↳ Iteracion U  
Iteracion n  
Iteracion a  
Iteracion  
Iteracion f  
Iteracion u  
Iteracion r  
Iteracion t  
Iteracion i  
Iteracion a  
Iteracion  
Iteracion l  
Iteracion a  
Iteracion g  
Iteracion r  
Iteracion i  
Iteracion m  
Iteracion a  
FIN
```

Reflexionemos:

- Considere la instrucción:
frase = 'Solo vuela el que se atreve a hacerlo'
 - ¿Cuál es la longitud de este string?
 - ¿Cuáles son los índices válidos?
 - ¿Cómo se expresa siempre el mayor índice en términos de la longitud?
 - ¿Cómo completaría el siguiente while para recorrer todos los índices del string S?

```
i = 0
while i < ???:
    i += 1
```
- Conversión de tipos (casting a/desde strings)
 - ¿Podemos convertir el string "23515" a un int? ¿cómo?
 - ¿Podemos hacer lo mismo con "veintitrés mil quinientos quince"?
- ¿Cuál es el resultado de estas comparaciones?
 - 'perro' < 'gato'
 - 'perro' == 'Perro'
 - 'perro' < 'Perro'
- ¿Cómo incluyo los caracteres " o ' en un string?

¿Qué hace el siguiente programa? Analice línea a línea.

```
nombre = 'Juan Carlos Bodoque'  
nombre.upper()  
print(nombre)  
print(nombre[3])  
nombre[3] = 'A'  
print('e' in 'Perro')
```

```
texto = 'gato grande, negro y gordo'  
print(texto[4])  
print(len(texto[5:8]))  
print(texto[:4])  
print(texto[-5:])
```

¿Qué hace el siguiente programa?

Haga el seguimiento con el string 'che gelida manina'.

```
texto = input('Texto: ')
inicio = True
convertido = ''
for c in texto:
    if inicio:
        convertido = convertido+c.upper()
        inicio = False
    elif c==' ':
        inicio = True
    else:
        convertido = convertido+c
print(convertido)
```

Sesión en Lab

Roberto Asín

Primitivas útiles

- ¿Cómo implementaría la búsqueda de un caracter en un string?
- ¿Y la búsqueda de un string en otro?
- Escriba un programa que encuentre la palabra de mayor longitud dentro de un texto cuyas palabras se separan por un único espacio. Considere que no hay espacio al final.