

Tipo de datos cadenas de caracteres

- Las cadenas de caracteres (`str`): Me permiten guardar secuencias de caracteres. Es un tipo inmutable. Como hemos comentado las cadenas de caracteres en python3 están codificada con Unicode.

Definición de cadenas. Constructor str

Podemos definir una cadena de caracteres de distintas formas:

```
>>> cad1 = "Hola"
>>> cad2 = '¿Qué tal?'
>>> cad3 = '''Hola,
que tal?'''
```

También podemos crear cadenas con el constructor `str` a partir de otros tipos de datos.

```
>>> cad1=str(1)
>>> cad2=str(2.45)
>>> cad3=str([1,2,3])
```

Operaciones básicas con cadenas de caracteres

Como veíamos en el apartado "Tipo de datos secuencia" podemos realizar las siguientes operaciones:

- Las cadenas se pueden recorrer.
- Operadores de pertenencia: `in` y `not in`.
- Concatenación: `+`
- Repetición: `*`
- Indexación
- Slice

Entre las funciones definidas podemos usar: `len`, `max`, `min`, `sorted`.

Las cadenas son inmutables

```
>>> cad = "Hola que tal?"
>>> cad[4]="."
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment
```

Comparación de cadenas

Las cadenas se comparan carácter a carácter, en el momento en que dos caracteres no son iguales se compara alfabéticamente (es decir, se convierte a código unicode y se comparan).

Ejemplos

```
>>> "a">"A"
True
>>> ord("a")
97
>>> ord("A")
65

>>> "informatica">"informacion"
True

>>> "abcde">"abcdef"
False
```

Funciones repr, ascii, bin

- `repr(objeto)` : Devuelve una cadena de caracteres que representa la información de un objeto.

```
>>> repr(range(10))
'range(0, 10)'
>>> repr("piña")
"'piña'"
```

La cadena devuelta por `repr()` debería ser aquella que, pasada a `eval()`, devuelve el mismo objeto.

```
>>> type(eval(repr(range(10))))
<class 'range'>
```

- `ascii(objeto)` : Devuelve también la representación en cadena de un objeto pero en este caso muestra los caracteres con un código de escape . Por ejemplo en `ascii` (Latin1) la `á` se presenta con `\xe1` .

```
>>> ascii("á")
"'\\xe1'"
>>> ascii("piña")
"'pi\\xf1a'"
```

- `bin(numero)` : Devuelve una cadena de caracteres que corresponde a la representación binaria del número recibido.

```
>>> bin(213)
'0b11010101'
```