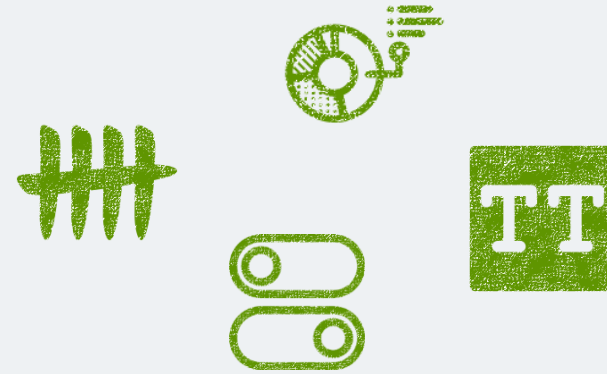


# NIVEL 2

---

## CADENAS DE CARACTERES O STRINGS





¿Qué tipos de datos  
hemos visto?

# RECORDAMOS...

# ASPECTOS DE CODIFICACIÓN



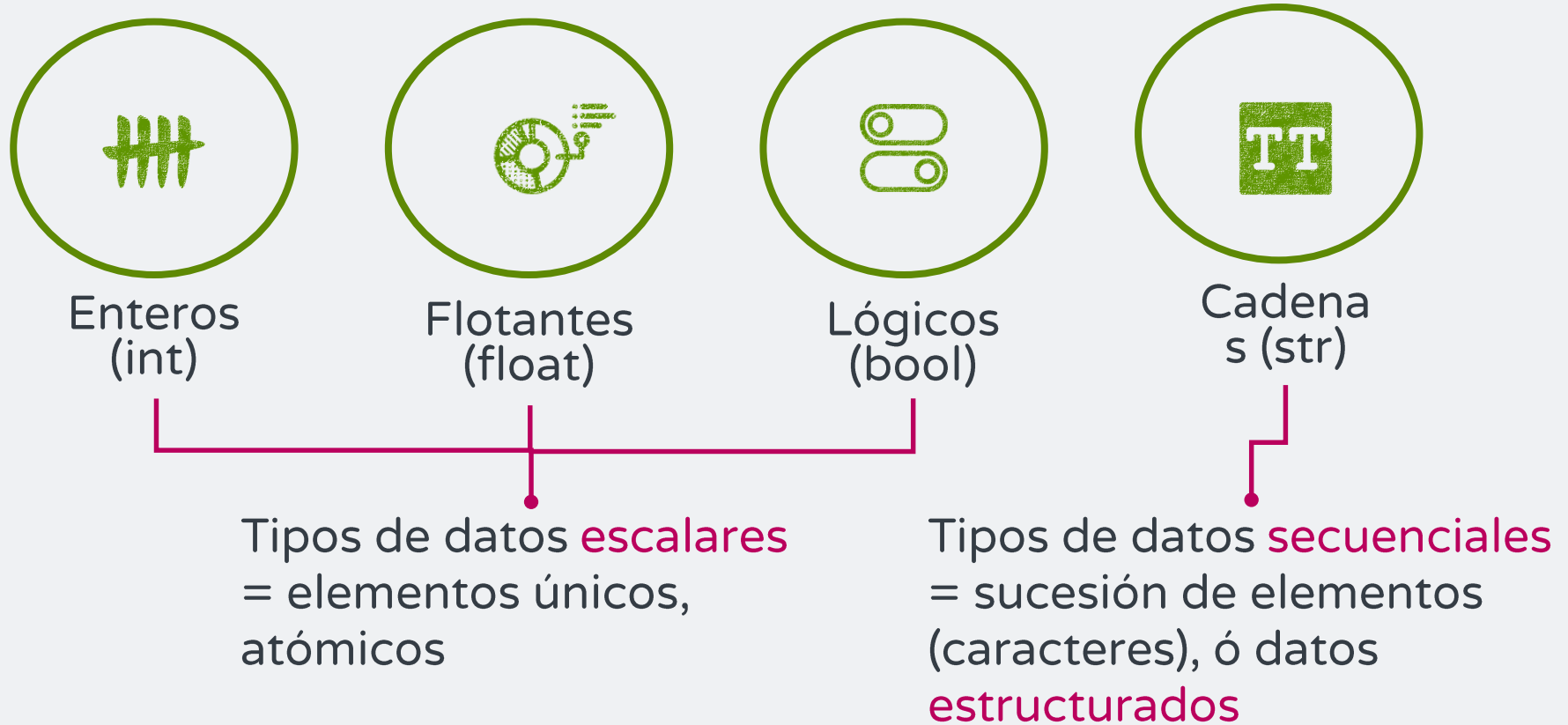
- ✓ ASCII (127 caracteres)
- ✓ ISO-8859-1 (incluye caracteres de uso frecuente en lenguajes originados en Europa occidental. Por ejemplo: á, Á, ñ, Ñ, ç, ü, ò)
- ✓ UNICODE (muchos más caracteres incluidos los emoticones)

```
# -*- coding: utf-8 -*-
```

```
Python 3.7.6 (default, Jan 8 2020, 13:42:34)  
Type "copyright", "credits" or "license" for more information.  
  
IPython 7.12.0 -- An enhanced Interactive Python.  
In [1]:  
  
# -*- coding: utf-8 -*-
```

The screenshot shows a terminal window titled 'Console 1/A'. It displays the Python 3.7.6 startup message, followed by the IPython 7.12.0 prompt. The user has entered 'In [1]:' and the encoding declaration from the previous block is shown below it. The status bar at the bottom indicates 'Python 3.7.6', 'Line 15, Col 1', 'UTF-8', 'LF', 'RW', and 'Mem 65%'.

# TIPOS DE DATOS



# PARTICULARIDADES DE LA CODIFICACIÓN DE LOS CARACTERES EN LAS CADENAS

---



Caracteres de control



Longitud de la cadena

# CARACTERES DE CONTROL

- Son caracteres especiales que no tienen una representación trivial
- Los más utilizados son:
  - `\n`: cambio de línea
  - `\t`: tabulador
  - `\\`: barra invertida
  - `\'`: comilla simple
  - `\"`: comilla doble
  - `\ ↵ -y salto de línea-`: se ignora (se usa para expresar una cadena en varias líneas)



# EJEMPLOS

```

EjemploCaracteresDeControl.py
1 a = "una\ncadena"
2 print(a,"\n")
3
4 b = "uno\tdos\ttres"
5 print(b,"\n")
6
7 c = "1\t2\t3"
8 print(c,"\n")
9
10 print("aa\tbb\tcc\nxx\tyy\tzz\n")
11
12 print("Una \"cosa\" rara.\n")
13
14 print("a\\b\n")
15
16 print("a\
17 b\n")
18
19 print("a\
20     b\n")
21
22 print("Una\ncadena que\nocupa varias líneas\n")
23
24 print("""Una
25     cadena que
26     ocupa varias líneas\n""")

```

```

Terminal 1/A
In [24]: runfile('C:/Users/Mhe
Mhernandez/Desktop/IP/N2-C3')
una
cadena

uno      dos      tres

1        2        3

aa       bb       cc
xx       yy       zz

Una "cosa" rara.

a\b

ab

a        b

Una
cadena que
ocupa varias líneas

Una
        cadena que
        ocupa varias líneas

```

# LONGITUD DE LA CADENA

La primera nueva función que estudiaremos es **len** (abreviatura del inglés «length», en español, «longitud»)

- Devuelve la longitud de una cadena, es decir, el número de caracteres que la forman. Se trata de una función predefinida, así que podemos usarla directamente

```
Terminal 1/A ✕  
  
In [27]: len("abc")  
Out[27]: 3  
  
In [28]: len("a")  
Out[28]: 1  
  
In [29]: len("abcd" * 4)  
Out[29]: 16  
  
In [30]: len("a\nb")  
Out[30]: 3  
  
In [31]: len("")  
Out[31]: 0  
  
In [32]: len(" ")  
...:  
Out[32]: 1  
  
In [33]: len("  ")  
Out[33]: 3
```

La cadena vacía (sin caracteres) es diferente de la cadena con un espacio o más en blanco