

Curso introductorio de Python

Julio 2020, Online



Written by

Impartido por Luis Ángel
Alcántara



Construyendo un mejor
entorno de negocios

“

When people make predictions about the future, it's a lot more informative about their own areas of interest, biases and knowledge gaps, than about the future...

François Chollet

Creator of Keras, neural networks library



Contenidos

1. Introducción a Python y ML
2. Tipos de datos
3. Estructuras algorítmicas y expresiones regulares
4. Funciones
5. Arreglos y matrices
6. Series y Dataframes
7. Visualización
8. Conceptos fundamentales de ML
9. Aprendizaje supervisado
10. Aprendizaje no supervisado

1. Introducción a Python y Machine Learning

¿Qué es y qué no es Machine Learning? ¿Qué es Python?
Inicialización del ambiente de trabajo.



“I’m not a great programmer: I’m just a good programmer with great habits”

Ken Beck

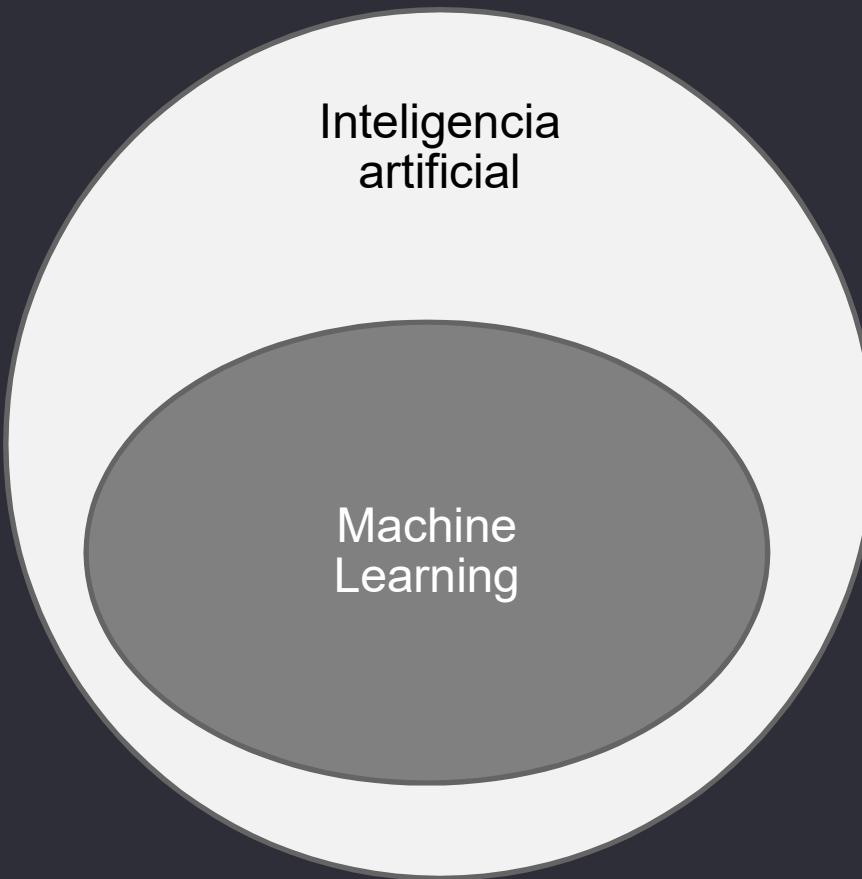


The better the question. The better the answer.
The better the world works.



Construyendo un mejor
entorno de negocios

¿Qué es y qué no es Machine Learning?



Inteligencia artificial

“Habilidad de una computadora digital o un robot controlado por computadora capaz de realizar acciones asociadas a seres inteligentes.”

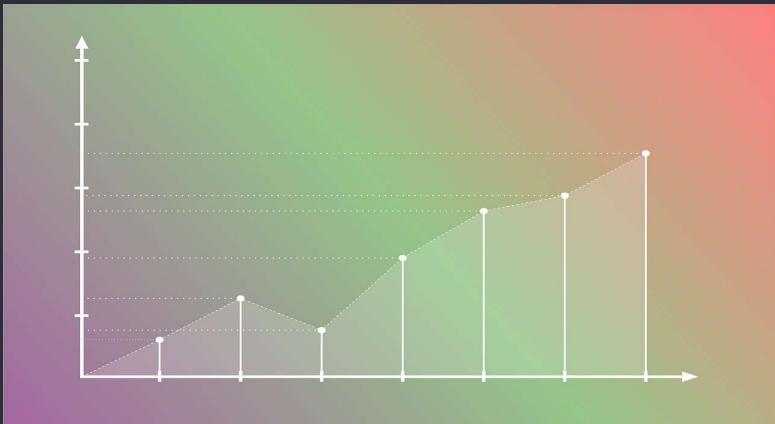
Encyclopedia britannica.

Machine learning

“En inteligencia artificial, es una disciplina concerniente a la implementación de software computacional que aprende autónomamente.”

Encyclopedia britannica.

¿Qué es y qué no es Machine Learning?



Reconocimiento de patrones

“El campo del reconocimiento de patrones concierne al descubrimiento automático de regularidades en datos mediante el uso de algoritmos computacionales”

Christopher Bishop.

Aprendizaje estadístico

*“En general, el aprendizaje estadístico involucra construir modelos estadísticos para predecir, o estimar, una **salida** basada en una o más **entradas**.”*

Gareth James.



¿Qué es Python?

Es un lenguaje de programación de alto nivel creado por **Guido Van Rossum**. Entre sus características encontramos:

- De código abierto.
- Enfatiza la legibilidad y limpieza de código.
- Multiparadigma (POO, funcional, imperativa) y multiplataforma.
- Permite hacer prototipos funcionales muy rápidamente.

¿Qué más se puede hacer con Python?

- Web development.
- Computer vision.
- Data Science.
- Machine learning.
- Web scraping.
- Robotics.

“

Over six years ago, in December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office (a government-run research lab in Amsterdam) would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus).

Guido Van Rossum



Ventajas y desventajas de Python

Ventajas

- ▶ Lenguaje de alto nivel, fácilmente integrable con código en C, C++
- ▶ Extensa colección de librerías.
- ▶ Fácil de aprender a programar, con estructuras de datos potentes.
- ▶ Multiplataforma.
- ▶ Pasar de prototipos a producción es más sencillo que a otros lenguajes.
- ▶ Cuenta con un gran soporte por parte de la comunidad de desarrolladores.
(Stackoverflow, etc)
- ▶ Gratuito.
- ▶ Gigantes tecnológicos que usan Python:
Dropbox, Google, Spotify, Netflix,
Instagram, Facebook, Paypal, Uber, etc.

Desventajas

- ▶ Errores en su tiempo de ejecución, ya que es un lenguaje interpretado.
- ▶ Ejecución lenta comparado con otros lenguajes interpretados.
- ▶ Al ser interpretado consume más memoria.
- ▶ Aún no es un lenguaje ideal para aplicaciones multiproceso.

Ambiente de trabajo

La distribución Anaconda es utilizada por científicos de datos, ya que facilita la administración de entornos de *Python* y *R*.

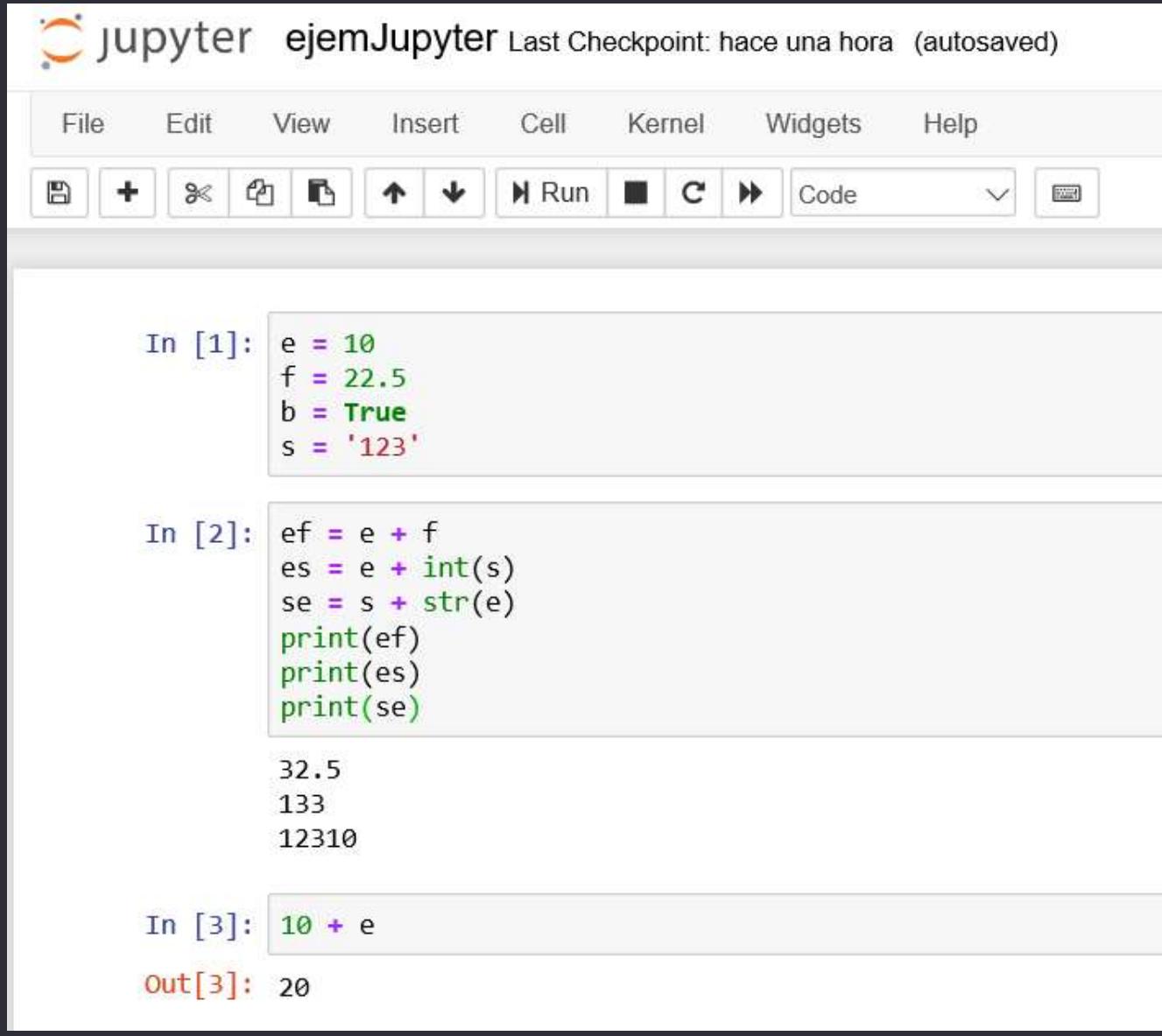
Anaconda tiene una gran colección de herramientas que permiten procesar grandes volúmenes de información, análisis predictivo , manejo de arreglos, aprendizaje de máquina y visualización.

Otra herramienta es Google Colab que funciona de manera remota prestando recursos de forma virtual, al igual que Anaconda tiene una gran cantidad de herramientas por default. No hay necesidad de instalar nada en la máquina de forma local.



1 Eligiendo ambiente de trabajo

¿Con qué vamos a trabajar?



The screenshot shows a Jupyter Notebook interface with the title "jupyter ejemJupyter Last Checkpoint: hace una hora (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with icons for file operations like save, new, and run, along with a "Code" dropdown.

In [1]:

```
e = 10
f = 22.5
b = True
s = '123'
```

In [2]:

```
ef = e + f
es = e + int(s)
se = s + str(e)
print(ef)
print(es)
print(se)
```

32.5
133
12310

In [3]:

```
10 + e
```

Out[3]: 20

¿Con qué vamos a trabajar?

The screenshot shows the Spyder Python IDE interface. The top menu bar includes Archivo, Editar, Buscar, Código fuente, Ejecutar, Depurar, Terminales, Proyectos, Herramientas, Ver, Ayuda. The toolbar has icons for file operations like Open, Save, Run, and Stop. The status bar shows the path: AR538KL\OneDrive - EY\Documentos\Curso_Prod_TL\Teresa\elCurso DA_I\paraPPTX. The code editor window contains the file ejemSpyder.py with the following content:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Dec 2 12:57:30 2019
4
5 @author: T.Solá
6 """
7
8 e = 10
9 f = 22.5
10 b = True
11 s = '123'
12
13 ef = e + f
14 es = e + int(s)
15 se = s + str(e)
16
17 10 + e
```

To the right of the code editor is the "Explorador de variables" (Variables Explorer) window, which displays the following variable table:

Nombre	Tipo	Tamaño	Valor
b	bool	1	True
e	int	1	10
ef	float	1	32.5
es	int	1	133
f	float	1	22.5
s	str	1	123
se	str	1	12310

Below the variables explorer is the "Terminal de IPython" (IPython Terminal) window, which shows the following output:

```
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

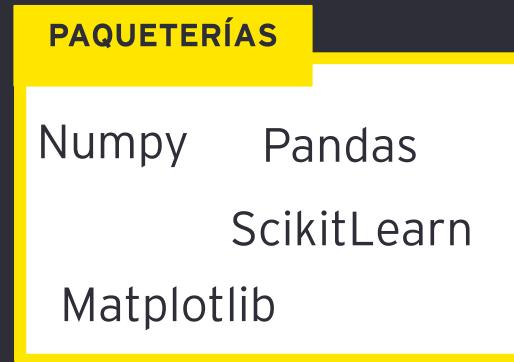
IPython 7.8.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/AR538KL/OneDrive - EY/Documentos/Curso_Prod_TL/Teresa/elCurso DA_I/paraPPTX/ejemSpyder.py', wdir='C:/Users/AR538KL/OneDrive - EY/Documentos/Curso_Prod_TL/Teresa/elCurso DA_I/paraPPTX')

In [2]: e * 123
Out[2]: 1230

In [3]:
```

¿Cómo funciona el ambiente integrado de Anaconda o Google Colab?



ANACONDA/ GOOGLE COLAB



Documentación

Documentación de Python:
<https://docs.python.org/3/>

Documentación de Anaconda:
<https://docs.anaconda.com/>



Import this

2. Tipos de datos

Estructuras iterables, slicing



*“Data, data data. I cannot make
bricks without clay.”*

Sherlock Holmes



The better the question. The better the answer.
The better the world works.



Construyendo un mejor
entorno de negocios

Conceptos generales

Operadores:

- ▶ Aritméticos: +, -, *, /, //, %, **
- ▶ Concatenación: *, +
- ▶ Comparación o igualdad: <, >, <=, >=, ==, !=
- ▶ Lógicos: and, or, not
- ▶ Pertenencia: in, not in
- ▶ Identidad: is, is not
- ▶ Asignación aumentada: +=, -=, *=, /=, %=, //=

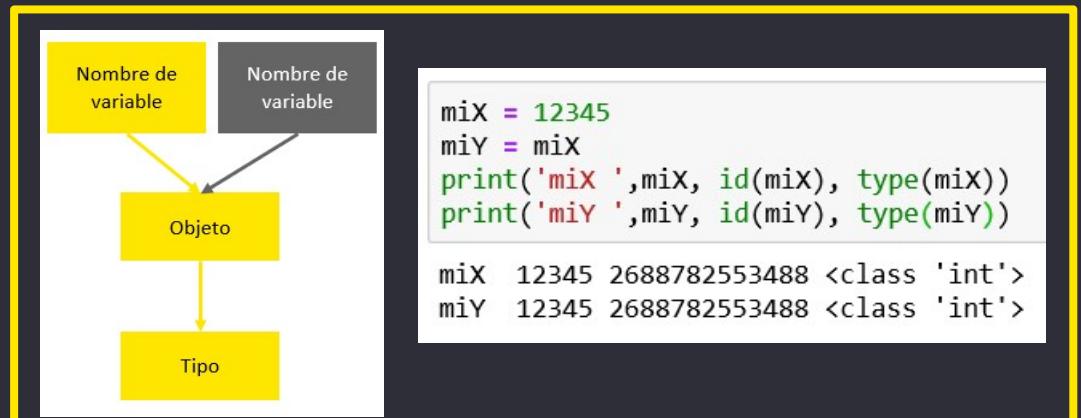
Conceptos generales

- ▶ Todas las variables de **Python** son objetos.
- ▶ El tipo del objeto determina la **mutabilidad** del valor que contiene.

- ▶ **Mutables:**

- ▶ Listas
- ▶ Conjuntos
- ▶ Diccionarios

Su valor puede cambiar



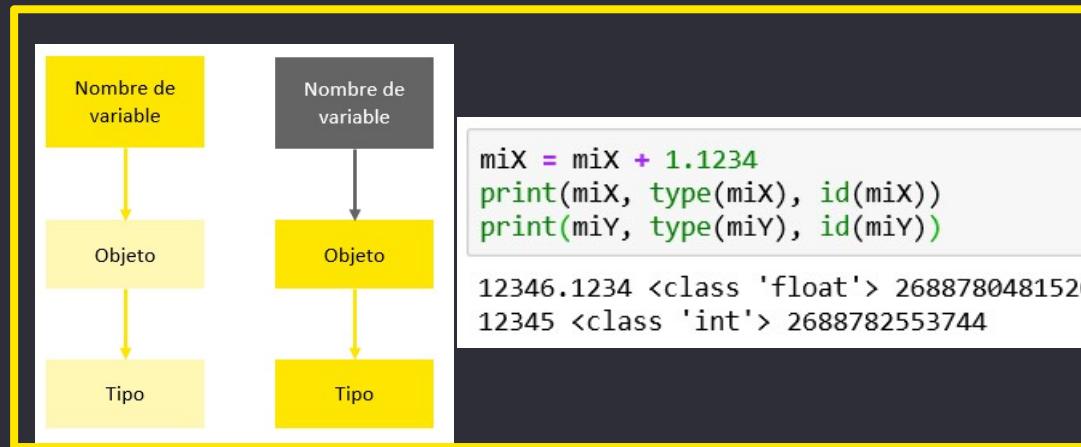
```
miX = 12345
miY = miX
print('miX ',miX, id(miX), type(miX))
print('miY ',miY, id(miY), type(miY))

miX 12345 2688782553488 <class 'int'>
miY 12345 2688782553488 <class 'int'>
```

- ▶ **Inmutables:**

- ▶ Números,
- ▶ Textos,
- ▶ Tuplas

Su valor **NO** puede cambiar



```
miX = miX + 1.1234
print(miX, type(miX), id(miX))
print(miY, type(miY), id(miY))

12346.1234 <class 'float'> 2688780481520
12345 <class 'int'> 2688782553744
```

Retomaremos la mutabilidad
después de revisar los tipos
de datos básicos e iterables
con los que cuenta Python.



Operadores lógicos

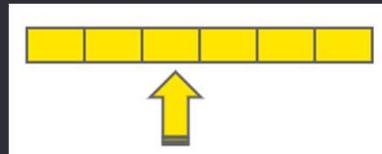


Iterables

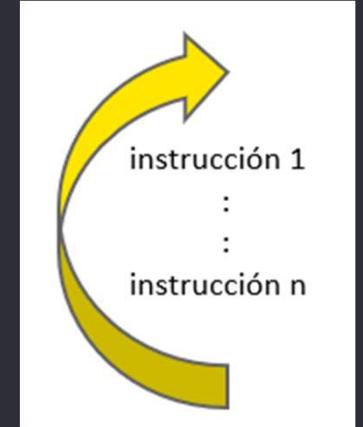
Conceptos Generales

Estructuras Algorítmicas y/o Slicing

- ▶ **Índice:** una posición



- ▶ **Ciclo:** permite repetir una secuencia de instrucciones



- ▶ **Iterable:** permite recorrer uno a uno los elementos de una estructura de datos.
- ▶ **Iteración:** es la idea de repetir un proceso sobre una secuencia de elementos.

Iterables (Texto)

- ▶ String (str) delimitador comillas simples “ ” o compuestas “”
Es una secuencia de caracteres utilizada para almacenar caracteres alfanuméricos y símbolos, por lo cual también se le conoce como cadena de caractéres.
 - ▶ Es **inmutable**.
 - ▶ Contiene métodos asociados para la manipulación de strings.
 - ▶ Un char es un string de longitud 1

```
strA = 'hola HSBC'
strB = strA

print( strA )
print( strB )

strA = strA.replace('h', 'j h')
print( strA )
print( strB )
```

hola HSBC
hola HSBC
; hola HSBC
hola HSBC

```
strC='Ciudad de México'
print(strC[0])
```

C

```
abc ='bonjour'
```

abc.

- capitalize
- casefold
- center
- count
- encode
- endswith
- expandtabs
- find
- format
- format_map

Iterables (Lista)

- ▶ Listas (list) delimitador []
 - ▶ Es una secuencia ordenada de elementos heterogéneos.
 - ▶ Es **mutable**.
 - ▶ Sus elementos pueden ser ingresados usando el índice o mediante *slicing*.
 - ▶ El tamaño de las lista no es fijo.
 - ▶ Cuenta con métodos asociados su manipulación.

```
In [6]: lstA = [3,6,10,100]
          lstA[0:2]

Out[6]: [3, 6]
```

```
lstA=[1,2]
lstA.

append
clear
copy
count
extend
index
insert
pop
remove
reverse ▾
```

Iterables (Tupla)

- ▶ Tupla (tuple) delimitador (), aunque basta con la coma para construirlas.
 - ▶ Es **inmutable** y está compuesta por una secuencia de elementos ordenados e inmutables o listas.
 - ▶ Se pueden acceder sus elementos usando el índice o mediante “slicing”, pero no pueden ser modificados.
 - ▶ Una lista dentro de la tupla es mutable.

```
tupla = (1, 'python', True)
print( type(tupla) )
print( tupla[1] )
print(tupla)
```



```
<class 'tuple'>
python
(1, 'python', True)
```

Iterables (Conjuntos)

- ▶ Conjunto (set) delimitador {}
 - ▶ Datos no repetidos, sin orden.
 - ▶ **Mutables**
 - ▶ Solo acepta datos de tipo inmutables.
 - ▶ Permite la operaciones entre conjuntos mediante operadores o métodos.

```
conj1=set(range(2,12,2))
conj2=set(range(3,15,3))
print(conj1)
print(conj2)

{2, 4, 6, 8, 10}
{9, 3, 12, 6}
```

```
lstA = [1,2,3,1,5,3,3,2,2]
set(lstA)
```

```
{1, 2, 3, 5}
```

```
conj1 | conj2 #Unión de conjuntos
```

```
{2, 3, 4, 6, 8, 9, 10, 12}
```

```
conj1 & conj2 #Intersección de conjuntos
```

```
{6}
```

Iterable (Diccionarios)

- ▶ Diccionario (dict) delimitador { llave: valor }
 - ▶ Estructura iterable, donde la llave es **inmutable** y el valor **mutable**, tiene mecanismos de acceso utilizando la llave. Las llaves pueden número, cadenas de caracteres, booleanos o tuplas.
 - ▶ Las llaves son única en el diccionario, ya que son el mecanismo de acceso.

```
Mi_diccionario={'País':'Francia','Cuenta':1620394, 'Credito_disponible':True}
```

```
Mi_diccionario['País']
```

```
'Francia'
```

```
Mi_diccionario.keys()
```

```
dict_keys(['País', 'Cuenta', 'Credito_disponible'])
```

Iterables (Slicing)

- ▶ Permite el acceso a porciones de las estructuras de datos string, tuple y list
- ▶ usando slicing. **x[start] : [stop] [: step]**, donde inicio, fin y paso son valores enteros los corchetes representan valores opcionales
- ▶ Inicia en el índice 0.

Slice	Resultado
[0:1]	primer elemento
:1]	primer elemento
[:-2]	del primer elemento todo excepto los 2 últimos
::2]	del primer elemento hasta el último con intervalos de 2
[-2:2:-2]	del penúltimo elemento al segundo con intervalos de 2

Iterables

Nombre del tipo de dato	Tipo en Python	Elementos	Inmutable / Mutable	Acceso
'texto'	<i>str</i>	alfanuméricos	inmutable	índices o slice
(tupla,)	<i>tuple</i>	de cualquier tipo	inmutable	índices o slice
[lista]	<i>list</i>	de cualquier tipo	mutable	índices o slice
{ conjunto }	<i>set</i>	elementos inmutables	mutable	no
diccionario { llave : valor }	<i>dict</i>	llave (tipo básico) o tupla o texto valor (cualquier tipo de dato)	inmutable llave-inmutable mutable-mutable	por las llaves