ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID

# Programación Python

José Javier Galán Hernández

ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID

# Programación Python

# 9. TUPLAS

# 9. TUPLAS

Parecidas a las listas, las tuplas no pueden modificarse.
No se escriben corchete [] (para tuplas se pueden usar paretesis ())
Se separan los elementos mediante comas ,.

In [34]:
```
tuplaNumeros = 1,2,3,4,5,6,7,8,9,10
print(tuplaNumeros)
```

```
(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
```

In [35]:
```
tuplaNumeros[1]=23
print(tuplaNumeros)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-35-84bbf407ff29> in <module>
----> 1 tuplaNumeros[1]=23
      2 print(tuplaNumeros)

TypeError: 'tuple' object does not support item assignment
```

Tupla de 1 elemento

In [36]:
```
tuplaNumeros = (1,)#tupla de un elemento se indica con coma , al final
print(tuplaNumeros)
```

```
(1,)
```

# 9. TUPLAS

Tupla vacia.

In [37]:
```
tuplaVacia = ()
print(tuplaVacia)
```

```
()
```

Acceso a tupla, similar a listas pero no se permite modificar

In [38]:
```
tuplaNumeros = (1,2,3,4,5,6,7,8,9,10)
print(tuplaNumeros[2])
print(tuplaNumeros[-2])
print(tuplaNumeros[2:4])
```

```
3
9
(3, 4)
```

ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID

# Programación Python

# 10. DICCIONARIOS

# 10. DICCIONARIOS

Son conjuntos de valores emparejados.

Los valores de cada par se separan mediante dos puntos (:), de otros pares por coma (,) y el conjunto por llaves ({})

Para acceder a un valor se accede por índice pero el índice puede no ser un numero.

```
In [56]: diccionario = {"nombre":"pepe","altura":"190","peso":"90","dni":"09040284P","direccion":"Madrid"}
         print(diccionario)

         {'nombre': 'pepe', 'altura': '190', 'peso': '90', 'dni': '09040284P', 'direccion': 'Madrid'}

In [57]: print(diccionario["nombre"])

         pepe

In [58]: print(diccionario["pepe"])#por el segundo elemento no se busca

         ---------------------------------------------------------------------------
         KeyError                                  Traceback (most recent call last)
         <ipython-input-58-18c3e9c3cf42> in <module>
         ----> 1 print(diccionario["pepe"])

         KeyError: 'pepe'
```

# 10. DICCIONARIOS

Podemos añadir

```
In [62]: diccionario.append("calle":"calle la goma")#Append no se usa con diccionario
         print(lista)
```

```
  File "<ipython-input-62-055d1c49f786>", line 1
    diccionario.append("calle":"calle la goma")#Indicamos que añada un valor mas en la lista creada
                               ^
SyntaxError: invalid syntax
```

```
In [65]: diccionario.update({"calle":"calle la goma"})#Si no existe lo crea
         print(diccionario)
```

```
{'nombre': 'Juan', 'altura': '190', 'peso': '90', 'dni': '09040284P', 'direccion': 'Madrid', 'calle': 'calle la goma'}
```

```
In [66]: diccionario.update({"calle":"calle la goma2"})#si existe modifica su valor
         print(diccionario)
```

```
{'nombre': 'Juan', 'altura': '190', 'peso': '90', 'dni': '09040284P', 'direccion': 'Madrid', 'calle': 'calle la goma2'}
```

y eliminar parejas de valores.

```
In [67]: del diccionario["calle"]
         print(diccionario)
```

```
{'nombre': 'Juan', 'altura': '190', 'peso': '90', 'dni': '09040284P', 'direccion': 'Madrid'}
```

ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID

# Programación Python

## 11. FOR

# 11. FOR

Se utiliza para recorrer objetos iteradores como listas o diccionarios.

Sintaxis:

**for** elemento **in** listaElementos**:**
    sentencias

Ejemplo para diccionarios:

In [68]:
```
diccionario = {"nombre":"pepe","altura":"190","peso":"90","dni":"09040284P","direccion":"Madrid"}
for palabra in diccionario:
    print(palabra)
```

```
nombre
altura
peso
dni
direccion
```

In [1]:
```
diccionario = {"nombre":"pepe","altura":"190","peso":"90","dni":"09040284P","direccion":"Madrid"}
for palabra in diccionario:
    print(diccionario[palabra])
```

```
pepe
190
90
09040284P
Madrid
```

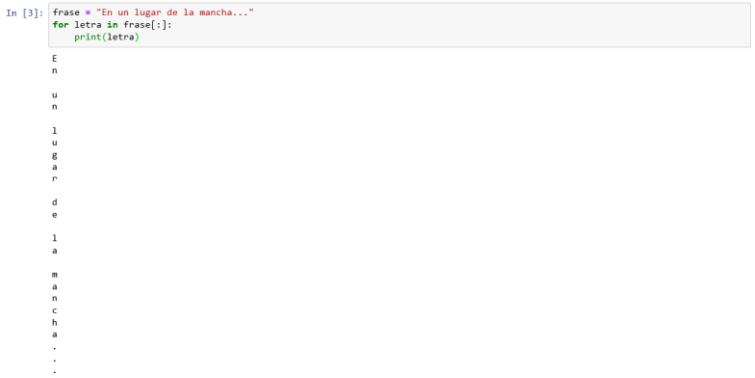
# 11. FOR

Ejemplo para listas:

In [2]:
```
diccionario =  ["uno","dos","tres","cuatro","cinco","seis","siete","ocho"]
for palabra in diccionario:
    print(palabra)
```

```
uno
dos
tres
cuatro
cinco
seis
siete
ocho
```

Ejemplo para cadenas:



ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID