

```
In [7]: #Ejercicio_04#  
#Encontrar todas las divisiones de un numero con los elementos de la lista#  
#ListaY=[6,6,7,9,0,3,3,5,0,5,8]
```

```
listaY=[6, 6, 7, 9, 0, 3, 3, 5, 0, 5, 8]
```

```
n = int(input("Ingrese un numero: "))
```

```
for i in range(len(listaY)-1):  
    try:  
        print(n/listaY[i])  
    except:  
        print("No existe la division entre cero")
```

```
Ingrese un numero: 12  
2.0  
2.0  
1.7142857142857142  
1.3333333333333333  
No existe la division entre cero  
4.0  
4.0  
2.4  
No existe la division entre cero  
2.4
```

```
In [17]: #-Hora, Minuto y Segundo-#
```

```
from datetime import time  
from datetime import datetime
```

```
hora1 = time(10, 35, 50)  
print(hora1)
```

```
fechaHora = datetime.now()  
print(fechaHora)  
print(type(fechaHora))
```

```
fecha = fechaHora.date()  
print(fecha)  
print(type(fecha))
```

```
hora1 = fechaHora.time()  
print(hora1)  
print(type(hora1))
```

```
h = fechaHora.hour  
print(h)  
m = fechaHora.minute  
print(m)  
s = fechaHora.second  
print(s)
```

```

10:35:50
2023-05-22 10:43:09.785394
<class 'datetime.datetime'>
2023-05-22
<class 'datetime.date'>
10:43:09.785394
<class 'datetime.time'>
10
43
9

```

In [24]: `from datetime import time`

```

hora2 = datetime.now()
#hora2 = time(11, 30, 5)
print(hora2)

formato = "Son las %H y %M con %S segundos"
hora2 = hora2.strftime(formato)
print(hora2)

```

```

2023-05-22 10:47:29.167213
Son las 10 y 47 con 29 segundos

```

In [33]: `fecha = input("Ingrese una fecha: ")`
`print(fecha)`

```

#-Convertir cadena en formato fecha-#
fecha = datetime.strptime(fecha, "%d/%m/%Y")
print(fecha)
print(type(fecha))

fecha = fecha.date()
print(fecha)
print(type(fecha))

#-Evaluacion de fecha como numero decimal-#
#fecha1 = eval(input("Ingrese otra fecha: "))

```

```

Ingrese una fecha: 22/5/1998
22/5/1998
1998-05-22 00:00:00
<class 'datetime.datetime'>
1998-05-22
<class 'datetime.date'>

```

In [45]: `###Diccionarios###`
`#Los identificamos con las llaves {}`

```

#Listas []
#Tuplas ()
#Diccionarios {}

dict1 = {}
print(type(dict1))

#Nombres: Claves, papel de indice#
#Numeros: Valores#
equipoF = {"Flores": 20, "Carrillo": 18, "Cueva": 10, "Lapadula": 11, "Gallese": 1}
print(equipoF)

print(equipoF["Carrillo"])
print(equipoF["Tapia"])

#Recorrido por for#

```

```
#Diccionarios por comprension#
```

```
for elem in equipoF:
    print(elem)
    print(equipoF[elem])
```

```
#Recorrido por comprension#
```

```
dic3 = {i:i*i for i in range(len(equipoF)-1)}
print(dic3)
```

```
<class 'dict'>
{'Flores': 20, 'Carrillo': 18, 'Cueva': 10, 'Lapadula': 11, 'Gallese': 1, 'Tapia':
24}
18
24
Flores
20
Carrillo
18
Cueva
10
Lapadula
11
Gallese
1
Tapia
24
{0: 0, 1: 1, 2: 4, 3: 9, 4: 16}
```

```
In [46]: vocal = {}
palabra = "auaue"

vocal["a"]=palabra.count("a")
print(vocal)

{'a': 2}
```

```
In [47]: #Ejercicio_05#
#Encontrar el numero de vocales de cada palabra#
#{'a': 5, 'e': 6, ...} usar diccionarios!!#

frase = "Los murcielagos aparecen por la noche junto a los arboles"

vocales = dict()

for ch in frase:
    print(ch, end = " ")

    vocales["a"] = frase.count("a")
    vocales["e"] = frase.count("e")
    vocales["i"] = frase.count("i")
    vocales["o"] = frase.count("o")
    vocales["u"] = frase.count("u")
```

L

```
-----  
AttributeError                                Traceback (most recent call last)  
Cell In[47], line 12  
      9 for ch in frase:  
     10     print(ch, end = " ")  
----> 12     vocales["a"] = frase.cont("a")  
     13     vocales["e"] = frase.cont("e")  
     14     vocales["i"] = frase.cont("i")  
  
AttributeError: 'str' object has no attribute 'cont'
```

In []: