ESTRUCTURA DE UN PROGRAMA

# I. Gracia, P. García, A. López Junio, 2023

**#ProDigital**

**millas.py**

## Archivo con extensión «.py»

millas = float(input('Introduce las millas: ')) km = millas \* 1.609

print(f'Equivale a {km} kilómetros')

## Este programa contiene 3 instrucciones

**millas.py**

millas = float(input('Introduce las millas: '))

km = millas \* 1.609

print(f'Equivale a {km} kilómetros')

## Esta instrucción

Esta instrucción pide al usuario el número de millas

## Esta instrucción muestra el resultado

calcula los kilómetros

Run: millas

Introduce las millas: \_

## Cuando ejecutemos el programa, aparecerá este mensaje…

…y el programa se queda esperando a que se introduzca un valor

Si escribimos un valor

y pulsamos la tecla

«Intro»…

Introduce las millas: 2.5

Run: millas

Run: millas

Introduce las millas: 2.5 Equivale a 4.0225 kilómetros


## …el programa calcula los kilómetros, muestra el resultado y termina

**millas.py**

## En un programa, además de instrucciones, podemos incluir comentarios

# Programa para convertir millas en kilómetros millas = float(input('Introduce las millas: ')) km = millas \* 1.609 # calcular km

print(f'Equivale a {km} kilómetros')

## El símbolo # indica que el resto de la línea es un comentario

Run: millas

Introduce las millas: 7.7 Equivale a 12.3893 kilómetros

## Cuando se ejecuta un programa, los comentarios son ignorados

**millas.py**

# Entrada de datos

millas = float(input('Introduce las millas: ')) # Cálculos

km = millas \* 1.609

# Mostrar resultados print(f'Equivale a {km} kilómetros')


## Un programa consta típicamente de estas tres partes

**millas.py**

# Entrada de datos

millas = float(input('Introduce las millas: '))

# Cálculos

km = millas \* 1.609 # Mostrar resultados

print(f'Equivale a {km} kilómetros')

## Primero, el programa pide los datos necesarios para su propósito

**millas.py**

# Entrada de datos

millas = float(input('Introduce las millas: ')) # Cálculos

km = millas \* 1.609

# Mostrar resultados

## En segundo lugar, hay que hacer los

print(f'Equivale a {km} kilómceátlrcouslo's) necesarios para conseguir

## los resultados deseados

**millas.py**

# Entrada de datos

millas = float(input('Introduce las millasP:or'ú)l)timo, hay

## que mostrar los

# Cálculos

km = millas \* 1.609 # Mostrar resultados

print(f'Equivale a {km} kilómetros')

## resultados

**millas.py**

# Entrada de datos

## ¡El orden importa!

= float(input('Introduce las millas: ')) # Cálculos

km

millas

= millas \* 1.609 # Mostrar resultados

print(f'Equivale a {km} kilómetros')

**segundos.py**

|  |
| --- |
| # Entrada de datoshoras = int(input('Introduce el número de horas: ')) minutos = int(input('Introduce el número de minutos: ')) |
| # Cálculossegundos = horas \* 3600 + minutos \* 60 |
| # Mostrar resultadosprint(f'Hay {segundos} segundos en total') |

Run: segundos

Introduce el número de horas: 2 Introduce el número de minutos: 15 Hay 8100 segundos en total

## Pide el número de horas, y escribimos 2

Pide el número de minutos, y escribimos 15


## Y, tras realizar los cálculos, muestra el resultado

**saludo.py**

# Entrada de datos

nombre = input('Introduce tu nombre: ') # Mostrar resultados

print(f'¡Hola, {nombre}!')

Pide el nombre…

Run: saludo

Introduce tu nombre: Laura

¡Hola, Laura!

…y escribe el saludo

# Isabel Gracia, Pedro García-Sevilla, Angeles López gracia@uji.es, pgarcia@uji.es, lopeza@uji.es

**#ProDigital**