



universidad  
cenfotec\_  
La U de la informática

# Principios de Programación.

Introducción al lenguaje de programación Python.

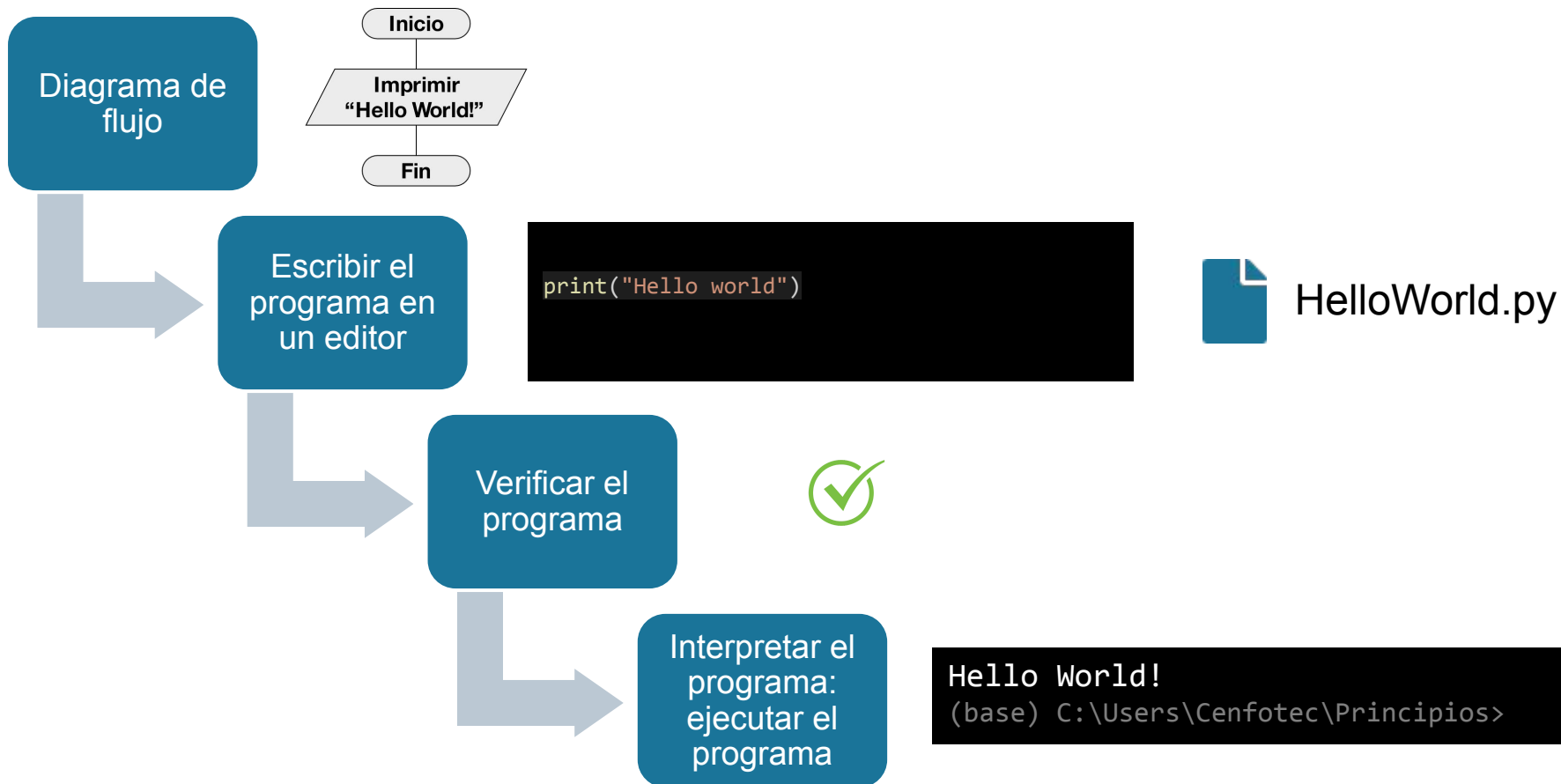
# Objetivos

- Conocer la historia del lenguaje de programación
- Conocer la estructura del lenguaje.
- Comprender y usar los conceptos de editar, interpretar y ejecutar un programa escrito en Python.

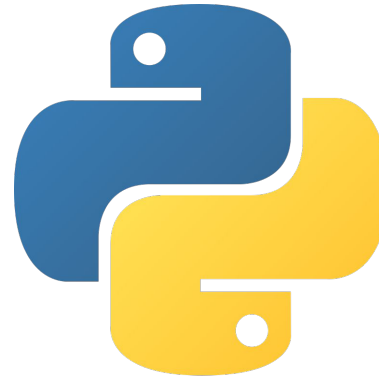
## Metodología de solución de un problema:

1. Definición del problema.
2. Análisis del problema: entender el problema y hacer modelado.
3. Diseño de la solución: hacer el algoritmo que solucione el problema.
4. Implementación: del programa correspondiente al algoritmo en lenguaje de programación.
5. Principios de control de calidad: hacer revisiones, pruebas y mantenimiento si es necesario.

# Implementación en Python



# El lenguaje Python



Es un lenguaje de programación de muy alto nivel creado en 1990 por Guido van Rossum.

Se considera un lenguaje de propósito general que se suele comparar con C, C++, TCL y Ruby.

Los programas desarrollados en Python son ejecutados por un intérprete (es interpretado), haciendo que la programación sea fácil, al no haber que compilar ni enlazar el código.

# El lenguaje Python



Actualmente se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation (PSF).

El nombre del lenguaje proviene de la admiración que sentía Rossum por los humoristas británicos del grupo Monty Python.

# Formas de uso de Python

Se puede utilizar de dos formas:

- Modo línea de comandos.
- Modo programa escrito.

Editor de Python online:

- <https://repl.it/languages/python3>

Consola de Python online:

- <https://www.python.org/shell/>

# Python en línea de comandos

```
Simbolo del sistema
Microsoft Windows [Versión 10.0.17763.557]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

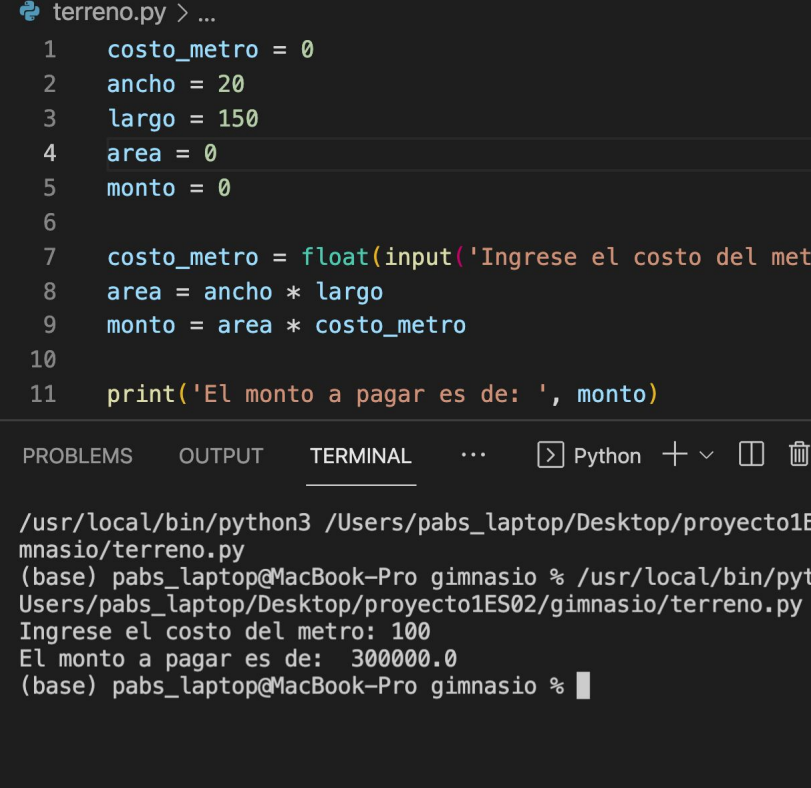
C:\Users\gchav>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("hola mundo!")
hola mundo!
>>> 5+5
10
>>> a = 50
>>> b = 10
>>> a+b
60
>>> frase = "Hola, como estan?"
>>> print(frase)
Hola, como estan?
>>> exit()

C:\Users\gchav>
```



# Python en programa escrito

\* **terreno.py** es el nombre del archivo de Python, el cual se encuentra dentro de la plataforma en la sección de ejemplos de código en semana 1.



```
terreno.py ×
terreno.py > ...
1 costo_metro = 0
2 ancho = 20
3 largo = 150
4 area = 0
5 monto = 0
6
7 costo_metro = float(input('Ingrese el costo del metro: '))
8 area = ancho * largo
9 monto = area * costo_metro
10
11 print('El monto a pagar es de: ', monto)
```

PROBLEMS OUTPUT TERMINAL ... Python + - □ □ ×

```
/usr/local/bin/python3 /Users/pabs_laptop/Desktop/proyecto1ES02/gimnasio/terreno.py
(base) pabs_laptop@MacBook-Pro gimnasio % /usr/local/bin/python3 /Users/pabs_laptop/Desktop/proyecto1ES02/gimnasio/terreno.py
Ingrese el costo del metro: 100
El monto a pagar es de: 300000.0
(base) pabs_laptop@MacBook-Pro gimnasio %
```

Python 3.9.4 64-bit 0 0 UTF-8 LF Python Port : 5501

# Conceptos importantes de Python

## imprimir\_fecha.py

```
1 import datetime
2
3 fecha_actual = datetime.datetime.now()
4 print(fecha_actual)
```

### import

Python tiene una gran cantidad de paquetes ya creados con funcionalidades distintas, organizados en varios **módulos**. Para utilizar estos recursos se debe importar en el archivo fuente utilizando la palabra reservada **import**. En este caso, se está importando **datetime**, que corresponde a las funcionalidades para interpretar y utilizar fechas.

# Conceptos importantes de Python

## numero\_mayor.py

```
1 primer_numero = 7
2 segundo_numero = 3
3 if primer_numero == segundo_numero:
4     print('Ambos números son iguales')
5 else:
6     if primer_numero > segundo_numero:
7         print('El número mayor es: ', primer_numero)
8     else:
9         print('El número mayor es: ', segundo_numero)
```

### Indentación

**Indentación** es una práctica que consiste en agregar **tabulaciones** o **espacios en blanco** al inicio de cada instrucción con el objetivo que el contenido del programa se estructure y así mejorar la comprensión del mismo. Este se puede comparar con el concepto de sangría en documentos.

En Python estos espacios en blanco **son obligatorios** para que el programa se pueda ejecutar correctamente. Sirve para indicar que una instrucción de código se va a ejecutar como “resolución” a una condición o rutina.

# Conceptos importantes de Python

## imprimir\_nombre.py

```
1 nombre = input('Ingrese su nombre: ')\n2 print('Hola su nombre es: ', nombre)
```

`input()`

En la línea 1 se utiliza la función **input** que permite recuperar una línea escrita usando el teclado, esta instrucción recupera una cadena de caracteres y la guarda en la variable nombre.

`print()`

Es una instrucción que permite imprimir mensajes en pantalla, en este caso se imprime el texto “Hola su nombre es: ” , acompañado del contenido de la variable nombre.

# Conceptos importantes de Python

## imprimir\_nombre.py

```
1 nombre = ''
2 edad = 0
3 estatura = 0.0
4
5 nombre = input('Ingrese su nombre: ')
6 edad = int(input('Ingrese su edad: '))
7 estatura = float(input('Ingrese su estatura en metros: '))
```

En este ejemplo se puede observar que en la línea número 1 se crea una variable de tipo **String** (cadena de caracteres o texto) llamada *nombre*, en la línea 2 una variable de tipo entero llamada *edad* y en la línea número 3 una variable de tipo float (números con decimales) llamada *estatura*.

En la línea número 5 se lee del teclado una cadena de caracteres y se asigna ese valor directamente a la variable **nombre**.

En la línea número 6 se lee también una cadena de caracteres, pero **edad** al ser una variable de tipo entero, es necesario convertir dicha cadena a un valor entero mediante la instrucción **int( )**.

Este proceso se repite para la variables **estatura**, convirtiendo las cadenas de caracteres a su respectivo tipo de dato, mediante las instrucciones **float( )**.



universidad  
cenfotec\_  
La U de la informática