

# Introducción a la programación con Python



# Elementos de un programa

Un programa de Python es un fichero de texto que contiene expresiones y sentencias del lenguaje Python. Esas expresiones y sentencias se consiguen combinando los elementos básicos del lenguaje.

El lenguaje Python está formado por elementos (*tokens*) de diferentes tipos:

- palabras reservadas (*keywords*)
- funciones integradas (*built-in functions*)
- literales
- operadores
- delimitadores
- identificadores



# Elementos de un programa

Para que un programa se pueda ejecutar, el programa debe ser sintácticamente correcto, es decir, utilizar los elementos del lenguaje Python respetando sus reglas de "ensamblaje".

Que un programa se pueda ejecutar no significa que vaya a realizar la tarea deseada.



# Elementos de un programa

Un programa de Python está formado por líneas de texto.

```
radio = 5  
area = 3.14159242 * radio ** 2  
print(area)
```



# Elementos de un programa

Se recomienda que cada línea contenga una única instrucción, aunque puede haber varias instrucciones en una línea, separadas por un punto y coma (;).

```
radio = 5; area = 3.14159242 * radio ** 2  
print(area)
```



# Elementos de un programa

Los elementos del lenguaje se separan por espacios en blanco (normalmente, uno), aunque en algunos casos no se escriben espacios:

- entre los nombres de las funciones y el paréntesis
- antes de una coma (,)
- entre los delimitadores y su contenido (paréntesis, llaves, corchetes o comillas)

Excepto al principio de una línea, los espacios no son significativos, es decir, da lo mismo un espacio que varios.



# Elementos de un programa

Los espacios al principio de una línea (el sangrado) son significativos porque indican un nivel de agrupamiento.

El sangrado inicial es una de las características de Python que lo distinguen de otros lenguajes, que utilizan un caracter para delimitar agrupamientos (en muchos lenguajes se utilizan las llaves { }).

Por ello, una línea no puede contener espacios iniciales, a menos que forme parte de un bloque de instrucciones o de una instrucción dividida en varias líneas.



# Elementos de un programa

## Palabras reservadas (keywords)

Las palabras reservadas de Python son las que forman el núcleo del lenguaje Python. Son las siguientes:

False await else import pass

None break except in raise

True class finally is return

and continue for lambda try

as def from nonlocal while

assert del global not with

async elif if or yield

Estas palabras no pueden utilizarse para nombrar otros elementos (variables, funciones, etc.), aunque pueden aparecer en cadenas de texto.





# Elementos de un programa

## Literales

Los literales son los datos simples que Python es capaz de manejar:

- números: valores lógicos, enteros, decimales y complejos, en notación decimal, octal o hexadecimal
- cadenas de texto



# Elementos de un programa

## Operadores

Los operadores son los caracteres que definen operaciones matemáticas (lógicas y aritméticas). Son los siguientes:

+   -   \*   \*\*   /   //   %   @

<<   >>   &   |   ^   ~

<   >   <=   >=   ==   !=



# Elementos de un programa

## Delimitadores

Los delimitadores son los caracteres que permiten delimitar, separar o representar expresiones. Son los siguientes:

' " # \

( ) [ ] { }

, : . ; @ = ->

+= -= \*= /= //= %= @=

&= |= ^= >>= <<= \*\*=



# Elementos de un programa

## Identificadores

Los identificadores son las palabras que se utilizan para nombrar elementos creados por el usuario u otros usuarios. Esos elementos pueden ser variables u objetos que almacenan información, funciones que agrupan instrucciones, clases que combinan ambos, módulos que agrupan los elementos anteriores, etc.

Los identificadores están formados por letras (mayúsculas y minúsculas), números y el carácter guión bajo (\_).

El primer caracter del identificador debe ser una letra.



# Elementos de un programa

## Funciones integradas (built-in functions)

Una función es un bloque de instrucciones agrupadas, que permiten reutilizar partes de un programa.

Python incluye algunas funciones que están disponibles sin necesidad de importar ninguna biblioteca:

# Elementos de un programa

## Funciones integradas (built-in functions)

<b>A</b> <u>abs()</u> <u>aiter()</u> <u>all()</u> <u>anext()</u> <u>any()</u> <u>ascii()</u>	<b>C</b> <u>callable()</u> <u>chr()</u> <u>classmeth</u> <u>od()</u> <u>compile()</u> <u>complex()</u>	<b>E</b> <u>enumerate</u> <u>()</u> <u>eval()</u> <u>exec()</u>	<b>H</b> <u>hasattr()</u> <u>hash()</u> <u>help()</u> <u>hex()</u>	<b>L</b> <u>len()</u> <u>list()</u> <u>locals()</u>	<b>O</b> <u>object()</u> <u>oct()</u> <u>open()</u> <u>ord()</u>	<b>R</b> <u>range()</u> <u>repr()</u> <u>reversed()</u> <u>)</u> <u>round()</u>	<b>T</b> <u>tuple()</u> <u>type()</u>
<b>B</b> <u>bin()</u> <u>bool()</u> <u>breakpoin</u> <u>t()</u> <u>bytearray</u> <u>()</u> <u>bytes()</u>	<b>D</b> <u>delattr()</u> <u>dict()</u> <u>dir()</u> <u>divmod()</u>	<b>F</b> <u>filter()</u> <u>float()</u> <u>format()</u> <u>frozenset</u> <u>()</u>	<b>I</b> <u>id()</u> <u>input()</u> <u>int()</u> <u>isinstanc</u> <u>e()</u> <u>issubclas</u> <u>s()</u> <u>iter()</u>	<b>M</b> <u>map()</u> <u>max()</u> <u>memoryvie</u> <u>w()</u> <u>min()</u>	<b>P</b> <u>pow()</u> <u>print()</u> <u>property()</u> <u>)</u>	<b>S</b> <u>set()</u> <u>setattr()</u> <u>slice()</u> <u>sorted()</u> <u>staticmet</u> <u>hod()</u> <u>str()</u> <u>sum()</u> <u>super()</u>	<b>V</b> <u>vars()</u>
		<b>G</b> <u>getattr()</u> <u>globals()</u>		<b>N</b> <u>next()</u>		<b>Z</b> <u>zip()</u>	<b>_</b> <u>import</u> <u>()</u>



# Elementos de un programa

## Funciones adicionales

Un programa puede definir nuevas funciones o redefinir las funciones integradas. Los nombres de las funciones no pueden coincidir con las palabras reservadas.

Un programa puede también importar nuevas funciones que se encuentran definidas en otros ficheros llamados módulos.

Python incluye una biblioteca de módulos (llamada Biblioteca estándar) especializados en todo tipo de tareas.



# Variables

Las variables son espacios en memoria o “cajas” donde podemos guardar valores y utilizarlas para por ejemplo, operar con ellas.

Las variables tienen un tipo, como por ejemplo “int”, “float”, “str”, “bool”, “list”.

Las variables en python se crean cuando se definen por primera vez, es decir, cuando se las da un valor.

Para asignar un valor a una variable se utiliza el operador de igualdad “=”

```
>>>  
>>> a = 4  
>>> a  
4  
>>> |
```





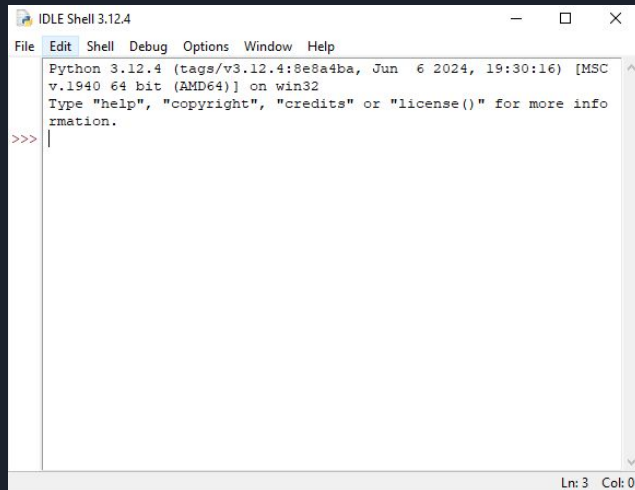
# Para empezar

Descargar e instalar python 3.12 y el entorno de desarrollo IDLE.

<https://www.python.org/downloads/>

**Download the latest version for Windows**

Download Python 3.12.4



The screenshot shows the IDLE Shell 3.12.4 window. The title bar reads "IDLE Shell 3.12.4". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following text: "Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32", "Type 'help', 'copyright', 'credits' or 'license()' for more information.", and a prompt ">>>". The status bar at the bottom right shows "Ln: 3 Col: 0".



# Para empezar

Funciones integradas:

- `print()` : Muestra el valor que se introduzca como argumento\*.
- `input()` : Muestra el mensaje del argumento\* y espera a que haya una entrada de datos por teclado.

argumento\*: Valores que se envían a las funciones introduciendolos entre los paréntesis de la llamada a dicha función.