

# FORMÁTICA GENERAL

## UCA - CICLO COMÚN INGENIERÍA

Unidad 01(PARTE B) - Entorno de desarrollo (IDE thonny). Lenguaje de programación. Concepto palabras reservadas. Escritura y ejecución de un programa básicos en python, con thonny. Elementos del lenguaje Python. Errores de sintaxis, de ejecución, de semántica.

## ÍNDICE

Entorno de desarrollo (IDE thonny).

Lenguaje de programación.

Concepto palabras reservadas.

Escritura y ejecución de un programa básicos en python, con thonny.

Elementos del lenguaje Python.

Errores de sintaxis, de ejecución, de semántica.

## IDE (Entorno de desarrollo integrado)

Un entorno de desarrollo integrado o entorno de desarrollo interactivo, en inglés Integrated Development Environment (IDE), es una aplicación informática que proporciona ciertos servicios integrados para facilitar al desarrollador o programador el desarrollo del software, es decir facilita la escritura de programas informáticos.

Existen muchas variedades y gustos de IDE. Nosotros utilizaremos un IDE simple que soporte el desarrollo de programa para el lenguaje de programación Python. El IDE seleccionado se llama “Thonny” ([www.thonny.org](http://www.thonny.org))

### Instalación:

Bajar el software “Thonny” de la página ([www.thonny.org](http://www.thonny.org)) o de acuerdo a sus sistema operativo hacer click en el link que corresponda:

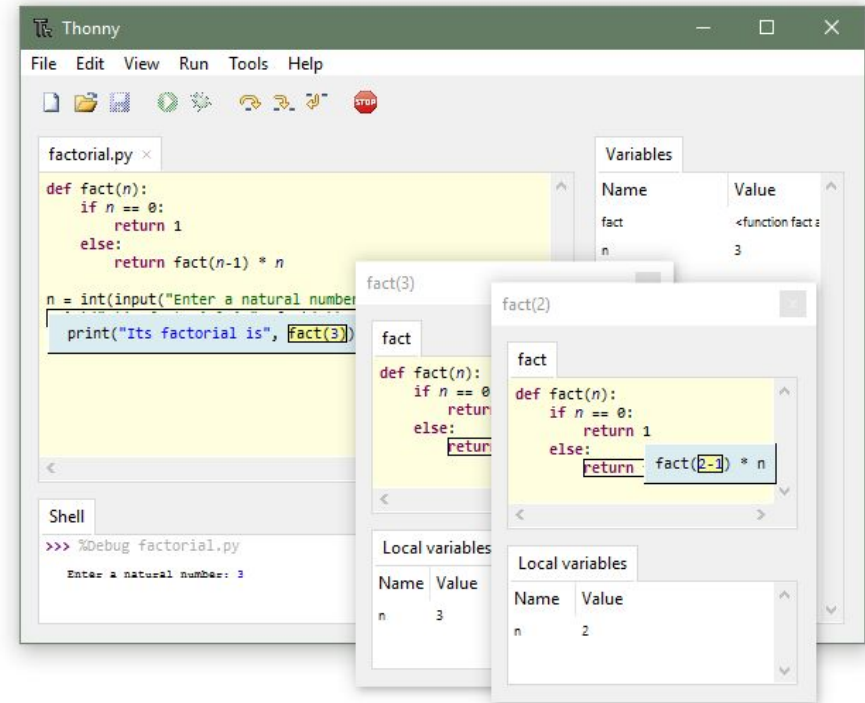
Descargue la versión **3.2.7** para

[Windows](#) • [Mac](#) • [Linux](#)

Su instalación es “sencilla”, practicamente es “next ...next” !!!

## IDE (Entorno de desarrollo integrado)

Normalmente, un IDE consiste de un editor de código fuente (*similar a un editor de texto*), y con el agregado de herramientas para compilar, ejecutar y depurar un programa.



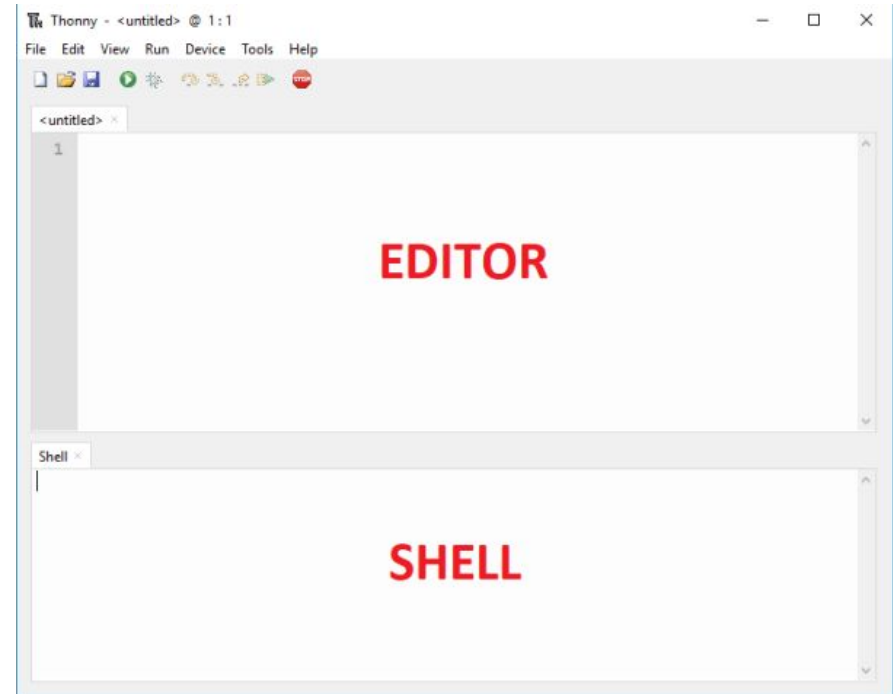
## IDE (Entorno de desarrollo integrado)

**El editor de código y Shell** Luego de instalado Thonny, abra la aplicación. Debería ver una ventana con varios iconos en la parte superior y dos áreas en blanco:

Observe las dos secciones principales de la ventana.

La sección superior **"Editor"** es el editor de código, donde escribirá todo el código del programa en lenguaje python. Para ejecutar el código hay que grabar previamente lo escrito en un archivo.

La mitad inferior **"Shell"**, es el Shell donde verá resultados de salida de su código. También puede ser utilizada para ejecutar comandos simples y ver su respuesta sin necesidad de cargarlos en un archivo.



## IDE (Entorno de desarrollo integrado)

### Los iconos

En la parte superior verás varios iconos. Exploremos lo que cada uno de ellos hace.

**A:** El ícono de papel que permite crear un nuevo archivo.

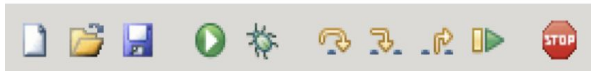
**B:** El ícono de abrir carpeta le permite abrir un archivo que ya existe en su computadora.

**C:** El icono del disquete le permite guardar su código.

**D:** El icono de reproducción te permite **ejecutar tu código**. Recuerde que el código que escribe está destinado a ser ejecutado. Ejecutar su código significa .....

**E:** El ícono le permite habilitar el modo depurador de código. ...

**A B C D E F G H I J**



**FH:** Los íconos de flecha le permiten ejecutar sus programas paso a paso. Esto puede ser muy útil cuando está depurando (*Debugger*) o, en otras palabras, tratando de encontrar esos errores en su código.

Estos íconos se usan después de presionar el ícono de error. Notarás que al presionar cada flecha, una barra resaltada en amarillo indicará qué línea o sección Python está evaluando actualmente:

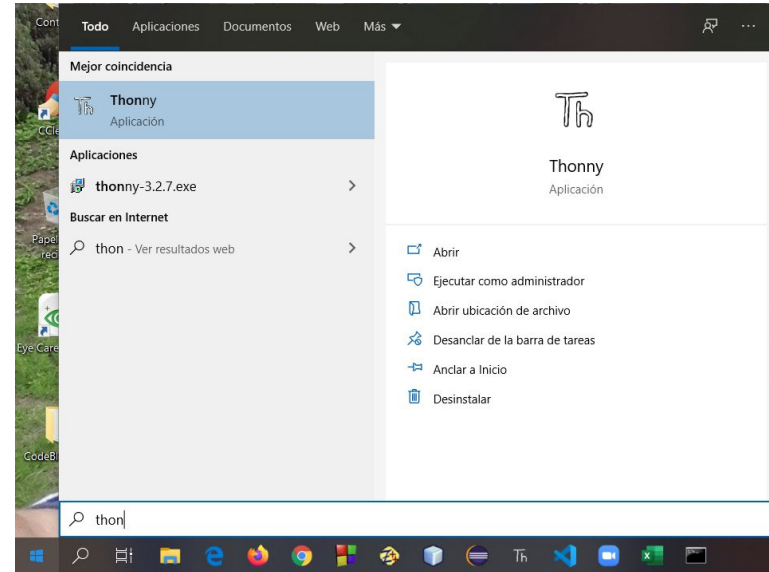
*La flecha **F** indica saltar a la siguiente línea o bloque de código. La flecha **G** indica sumergirse en cada componente de una expresión. La flecha **H** indica salir del depurador. **I:** El icono de reanudar le permite volver al modo de reproducción desde el modo de depuración.*

**J:** El ícono detener le permite dejar de ejecutar su código. Esto puede ser particularmente útil si, digamos, su código ejecuta y desea detener ese programa antes que este termine o bien porque el programa nunca termina.


Escritura y ejecución de un programa básicos en python, con thonny.

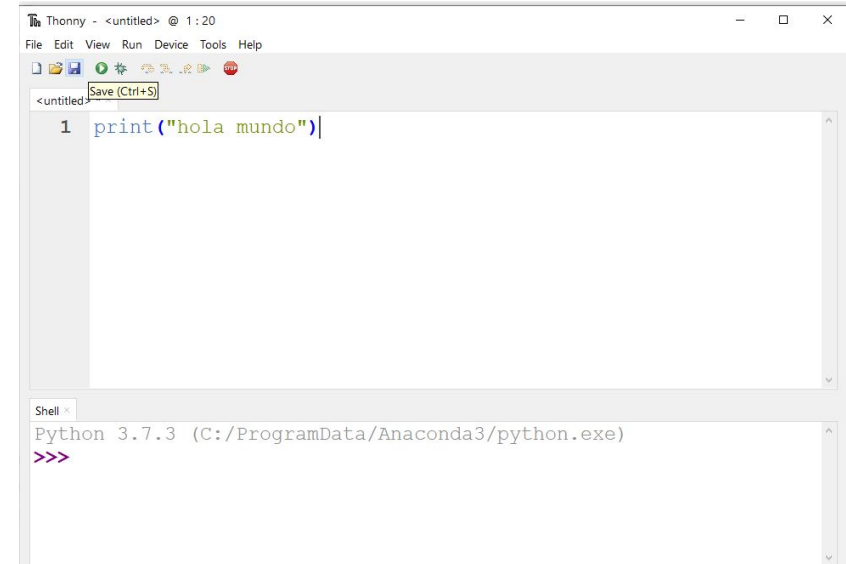
1. Previamente, instalar “Thonny”
2. Buscar el software “Thonny” dentro de tu computadora.
3. Ejecutar “Thonny”

*El detalle de la imagen a la derecha es la de un SO Windows 10*



## Escritura y ejecución de un programa básicos en python, con thonny.

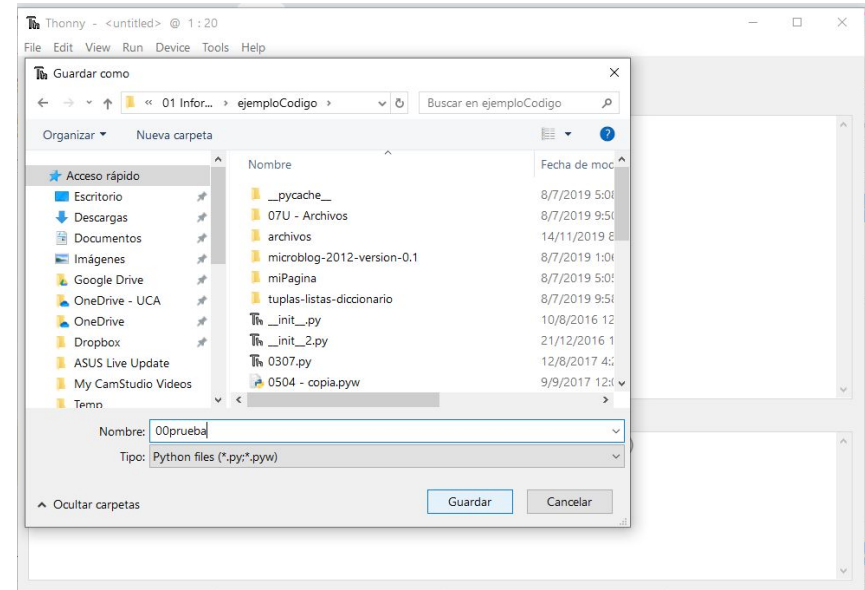
1. Escribir en la zona de edición la siguiente línea de código en python: `print("Hola mundo")`
2. Hacer click en  para guardar el archivo en la máquina con el contenido escrito en python






## Escritura y ejecución de un programa básicos en python, con thonny.

1. Colocar el nombre del archivo cual el cual quiero identificar el texto escrito en lenguaje python. En este caso el nombre que figura en la imagen es: `00python`
2. Luego hacer click en Guardar.



Estamo en condiciones de ejecutar el programa.

1. Hacer click en  para ejecutar el programa.
2. Se puede ver en el cuadro del shell (abajo) la salida producida por el programa.



The screenshot shows the Thonny IDE interface. The top menu bar includes File, Edit, View, Run, Device, Tools, and Help. Below the menu is a toolbar with icons for file operations and running the script. The main editor area displays a Python script named '00prueba.py' with the following code:

```
1 print("hola mundo")
```

Below the editor is a 'Shell' window showing the command prompt output:

```
Python 3.7.3 (C:/ProgramData/Anaconda3/python.exe)  
>>> %cd 'C:\Users\ASUS\Google Drive\NUBE\Docencia\UCA\_Materias\0  
1 Informática General'  
>>> %Run 00prueba.py  
hola mundo  
>>> |
```

## Elementos del lenguaje Python

Un programa de Python es un fichero de texto (normalmente guardado con el juego de caracteres UTF-8) 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

[https://docs.python.org/3/reference/lexical\\_analysis.html](https://docs.python.org/3/reference/lexical_analysis.html)

## Elementos del lenguaje Python

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

<b>False</b>	await	<b>else</b>	<b>import</b>	pass
<b>None</b>	break	except	<b>in</b>	raise
<b>True</b>	class	finally	is	<b>return</b>
<b>and</b>	continue	<b>for</b>	lambda	try
as	<b>def</b>	<b>from</b>	nonlocal	<b>while</b>
assert	<b>del</b>	global	<b>not</b>	with
async	<b>elif</b>	<b>if</b>	<b>or</b>	yield

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

En **negrita y sombreada gris** se encuentran **marcadas la que** utilizaremos en la cátedra.

## Elementos del lenguaje Python

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 del lenguaje Python

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

+	-	*	**	/	//	%	@
<<	>>	&		^	~		
<	>	<=	>=	==	!=		

## Elementos del lenguaje Python

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

'	"	#	\			
(	)	[	]	{	}	
,	:	.	;	@	=	->
+=	-=	*=	/=	//=	%=	@=
&=	=	^=	>>=	<<=	**=	

## Elementos del lenguaje Python

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 podrían estar formados por letras (mayúsculas y minúsculas), números y el carácter guión bajo (\_). Pueden ser caracteres [Unicode](#) ([tabla unicode](#)), aunque normalmente se recomienda utilizar caracteres [ASCII](#) ([tabla ascii](#)) para evitar complicaciones a usuarios de otros países que utilizan juegos de caracteres diferentes.

El primer carácter del identificador debe ser una letra.



## Elementos del lenguaje Python

**Funciones integradas (built-in functions).** Una función es un bloque de instrucciones agrupadas, que permiten reutilizar partes de un programa. Python incluye las siguientes funciones de forma predeterminada (es decir, estas funciones siempre están disponibles):

<code>__import__()</code>	<code>compile()</code>	<code>frozenset()</code>	<code>len()</code>	<code>ord()</code>	<code>staticmethod()</code>
<code>abs()</code>	<code>complex()</code>	<code>getattr()</code>	<code>list()</code>	<code>pow()</code>	<code>str()</code>
<code>all()</code>	<code>delattr()</code>	<code>globals()</code>	<code>locals()</code>	<code>print()</code>	<code>sum()</code>
<code>any()</code>	<code>dict()</code>	<code>hasattr()</code>	<code>map()</code>	<code>property()</code>	<code>super()</code>
<code>ascii()</code>	<code>dir()</code>	<code>hash()</code>	<code>max()</code>	<code>range()</code>	<code>tuple()</code>
<code>bin()</code>	<code>divmod()</code>	<code>help()</code>	<code>memoryview()</code>	<code>repr()</code>	<code>type()</code>
<code>bool()</code>	<code>enumerate()</code>	<code>hex()</code>	<code>min()</code>	<code>reversed()</code>	<code>vars()</code>
<code>bytearray()</code>	<code>eval()</code>	<code>id()</code>	<code>next()</code>	<code>round()</code>	<code>zip()</code>
<code>bytes()</code>	<code>exec()</code>	<code>int()</code>	<code>nput()</code>	<code>set()</code>	
<code>callable()</code>	<code>filter()</code>	<code>isinstance()</code>	<code>object()</code>	<code>setattr()</code>	
<code>chr()</code>	<code>float()</code>	<code>issubclass()</code>	<code>oct()</code>	<code>slice()</code>	
<code>classmethod()</code>	<code>format()</code>	<code>iter()</code>	<code>open()</code>	<code>sorted()</code>	

## Elementos del lenguaje Python

**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.

Además de la biblioteca estándar, existen miles de módulos escritos por diferentes programadores y accesibles en Internet. El principal repositorio de módulos es el [Python Package Index](#) (Índice de paquetes de Python), más conocido por PyPI.

FIN!