

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.



INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

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(*tokens*) :

- Palabras reservadas (*keywords*).
- Funciones integradas (*built-in functions*).
- Literales.
- Operadores.
- Delimitadores.
- Identificadores.

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Líneas y espacios.

Un programa de Python está formado por líneas de texto:

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

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)
```

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Líneas y espacios.

Por motivos de legibilidad, se recomienda que las líneas no superen los 79 caracteres. Si una instrucción supera esa longitud, se puede dividir en varias líneas usando el caracter contrabarra (\):

```
radio = 5
area = 3.14159265358979323846 \
    * radio ** 2
print(area)
```

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Líneas y espacios.

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).

```
semanas = 4  
print("En", semanas, "semanas hay", 7 * semanas, "días.")
```

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Líneas y espacios.

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

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 carácter 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.

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Líneas y espacios.

```
def sumar(number1, number2):  
    total = number1 + number2  
    return total
```

```
sumar(10, 15) # 25
```

```
int sumar(number1, number2){  
    int suma = number1 + number2;  
    return suma;  
}
```

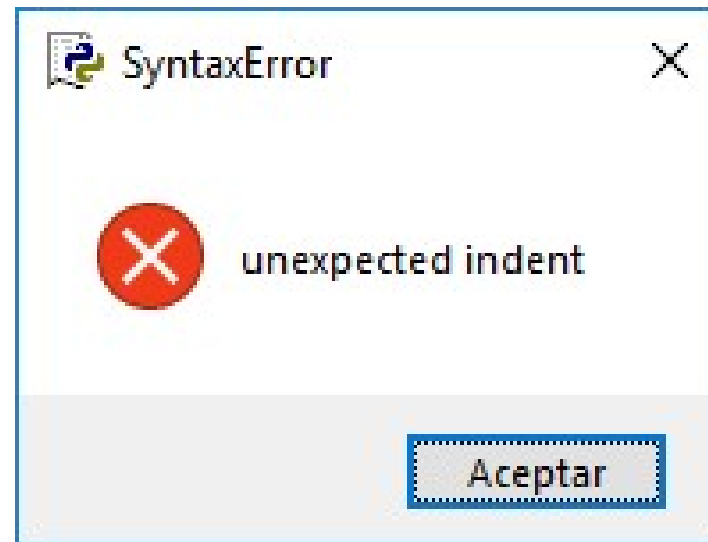
INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Líneas y espacios.

Al ejecutar en IDLE una instrucción con espacios iniciales, se mostraría un aviso de error de sintaxis: **SyntaxError: unexpected indent**

Al ejecutar un programa, se mostraría el mensaje de error de sintaxis:



INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Líneas y espacios.

El carácter almohadilla (#) marca el inicio de un comentario. Python ignora el resto de la línea (desde la almohadilla hasta el final de la línea).

```
area = 3.14159242 * radio ** 2    # Fórmula del área de un círculo
```

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Palabras reservadas (keywords).

Las palabras reservadas de Python son las que forman el núcleo del lenguaje Python. Estas palabras no pueden utilizarse para nombrar otros elementos (variables, funciones, etc.), aunque pueden aparecer en cadenas de texto.

Palabras reservadas de Python 3

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

Y con Python 3.7:

`async`
`await`

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Literales.

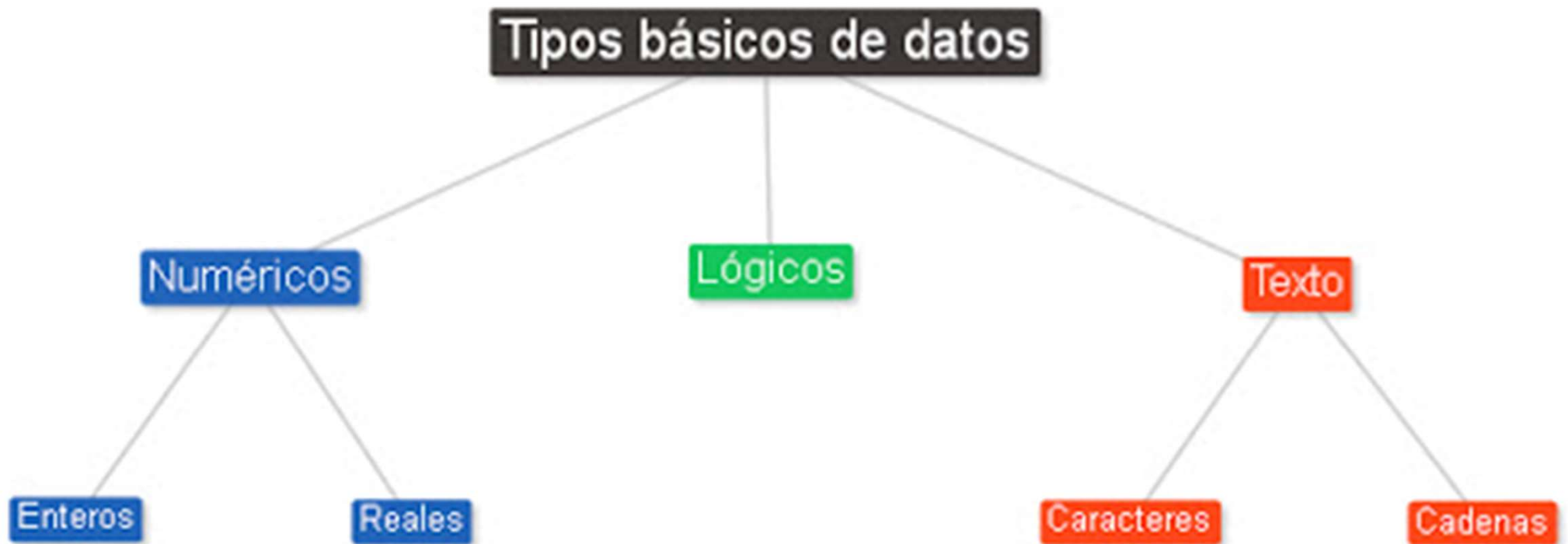
Los literales son los **datos simples** que Python es capaz de manejar:

- ✓ Números: valores lógicos, enteros, decimales y complejos, en notación decimal, octal o hexadecimal
- ✓ Cadenas de texto

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Literales.



INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Operadores.

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

Símbolo	Significado	Ejemplo	Resultado
+	Suma	<code>a = 10 + 5</code>	<code>a es 15</code>
-	Resta	<code>a = 12 - 7</code>	<code>a es 5</code>
-	Negación	<code>a = -5</code>	<code>a es -5</code>
*	Multiplicación	<code>a = 7 * 5</code>	<code>a es 35</code>
**	Exponente	<code>a = 2 ** 3</code>	<code>a es 8</code>
/	División	<code>a = 12.5 / 2</code>	<code>a es 6.25</code>
//	División entera	<code>a = 12.5 / 2</code>	<code>a es 6.0</code>
%	Módulo	<code>a = 27 % 4</code>	<code>a es 3</code>

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Operadores.

Signo del Operador	Nombre del Operador
<	Menor que
>	Mayor que
<=	Menor o igual a
>=	Mayor o igual a
==	Igual a
!=	Diferente a

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Delimitadores

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

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

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Delimitadores

```
print("¡Hola, mundo!")  
print("¡Adiós, mundo!")
```

```
edad = int(input("¿Cuántos años tiene? "))  
if edad < 18:  
    print("Es usted menor de edad")  
else:  
    print("Es usted mayor de edad")  
print("¡Hasta la próxima!")
```

```
lista = [1, "abcde", 45.5, -32]
```


INTRODUCCIÓN A LA PROGRAMACIÓN

Python – 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 **guion bajo** (_).

El primer carácter del identificador debe ser una **letra**.

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Variables

Una variable es un espacio reservado en la memoria de nuestro ordenador para almacenar un dato que puede ser usado o modificado tantas veces como se desee.



INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Variables

Las variables en Python **se crean cuando se definen por primera vez**, es decir, cuando se les asigna un valor por primera vez. Para asignar un valor a una variable se utiliza el operador de igualdad (=). A la izquierda de la igualdad se escribe el nombre de la variable y a la derecha el valor que se quiere dar a la variable.

En Python, una variable se define con la sintaxis:

```
nombre_de_la_variable = valor_de_la_variable
```

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Variables

En el ejemplo siguiente se almacena el número decimal 2.5 en una variable de nombre x. Fíjese en que los números decimales se escriben con punto (.) y no con coma (,).

```
x = 2.5
```

La variable se escribe siempre a la izquierda de la igualdad. Si se escribe al revés, Python genera un mensaje de error:

```
2.5 = x
```

SyntaxError: can't assign to literal

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Variables

Aunque no es obligatorio, se recomienda que **el nombre de la variable esté relacionado con la información que se almacena en ella**, para que sea más fácil entender el programa. Si el programa es trivial o mientras se está escribiendo un programa, esto no parece muy importante, pero si se consulta un programa escrito por otra persona o escrito por uno mismo pero hace tiempo, resultará mucho más fácil entender el programa si los nombres están bien elegidos.

```
fecha_de_nacimiento = "27 de octubre de 1997"
```

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Variables

- ✓ El nombre de una variable debe **empezar por una letra** o por un guion bajo (_) y puede seguir con más letras, números o guiones bajos.
- ✓ Los nombres de variables **no** pueden incluir **espacios en blanco**.
- ✓ Los nombres de variables pueden contener cualquier carácter alfabético (los del alfabeto inglés, pero también ñ, ç o vocales acentuadas), aunque se recomienda utilizar únicamente los caracteres del alfabeto inglés.
- ✓ Los nombres de las variables pueden contener mayúsculas, pero tenga en cuenta que Python distingue entre mayúsculas y minúsculas (en inglés se dice que Python es **case-sensitive**).

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Variables

- ✓ Cuando el nombre de una variable contiene varias palabras, se aconseja **separarlas con guiones bajos** para facilitar la legibilidad, aunque también se utiliza **la notación camelCase**, en las que las palabras no se separan pero empiezan con mayúsculas (salvo la primera palabra).
- ✓ Las **palabras reservadas** del lenguaje (las que IDLE escribe en naranja) **están prohibidas** como nombres de variables.
- ✓ Los nombres de las funciones integradas sí que se pueden utilizar como nombres de variables, pero más vale no hacerlo porque a continuación ya no se puede utilizar la función como tal. Borrando con **del** la variable con nombre de función, se recupera la función.

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Variables

Una variable puede almacenar números, texto o estructuras más complicadas (que se verán más adelante). Si se va a almacenar texto, el texto debe escribirse entre comillas simples (') o dobles ("), que son equivalentes. A las variables que almacenan texto se les suele llamar cadenas (de texto).

Algunos lenguajes de programación (C, C++, Java, etc) exigen que antes de utilizar una variable se defina el tipo de información que se va a guardar en esa variable. Otros lenguajes de programación (Python, PHP, etc.) no lo exigen y es el intérprete del lenguaje el que decide el tipo de variable a utilizar en el momento que se guarda la información.

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Funciones integradas (built-in functions).

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

Los nombres de las funciones integradas se pueden utilizar para nombrar variables, pero entonces las funciones ya no estarán disponibles en el programa. Si se eliminan las variables, las funciones vuelven a estar disponibles.

La instrucción **del** borra completamente una variable.

del nombre

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Funciones integradas (built-in functions).

Python incluye las siguientes funciones de forma predeterminada:

abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

Funciones integradas (built-in functions).

bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()		set()

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – 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.

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**.

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Elementos de un programa.

“Fin del tema”

