

### Identificadores

Nombres dados a objetos  
`mi_variable, área_1, _temp`

### Palabras Clave

No pueden usarse como identificadores

<code>and</code>	Operador lógico
<code>as</code>	Crea alias
<code>assert</code>	Depura
<code>break</code>	Sale de un loop
<code>class</code>	Define una clase
<code>continue</code>	Continúa la iteración de un loop
<code>def</code>	Define una función
<code>del</code>	Elimina un objeto
<code>elif</code>	Crea una condición
<code>else</code>	Crea una condición
<code>except</code>	Da lugar a una excepción
<code>false</code>	Valor booleano
<code>finally</code>	Siempre se ejecuta, independientemente de si hay una excepción o no
<code>for</code>	Crea un loop
<code>from</code>	Importa cosas específicas de un módulo
<code>global</code>	Declarar una variable global
<code>if</code>	Crea una condición
<code>import</code>	Para importar un módulo
<code>in</code>	Verifica si un valor específico está dentro de una secuencia
<code>is</code>	Prueba si dos variables son iguales
<code>lambda</code>	Crea una función anónima
<code>none</code>	Representa un valor null
<code>nonlocal</code>	Declara una variable no local
<code>not</code>	Operador Lógico
<code>or</code>	Operador Lógico
<code>raise</code>	Plantea una excepción

### Palabras Clave (cont)

<code>return</code>	Sale de una función y retorna un valor
<code>true</code>	Valor booleano
<code>try</code>	Comienza una sentencia
	<code>try...except</code>
<code>while</code>	Comienza un loop
<code>with</code>	Simplifica el manejo de excepciones
<code>yield</code>	Devuelve una lista de valores de un generador

### Booleanos

<code>True</code>	1
<code>False</code>	0
<code>es_verdad = True</code>	
<code>no_es_verdad = False</code>	

### Operadores Numéricos

<code>x + y</code>	Suma
<code>x - y</code>	Resta
<code>x * y</code>	Multiplicación
<code>x / y</code>	División, devuelve un flotante
<code>x // y</code>	División entera
<code>x % y</code>	Módulo o resto
<code>x y**</code>	Potencia
<code>-x</code>	Negativo
<code>+x</code>	Positivo
<code>abs(x)</code>	Valor absoluto de x
<code>divmod(x, y)</code>	Cociente y resto de x entre y
<code>pow(x, y)</code>	Eleva x a la potencia de y
<code>pow(x, y, z)</code>	Calcula (x ** y) % z
<code>round(x, n)</code>	Redondea x a n decimales

### Conversión de Enteros

<code>bin(i)</code>	Convierte i a binario
<code>hex(i)</code>	Convierte i a hexadecimal
<code>int(x)</code>	Convierte x a entero
<code>int(s, base)</code>	Convierte la cadena s a entero en la base especificada
<code>oct(i)</code>	Convierte i a octal

### Números Complejos

<code>a + bj</code>	Define un número complejo (a real, b imaginario)
<code>z.real</code>	Parte real de z
<code>z.imag</code>	Parte imaginaria de z
<code>z.conjugate()</code>	Cambia el signo de la parte imaginaria
<code>complex(x, y)</code>	Crea un número complejo a partir de x (real) y y (imaginario)

### Escapes de Strings

<code>\newline</code>	Escape
<code>\\</code>	Barra invertida (\)
<code>'</code>	Comilla simple (')
<code>"</code>	Comilla doble (")
<code>\a</code>	Campana ASCII
<code>\b</code>	Retrosceso ASCII
<code>\f</code>	Salto de página ASCII
<code>\n</code>	Nueva línea ASCII
<code>\N{name}</code>	Carácter Unicode con el nombre dado
<code>\ooo</code>	Carácter con valor octal ooo
<code>\r</code>	Retorno de carro ASCII
<code>\t</code>	Tabulación ASCII
<code>\uhhhh</code>	Carácter Unicode de 16 bits hhhh
<code>\Uhhhhhhh</code>	Carácter Unicode de 32 bits hhhhhhhh
<code>\v</code>	Tabulación vertical ASCII
<code>\xhh</code>	Carácter con valor hexadecimal hh



By **schmidtlady16**

Published 30th October, 2024.  
 Last updated 30th October, 2024.  
 Page 1 of 2.

Sponsored by **Readable.com**  
 Measure your website readability!  
<https://readable.com>

### Slicing y Striding de Strings

<code>s[i]</code>	Devuelve el carácter en la posición i
<code>s[start:end]</code>	Extrae una subcadena desde start hasta end (excluye end)
<code>s[start:end:step]</code>	Extrae caracteres desde start hasta end con un intervalo step
<code>s[:end]</code>	Extrae desde el inicio hasta end
<code>s[start:]</code>	Extrae desde start hasta el final de la cadena
<code>s[::-1]</code>	Invierte la cadena

### Operadores Bit a Bit

<code>i   j</code>	OR bit a bit entre i y j
<code>i ^ j</code>	XOR bit a bit entre i y j
<code>i &amp; j</code>	AND bit a bit entre i y j
<code>i &lt;&lt; j</code>	Desplaza i a la izquierda j bits
<code>i &gt;&gt; j</code>	Desplaza i a la derecha j bits
<code>~i</code>	Invierte los bits de i



By **schmidtlady16**

[cheatography.com/schmidtlady16/](https://cheatography.com/schmidtlady16/)

Published 30th October, 2024.

Last updated 30th October, 2024.

Page 2 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>