

# Clase 1 curso python para profesores 2026

Vamos a trabajar con el intérprete de python para ir familiarizándonos con el entorno. Para ello tecleamos python3.

En esta clase introduciremos el concepto de variable, cómo es la forma correcta de nombrar a las variables y tendremos una toma de contacto con pygame en la que aplicaremos lo aprendido y solicitaremos varios retos a los alumnos.

## 1.- Operadores matemáticos

Suma: +

Multiplicación: \*

Resta: -

Resto o módulo: % (tiene la particularidad de que el resultado siempre tiene el mismo signo que el divisor, utiliza la fórmula  $a \% b = a - b \cdot \text{floor}(a/b)$ )

Potencia: \*\*

División: /

División entera: //

Realiza, desde el intérprete de python, estos 10 ejercicios. Tendrás que introducir línea a línea. En breve veremos cómo crear un programa que nos permita no tener que volver a introducir las expresiones (definen o calculan un valor) / sentencias (indican una acción a ejecutar) cada vez.

### Ejercicio 1:

```
a = 15  
b = 8  
resultado = a + b  
print(resultado)
```

### Ejercicio 2:

```
x = 50  
y = 17  
resultado = x - y  
print(resultado)
```

### Ejercicio 3:

```
horas = 7  
salario_por_hora = 20  
total = horas * salario_por_hora
```

```
print(total)
```

#### Ejercicio 4:

```
total_dinero = 120
personas = 5
dinero_por_persona = total_dinero / personas
print(dinero_por_persona)
```

#### Ejercicio 5:

```
a = 23
b = 4
resultado = a // b
print(resultado)
```

#### Ejercicio 6:

```
a = 23
b = 4
resultado = a % b
print(resultado)
```

#### Ejercicio 7:

```
base = 2
exponente = 5
resultado = base ** exponente
print(resultado)
```

#### Ejercicio 8:

```
a = -13
b = 5
resultado = a % b
print(resultado)
```

#### Ejercicio 9:

```
resultado = 4 + 3 * 2 ** 2
print(resultado)
```

#### Ejercicio 10:

```
a = 20.5
b = 6
print(a / b)
print(a % b)
```

Observa que aparecen letras como a y b, o resultado, base, exponente, etc (las he marcado en rojo en el ejercicio. Se denominan variables y hablaremos de ellas un poco más adelante. También puedes ver que las “variables” que he nombrado con más de una palabra tienen la forma `dinero_por_persona` este estilo de nombrado de variables se denomina snake case y es el recomendado por python. Recomendado no quiere decir obligatorio

## **2.- Tipos de datos más comunes (¿sencillos también?)**

entero

flotante

lista

tupla

booleano

### **Ejercicio 1:**

```
x = 10  
y = 3.5  
resultado = x + y  
print(type(resultado))
```

### **Ejercicio 2:**

```
nombre = "Ana"  
saludo = "Hola, " + nombre  
print(saludo)
```

### **Ejercicio 3:**

```
mensaje = "Estoy aprendiendo python"  
print(mensaje * 4)
```

### **Ejercicio 4:**

```
numeros = [3, 5, 7, 9]  
print(numeros[2])
```

### **Ejercicio 5:**

```
colores = ["rojo", "verde", "azul"]  
colores[1] = "amarillo"  
print(colores)
```

### **Ejercicio 6:**

```
coordenadas = (4, 8)  
print(coordenadas[0])
```

### **Ejercicio 7: (Al ejecutar la segunda línea se produce un error. Explicaremos por qué)**

```
punto = (1, 2)  
punto[0] = 3
```

**Ejercicio 8:**

```
a = 7  
b = 10  
print(a > b)
```

**Ejercicio 9:**

```
es_mayor = True  
es_menor = False  
print(es_mayor and es_menor)
```

**Ejercicio 10:**

```
datos = [1, "uno", True, 3.14]  
print(type(datos[2]))
```

### 3.- Operadores lógicos

==

!=

in

and

or

>

<

>=

<=

**Ejercicio 1:**

```
a = 5  
b = 10  
print(a < 10 and b > 5)
```

**Ejercicio 2:**

```
x = 3  
y = 7  
print(x > 5 or y == 7)
```

**Ejercicio 3:**

```
activo = False  
print(not activo)
```

**Ejercicio 4:**

```
edad = 20  
tiene_permiso = False
```

```
print(edad >= 18 and not tiene_permiso)
```

### Ejercicio 5:

```
a = True  
b = False  
c = True  
print(a and (b or c))
```

## 4.- ¿Qué es una variable?

Si has realizado los ejercicios anteriores habrás visto que en ocasiones hacemos cosas como esta: a = 7.

En esa línea lo que hemos hecho ha sido decirle a python que cree una variable (una especie de caja en la que guardar datos) que se llame a y que en esa caja se guarde un valor 7

Esto va a hacer que, cada vez que utilicemos a realmente estemos utilizando el valor que tiene guardado la variable a.

Hemos visto la función type. Esta función nos dice el tipo del dato que, en ese momento, está guardando la variable a. Y digo en ese momento porque python, a diferencia de otros lenguajes de programación, permite que durante la ejecución del programa se cambie el dato guardado, pero también el tipo del dato guardado.

Para familiarizarnos con type vamos a ejecutar las siguientes líneas:

```
a = 7  
type(a)  
a="holá"  
type(a)
```

Vamos a estudiar las líneas del primerprograma.py que ya utilizamos para familiarizarnos con el editor VisualStudio Code. Las líneas son las siguientes:

```
# Solicita el nombre al usuario  
nombre = input("¿Cómo te llamas? ")  
  
# Muestra un saludo personalizado  
print(f"Hola, {nombre}!")
```

Cambia el programa para que la variable no se llame nombre sino numero y que el texto de la pregunta sea ¿introduce un número? Y que el print muestre el número introducido por el usuario multiplicado por 8

¿Se ha dado bien? ¿Por qué crees que se produce el resultado que se muestra en pantalla?

Bien, modifica el código para que quede como el siguiente:

```
# Solicita el número al usuario  
numero = input("Introduce un número que multiplicaré por 8")  
print(type(numero))  
# Muestra el número multiplicado por 8  
print(f"El resultado {numero * 8}!")
```

Observa que el tipo de numero es str, una cadena de caracteres, y al multiplicar una cadena de caracteres por un número lo que ocurre es que se muestra la cadena tantas veces como indica el número.

Modifica el programa para que quede como el siguiente:

```
# Solicita el número al usuario y cambia el tipo a entero
numero = int(input("Introduce un número que multiplicaré por 8"))

# Muestra el número multiplicado por 8
print(f"¡El resultado {numero * 8}!")
```

Ahora todo funciona. Bueno, todo funciona menos si al introducir el número el usuario introduce algo parecido a J.

Aquí python protesta, ya que sabemos que el usuario puede introducir lo que le dé la gana y nosotros no hemos tenido en cuenta esta situación excepcional. Y es así como se soluciona, controlando excepciones, pero de momento se nos queda un poco lejano.

### **Reto 1:**

Realiza un programa que pida al usuario dos valores y los guarde en dos variables (a y b). El programa tiene que intercambiar los valores: a tendrá el valor de b y b el valor de a

### **Reto 2:**

Tienes a tu disposición los siguientes programas:

**primer\_pygame.py**  
**un\_rectangulo\_con\_eventos.py**  
**un\_rectangulo\_con\_teclas.py**  
**un\_rectangulo\_con\_gravedad.py**  
**dos\_rectangulos.py**

Descarga y abre en VisualStudio **primer\_pygame.py**. Ejecútalo e intenta, a continuación, traducir cada una de las líneas de ese código desde la línea **#Bucle principal**.

Descarga y ejecuta **un\_rectangulo\_con\_eventos.py** y ejecútalo.

Haz lo mismo con **un\_rectangulo\_con\_teclas.py** y con **un\_rectangulo\_con\_gravedad.py**

Termina con **dos\_rectangulos.py**

No te preocunes si no entiendes alguna cosa. Después de realizar los retos siguientes probablemente todo te parezca más claro.

### **Reto 3**

En “primer\_pygame.py”, programa que únicamente muestra una ventana en blanco con un cuadrado rojo solo hay una variable llamada ventana.

Cambia (255,255,255) y (255,0,0) por dos variables que vamos a llamar BLANCO y ROJO

Cambia también la tupla que aparece en pygame.draw.rect (50,50,50,50) para que esos 50 estén almacenados en las variables rect\_x, rect\_y, rect\_ancho y rect\_alto.

El texto de la ventana: “cuadro rojo sobre fondo blanco” debe almacenarse en la variable TITULO.

Una vez hechos los cambios, modifica el valor de las variables y observa que ocurre.

### **Reto 4:**

En “primer\_pygame.py” y en la línea pygame.draw.rect (rect\_x, rect\_y, rect\_ancho,rect\_alto), cambia estos valores y describe que está pasando. Realiza una tabla como la siguiente:

#### **Parámetro cambiado Aumento o decremento Consecuencia**

rect_x	+10
rect_x	-10
rect_y	+10
rect_y	-10
rect_ancho	+10
rect_ancho	-10
rect_alto	+10
rect_alto	-10

### **Reto 5:**

El valor (600, 400) que aparece en este programa ¿Pueden cambiarse esos valores por una variable? ¿Cómo?

### **Reto 6:**

¿Qué tendrías que hacer para que el rectángulo apareciera siempre en medio de la pantalla aunque cambiaren los valores de ANCHO y ALTO del ejercicio anterior?

### **Reto 7:**

Crea un segundo programa de pygame en el que aparezcan dos cuadros, uno blanco y otro verde, el primero en mitad de la pantalla (teniendo en cuenta las variables ANCHO Y ALTO) y el segundo 20 píxeles a la izquierda el primero.

---

Para terminar la clase pondremos en común lo que hemos aprendido visualizando el código de todos los programas