

CLASE 4

Operadores y String Methods

Curso de Python - Fundamentos

Material de Estudio y Referencia

CONTENIDO

1. Operadores Aritmeticos
2. Operadores de Comparacion
3. Operadores Logicos
4. String Methods - Transformacion
5. String Methods - Limpieza
6. String Methods - Busqueda y Reemplazo
7. String Methods - Separacion y Union
8. Tabla de Referencia Rapida
9. Ejercicios Resueltos
10. Errores Comunes y Soluciones

1. OPERADORES ARITMETICOS

Los operadores aritmeticos realizan operaciones matematicas entre valores numericos.

Operador	Nombre	Ejemplo	Resultado
+	Suma	10 + 3	13
-	Resta	10 - 3	7
*	Multiplicacion	10 * 3	30
/	Division	10 / 3	3.333...
//	Division entera	10 // 3	3
%	Modulo (residuo)	10 % 3	1
**	Potencia	10 ** 3	1000

Ejemplo de codigo:

```
a = 10
b = 3
print(f"Suma: {a + b}")           # 13
print(f"Division: {a / b}")       # 3.333...
print(f"Division entera: {a // b}")# 3
print(f"Residuo: {a % b}")        # 1
print(f"Potencia: {a ** b}")      # 1000
```

Orden de operaciones (PEMDAS):

1. Parentesis () - 2. Exponentes ** - 3. Multiplicacion * - 4. Division / // % - 5. Adicion + - 6. Sustraccion -

```
resultado = 2 + 3 * 4      # = 14 (3*4 primero, luego +2)
resultado = (2 + 3) * 4     # = 20 (parentesis primero)
```

2. OPERADORES DE COMPARACION

Los operadores de comparacion comparan dos valores y devuelven True o False.

Operador	Significado	Ejemplo	Resultado
==	Igual a	5 == 5	True
!=	Diferente de	5 != 3	True
>	Mayor que	5 > 3	True
<	Menor que	5 < 3	False
>=	Mayor o igual	5 >= 5	True
<=	Menor o igual	5 <= 3	False

IMPORTANTE: Diferencia entre = y ==

```
x = 5      # Asignacion: guarda el valor 5 en x
x == 5     # Comparacion: pregunta si x es igual a 5 (True)
```

3. OPERADORES LOGICOS

Los operadores logicos combinan condiciones booleanas.

Operador	Descripcion	Ejemplo
and	True si AMBAS son True	True and True = True
or	True si AL MENOS UNA es True	True or False = True
not	Invierte el valor	not True = False

Tablas de Verdad:

AND - Ambas deben ser True:

A	B	A and B
True	True	True
True	False	False
False	True	False
False	False	False

OR - Al menos una debe ser True:

A	B	A or B
True	True	True
True	False	True
False	True	True
False	False	False

4. STRING METHODS - TRANSFORMACION

Metodo	Descripcion	Ejemplo	Resultado
.upper()	Todo a mayusculas	"hola".upper()	"HOLA"
.lower()	Todo a minusculas	"HOLA".lower()	"hola"
.title()	Capitaliza cada palabra	"hola mundo".title()	"Hola Mundo"
.capitalize()	Solo primera letra	"hola mundo".capitalize()	"Hola mundo"

5. STRING METHODS - LIMPIEZA

Metodo	Descripcion	Ejemplo	Resultado
.strip()	Elimina espacios inicio/fin	" hola ".strip()	"hola"
.lstrip()	Elimina espacios inicio	" hola ".lstrip()	"hola "
.rstrip()	Elimina espacios fin	" hola ".rstrip()	" hola"

6. STRING METHODS - BUSQUEDA Y REEMPLAZO

Metodo	Descripcion	Ejemplo	Resultado
.find(x)	Posicion de x (-1 si no existe)	"hola".find("o")	1
.count(x)	Cuantas veces aparece x	"abracadabra".count("a")	5
.replace(v,n)	Reemplaza v por n	"hola".replace("o","0")	"h0la"
.startswith(x)	Empieza con x?	"hola".startswith("ho")	True
.endswith(x)	Termina con x?	"hola".endswith(".py")	False

7. STRING METHODS - SEPARACION Y UNION

Metodo	Descripcion	Ejemplo	Resultado
.split()	Divide por espacios	"a b c".split()	["a" , "b" , "c"]
.split(x)	Divide por x	"a-b-c".split("-")	["a" , "b" , "c"]
x.join(lista)	Une lista con x	"-".join(["a" , "b"])	"a-b"

Method Chaining (Encadenar metodos):

Puedes aplicar varios metodos seguidos porque cada uno devuelve un nuevo string:

```
texto = "    HOLA MUNDO    "
resultado = texto.strip().lower().replace(" ", "-")
print(resultado) # "hola-mundo"
```

8. TABLA DE REFERENCIA RAPIDA

Operadores:

ARITMETICOS: + - * / // % **
COMPARACION: == != > < >= <=

LOGICOS: and or not

String Methods:

TRANSFORMAR: .upper() .lower() .title() .capitalize()
LIMPIAR: .strip() .lstrip() .rstrip()
BUSCAR: .find() .count() .replace()
VERIFICAR: .startswith() .endswith()
SEPARAR: .split() .join()
LONGITUD: len(texto) # Es funcion, no metodo

9. EJERCICIOS RESUELTOS

Ejercicio 1: Calculadora

```
num1 = float(input("Numero 1: "))
num2 = float(input("Numero 2: "))

print(f"Suma: {num1 + num2}")
print(f"Resta: {num1 - num2}")
print(f"Multiplicacion: {num1 * num2}")
print(f"Division: {num1 / num2}")
print(f"Potencia: {num1 ** num2}")
```

Ejercicio 2: Validador de Password

```
password = input("Contrasena: ")

longitud_ok = len(password) >= 8
no_prohibida = password != "password" and password != "12345678"

es_valida = longitud_ok and no_prohibida
print(f"Contrasena valida: {es_valida}")
```

Ejercicio 3: Generador de Username

```
nombre = input("Nombre completo: ")
username = nombre.strip().lower().replace(" ", "_")
print(f"Tu username: {username}")
```

Ejercicio 4: Contador de Palabras

```
texto = input("Escribe una oracion: ")

palabras = texto.split()
num_palabras = len(palabras)
num_caracteres = len(texto)

print(f"Palabras: {num_palabras}")
print(f"Caracteres: {num_caracteres}")
```

10. ERRORES COMUNES Y SOLUCIONES

Error 1: Usar = en lugar de ==

```
# INCORRECTO
if x = 5: # Error de sintaxis!

# CORRECTO
if x == 5: # Comparacion
```

Error 2: No guardar resultado de string method

```
# INCORRECTO
nombre = "juan"
nombre.upper() # No se guarda!
print(nombre) # Sigue siendo "juan"

# CORRECTO
nombre = "juan"
nombre = nombre.upper() # Guardar el resultado
print(nombre) # "JUAN"
```

Error 3: Olvidar convertir input

```
# INCORRECTO
num = input("Numero: ")
print(num + 5) # TypeError! num es string

# CORRECTO
num = int(input("Numero: "))
print(num + 5) # Funciona
```

Error 4: Division por cero

```
# INCORRECTO
resultado = 10 / 0 # ZeroDivisionError!

# CORRECTO: verificar antes
if divisor != 0:
    resultado = 10 / divisor
```

Recuerda: La practica hace al maestro. Escribe codigo todos los dias!