

Módulo 2.2: Operaciones con datos y variables

Prof. Carlos Cedeño

En los módulos anteriores, hablamos sobre las variables, como asignar valores a las variables y los distintos tipos de datos primitivos que tenemos en Python. En esta clase, profundizaremos sobre las operaciones que podemos realizar con estos datos.

1. Conversión entre Tipos de Datos (Casting)

En Python, a veces necesitamos convertir un valor de un tipo de dato a otro. Esto se conoce como "casting" o "conversión de tipos".

Funciones comunes de casting:

- `int()` : Convierte a un número entero.
- `float()` : Convierte a un número de punto flotante.
- `str()` : Convierte a una cadena de texto.
- `bool()` : Convierte a un valor booleano.

Ejemplos:

```
# De string a entero
numero_texto = "100"
numero_entero = int(numero_texto)
print(f"String a entero: {numero_entero}, tipo: {type(numero_entero)}")

# De entero a string
edad = 25
edad_texto = str(edad)
print(f"Entero a string: '{edad_texto}', tipo: {type(edad_texto)}")
```

```
# De flotante a entero (se trunca la parte decimal)
precio_decimal = 19.99
precio_entero = int(precio_decimal)
print(f"Flotante a entero: {precio_entero}, tipo: {type(precio_entero)}")

# De entero a flotante
cantidad = 5
cantidad_flotante = float(cantidad)
print(f"Entero a flotante: {cantidad_flotante}, tipo: {type(cantidad_flotante)}")

# A booleano (0, string vacío "", lista vacía [], etc., se convierten a False)
valor1 = 0
booleano1 = bool(valor1)
print(f"'{valor1}' a booleano: {booleano1}")

valor2 = "Hola"
booleano2 = bool(valor2)
print(f"'{valor2}' a booleano: {booleano2}")
```

2. Operaciones Numéricas

Python soporta las operaciones aritméticas básicas con números (enteros y flotantes).

Operadores Aritméticos:

- `+` : Suma
- `-` : Resta
- `*` : Multiplicación
- `/` : División (resultado flotante)
- `//` : División entera (resultado entero, trunca decimales)
- `%` : Módulo (resto de la división)
- `**` : Exponenciación (potencia)

Ejemplos:

```
a = 10
```

```
b = 3
```

```
suma = a + b
```

```
resta = a - b
```

```
multiplicacion = a * b
```

```
division = a / b
```

```
division_entera = a // b
```

```
modulo = a % b
```

```
potencia = a ** b
```

```
print(f"{a} + {b} = {suma}")
```

```
print(f"{a} - {b} = {resta}")
```

```
print(f"{a} * {b} = {multiplicacion}")
```

```
print(f"{a} / {b} = {division}")
```

```
print(f"{a} // {b} = {division_entera}")
```

```
print(f"{a} % {b} = {modulo}")
```

```
print(f"{a} ** {b} = {potencia}")
```

```
# Orden de operaciones (PEMDAS Significa Paréntesis, Exponentes,  
# Multiplicación y División (de izquierda a derecha),  
# Suma y Resta (de izquierda a derecha)  
  
resultado_complejo = (10 + 5) * 2 - 3**2 / 2  
print(f"Resultado complejo: {resultado_complejo}")
```

3. Operaciones Booleanas

Los booleanos representan dos valores: `True` (verdadero) o `False` (falso). Son fundamentales para la lógica condicional.

Operadores Lógicos:

- `and` : Devuelve `True` si ambos operandos son `True` .
- `or` : Devuelve `True` si al menos uno de los operandos es `True` .
- `not` : Invierte el valor booleano (si es `True` lo convierte a `False` y viceversa).

Operadores de Comparación (devuelven booleanos):

- `==` : Igual a
- `!=` : No igual a (diferente de)
- `>` : Mayor que
- `<` : Menor que
- `>=` : Mayor o igual que
- `<=` : Menor o igual que

Ejemplos:

```
# Operadores Lógicos
```

```
verdadero = True
```

```
falso = False
```

```
print(f"{verdadero} and {falso} = {verdadero and falso}") # False
```

```
print(f"{verdadero} or {falso} = {verdadero or falso}") # True
```

```
print(f"not {verdadero} = {not verdadero}") # False
```

Operadores de Comparación

```
x = 5  
y = 10  
z = 5
```

```
print(f"{x} == {y} = {x == y}") # False  
print(f"{x} != {y} = {x != y}") # True  
print(f"{x} > {y} = {x > y}")   # False  
print(f"{x} < {y} = {x < y}")   # True  
print(f"{x} >= {z} = {x >= z}") # True  
print(f"{x} <= {z} = {x <= z}") # True
```

Combinando operadores

```
condicion1 = x < y and y > z # True and True -> True
condicion2 = x > y or y == z # False or False -> False
print(f"({x} < {y}) and ({y} > {z}) = {condicion1}")
print(f"({x} > {y}) or ({y} == {z}) = {condicion2}")
```

4. Operaciones entre Strings

Las cadenas de texto (strings) tienen operaciones específicas.

Operaciones Básicas:

- **Concatenación** (`+`): Une dos o más strings.
- **Repetición** (`*`): Repite un string un número específico de veces.
- **Indexación**: Accede a un carácter específico del string usando su índice (comienza en 0).
- **Pertenencia** (`in` , `not in`): Verifica si un substring está presente en un string.
- **Longitud** (`len()`): Obtiene el número de caracteres en un string.

Ejemplos:

```
saludo = "Hola"
nombre = "Mundo"

# Concatenación
mensaje_completo = saludo + " " + nombre + "!"
print(f"Concatenación: {mensaje_completo}")

# Repetición
risa = "ja" * 5
print(f"Repetición: {risa}")

# Indexación
texto = "Python"
print(f"Primer carácter de '{texto}': {texto[0]}") # P
print(f"Último carácter de '{texto}': {texto[1]}") # y

# Longitud
print(f"Longitud de '{mensaje_completo}': {len(mensaje_completo)}")
```

5. Manipulaciones de Strings

Python ofrece métodos integrados para modificar y trabajar con strings. Estos métodos devuelven un *nuevo* string modificado, no alteran el original.

Métodos Básicos:

- `upper()` : Convierte todo el string a mayúsculas.
- `lower()` : Convierte todo el string a minúsculas.
- `capitalize()` : Pone en mayúscula el primer carácter del string.
- `title()` : Pone en mayúscula el primer carácter de cada palabra.
- `strip()` : Elimina espacios en blanco (o caracteres especificados) al principio y al final.

- `replace(viejo, nuevo)` : Reemplaza todas las ocurrencias de un substring por otro.
- `split(separador)` : Divide el string en una lista de substrings usando un separador. Si no se especifica, divide por espacios en blanco.
- `find(substring)` : Devuelve el índice de la primera ocurrencia del substring (o -1 si no se encuentra).
- `count(substring)` : Cuenta cuántas veces aparece un substring.

Ejemplos:

```
texto_original = "    hola mundo, bienvenidos a python.    "
print(f"Original: '{texto_original}'")

print(f"Mayúsculas: '{texto_original.upper()}'")
print(f"Minúsculas: '{texto_original.lower()}'")
print(f"Capitalizado: '{texto_original.strip().capitalize()}'")
print(f"Título: '{texto_original.strip().title()}'")

# Borrar (eliminar espacios)
texto_limpio = texto_original.strip()
print(f"Sin espacios extra: '{texto_limpio}'")
texto_con_guiones = "---Hola---".strip('-')
print(f"Sin guiones extra: '{texto_con_guiones}'")
```

```
# Reemplazar
texto_reemplazado = texto_limpio.replace("mundo", "Pythonistas")
print(f"Reemplazado: '{texto_reemplazado}'")

# Dividir (Split)
palabras = texto_limpio.split(" ")
print(f"Dividido por espacios: {palabras}")
data = "nombre,edad,ciudad"
campos = data.split(",")
print(f"Dividido por comas: {campos}")

# Búsqueda y Conteo
email = "usuario@ejemplo.com"
print(f"Posición de '@' en '{email}': {email.find('@')}") # 7
print(f"Cuántas 'o' hay en 'hola mundo': {'hola mundo'.count('o')}") # 2
```