



Introducción a la Programación Segura

Estructuras de decisión, bucles y funciones del lenguaje Python.



INTRODUCCIÓN A LA PROGRAMACIÓN SEGURA

UNIDAD 2: ESTRUCTURAS DE DECISIÓN, BUCLES Y FUNCIONES DEL LENGUAJE PYTHON.

Contenidos

1. Variables.
2. Literales (enteros, flotantes, cadenas, valores booleanos).
3. La función print() y atributos de formato en la salida.
4. Operadores y expresiones.



UNIDAD DE APRENDIZAJE

UNIDAD N°2: ESTRUCTURAS DE DECISIÓN, BUCLES Y FUNCIONES DEL LENGUAJE PYTHON.

Aprendizaje Esperado:

2.1. Desarrolla scripts de Python, para entregar solución a problemas de mediana complejidad, considerando los estándares internacionales de programación y la evaluación de los resultados obtenidos en base a la evidencia.

Operadores y expresiones:

En Python, los operadores son **símbolos especiales** que se utilizan para realizar **operaciones** en variables y valores. **Las expresiones** son combinaciones de operadores y operandos que, cuando se evalúan, producen un resultado.

Operadores aritméticos:

Los operadores aritméticos **realizan operaciones matemáticas** entre operandos numéricos.

Suma:

```
resultado = 540 + 300 840
```

Resta:

```
resultado = 540 - 300 240
```

Multiplicación:

```
resultado = 540 * 300 162000
```

División:

```
resultado = 540 / 300 1.8
```


Operadores y expresiones:

Operadores aritméticos:

Los operadores aritméticos **realizan operaciones matemáticas** entre operandos numéricos.

Potencia:

```
resultado = 3 ** 5
```

243

División entera:

```
resultado = 11 // 3
```

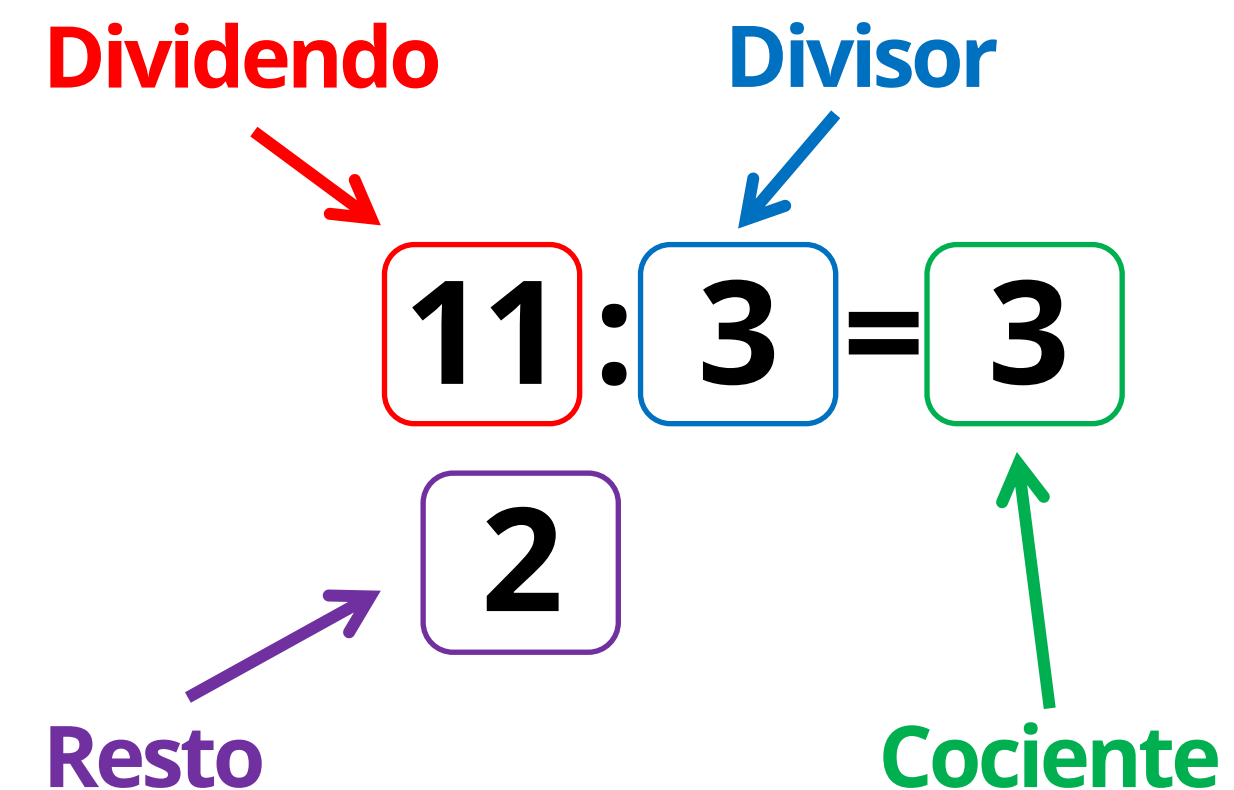
3

Módulo:

```
resultado = 11 % 3
```

2

División:



La División entera nos entrega el Cociente y el Módulo nos entrega el Resto de la división

Prioridades de los Operadores:

Los operadores tienen diferentes prioridades que determinan el orden en que se evalúan en una expresión.

Prioridades ordenadas de mayor a menor:

1. Potencia (**)
2. Multiplicación (*), División (/), División entera (//), Módulo (%)
3. Suma (+), Resta (-)

```
resultado = 3 ** 4 * 4 + 10 / 2 - 5
```

```
324.0
```

- | | |
|--------------------|---|
| 1. Potencia: | $3 ** 4 = 81$ |
| 2. Multiplicación: | $81 * 4 = 324$ |
| 3. División: | $10 / 2 = 5$ (resultado como número entero) |
| 4. Suma: | $324 + 5 = 329$ |
| 5. Resta: | $329 - 5 = 324$ |

Operadores y sus enlaces:

En Python, los operadores **se evalúan de izquierda a derecha**. Esto significa que cuando hay múltiples operadores del mismo nivel de prioridad en una expresión, se evalúan de izquierda a derecha.

```
resultado = 10 - 2 * 3 ** 2 + 60 / 2
```

```
22.0
```

1. La potencia tiene el mayor enlace, por lo que se evalúa primero:
2. Luego, la multiplicación tiene el siguiente enlace, por lo que se realiza la operación:
3. La división tiene el mismo enlace, pero esta hacia la derecha, por lo que no está primero:
4. La suma y resta tienen el mismo enlace, pero los enlaces de izquierda a derecha se evalúan secuencialmente:

$$3 ** 2 = 9$$

$$2 * 9 = 18$$

$$60 - 2 = 30$$

Se realiza la resta: $10 - 2 = -8$

Se realiza la suma: $-8 + 30 = 22$

Operadores de acceso directo:

También conocidos como **operadores de asignación aumentada** o shorthand operators en inglés, son una forma concisa de **realizar operaciones aritméticas y asignaciones en una sola expresión**. Estos operadores **combinan una operación aritmética con una asignación** y se utilizan para modificar el valor de una variable de manera más eficiente y legible.

Operador de suma y asignación (+=):

Realiza una **suma entre el valor actual** de la variable **y el valor especificado** y luego **asigna el resultado a la variable**.

```
x = 10
x += 5 # Esto es lo mismo que x = x + 5
# Con lo anterior x es igual a 15
```

Operador de resta y asignación (-=):

Realiza una **resta entre el valor actual** de la variable **y el valor especificado** y luego **asigna el resultado a la variable**.

```
x = 10
x -= 5 # Esto es lo mismo que x = x - 5
# Con lo anterior x es igual a 5
```


Operadores de acceso directo:

Operador de multiplicación y asignación (`*=`):

Realiza una **multiplicación** entre el **valor actual** de la variable **y el valor especificado** y luego **asigna el resultado a la variable**.

```
x = 10
x *= 5 # Esto es lo mismo que x = x * 5
# Con lo anterior x es igual a 50
```

Operador de división y asignación (`/=`):

Realiza una **división** entre el **valor actual** de la variable **y el valor especificado** y luego **asigna el resultado a la variable**.

```
x = 11
x /= 5 # Esto es lo mismo que x = x / 5
# Con lo anterior x es igual a 2.2
```

Operador de división entera y asignación (`//=`):

Realiza una **división entera** entre el **valor actual** de la variable **y el valor especificado** y luego **asigna el resultado a la variable**.

```
x = 11
x //= 5 # Esto es lo mismo que x = x // 5
# Con lo anterior x es igual a 2
```

Operador de módulo y asignación (`%=`):

Realiza una **operación de módulo** entre el **valor actual** de la variable **y el valor especificado** y luego **asigna el resultado a la variable**.

```
x = 11
x %= 5 # Esto es lo mismo que x = x % 5
# Con lo anterior x es igual a 1
```

Operadores de acceso directo:

Operador de potencia y asignación (`**=`):

Realiza una **operación de potencia** entre el **valor actual** de la variable y el **valor especificado** y luego **asigna el resultado** a la variable.

```
x = 2
x **= 8 # Esto es lo mismo que x = x ** 8
# Con lo anterior x es igual a 256
```

Conversión de cadenas en números:

Para **convertir cadenas** (strings) **en números** utilizamos las funciones `int()` y `float()`. Estas funciones **permiten tomar una cadena que contiene un número y convertirla en un tipo de dato numérico**, ya sea un entero (`int`) o un número de punto flotante (`float`).

Conversión a entero (`int()`):

La función `int()` **se utiliza para convertir una cadena en un entero**. Si la cadena contiene un número entero válido, esta devolverá el valor correspondiente como entero, pero si la cadena no representa un número entero válido, generará un error.

```
cadena_edad = "27"  
entero_edad = int(cadena_edad)  
print(entero_edad)
```



27

```
cadena_edad = "hola"  
entero_edad = int(cadena_edad)  
print(entero_edad)
```



```
ValueError: invalid literal for int()  
with base 10: 'hola'
```

Conversión de cadenas en números:

Para **convertir cadenas** (strings) **en números** utilizamos las funciones `int()` y `float()`. Estas funciones **permiten tomar una cadena que contiene un número y convertirla en un tipo de dato numérico**, ya sea un entero (`int`) o un número de punto flotante (`float`).

Conversión a punto flotante (`float()`):

La función `float()` se utiliza para convertir una cadena en un número de punto flotante. Si la cadena contiene un número válido con decimales, esta devolverá el valor correspondiente como número de punto flotante, pero si la cadena no representa un número válido, generará un error.

```
cadena_peso = "78.450"  
flotante_peso = float(cadena_peso)  
print(flotante_peso)
```



78.45

```
cadena_peso = "hola"  
flotante_peso = float(cadena_peso)  
print(flotante_peso)
```



ValueError: could not convert string
to float: 'hola'

Operadores de cadenas:

Los operadores de cadena **son utilizados para realizar operaciones específicas con cadenas de texto**. Estos operadores te permiten concatenar, repetir, buscar y comparar cadenas, entre otras cosas.

Operador de concatenación(+):

Este operador se utiliza para **unir dos cadenas en una sola** cadena más larga.

```
nombre = "Teresa"  
apellido = "Rojas"  
nombre_completo = nombre + " " + apellido  
print(nombre_completo)
```

Teresa Rojas

Operador de repetición(*):

Este operador se utiliza para **repetir una cadena un cierto número de veces**.

```
nombre = "Teresa"  
resultado = nombre * 3  
print(resultado)
```

TeresaTeresaTeresa

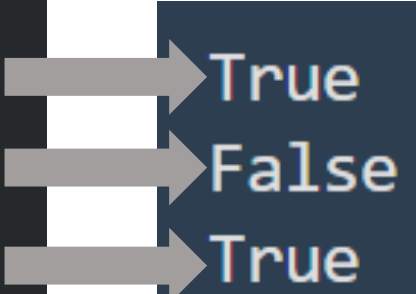
Operadores de cadenas:

Los operadores de cadena **son utilizados para realizar operaciones específicas con cadenas de texto**. Estos operadores te permiten concatenar, repetir, buscar y comparar cadenas, entre otras cosas.

Operador de pertenencia (in):

Este operador se utiliza para **verificar si una subcadena está presente en una cadena** más grande.

```
nombre_completo = "Teresa Rojas"
print("Teresa" in nombre_completo)
print("Araya" in nombre_completo)
print("Rojas" in nombre_completo)
```



True
False
True

Operadores de cadenas:

Los operadores de cadena **son utilizados para realizar operaciones específicas con cadenas de texto**. Estos operadores te permiten concatenar, repetir, buscar y comparar cadenas, entre otras cosas.

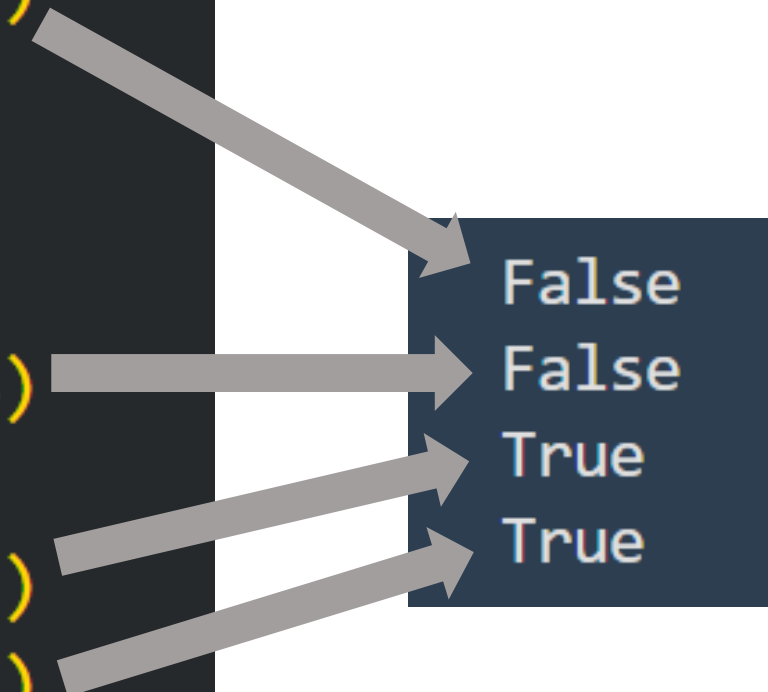
Operadores de comparación (`==`, `!=`, `<`, `>`, `<=`, `>=`):

Estos operadores se utilizan para **comparar dos cadenas y devuelven un valor booleano** (True o False) según el resultado de la comparación.

```
ciudad1 = "La Serena"  
ciudad2 = "Coquimbo"  
print(ciudad1 == ciudad2)
```

```
ciudad3 = "Osorno"  
ciudad4 = "La Serena"  
print(ciudad3 == ciudad4)
```

```
print(ciudad1 != ciudad2)  
print(ciudad3 != ciudad4)
```



False
False
True
True

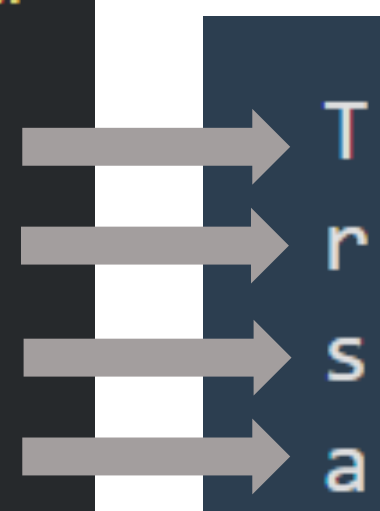
Operadores de cadenas:

Los operadores de cadena **son utilizados para realizar operaciones específicas con cadenas de texto**. Estos operadores te permiten concatenar, repetir, buscar y comparar cadenas, entre otras cosas.

Operador de indexación ([]):

Este operador se utiliza para **acceder a caracteres individuales en una cadena utilizando su índice**. Los índices en Python comienzan en 0.

```
nombre = "Teresa"
print(nombre[0])
print(nombre[2])
print(nombre[4])
print(nombre[5])
```



Índice	Carácter
0	T
1	e
2	r
3	e
4	s
5	a

Conversión de números en cadenas:

En Python, puedes **convertir números en cadenas utilizando la función `str()`**. Esta función toma cualquier valor numérico y lo convierte en una cadena de texto. Es útil **cuando necesitas combinar números con otras cadenas** o mostrarlos en la consola **utilizando la función `print()`**.

Conversión de un entero a cadena:

```
entero_edad = 42  
cadena_edad = str(entero_edad)  
print("Tienes ", cadena_edad, " años.")
```

Tienes 42 años.

Conversión de un número de punto flotante a cadena:

```
flotante_peso = 75.8  
cadena_peso = str(floteante_peso)  
print("Tu peso es ", cadena_peso, " kilos.")
```

Tu peso es 75.8 kilos.

Conversión de números en cadenas:

En Python, puedes **convertir números en cadenas utilizando la función `str()`**. Esta función toma cualquier valor numérico y lo convierte en una cadena de texto. Es útil **cuando necesitas combinar números con otras cadenas** o mostrarlos en la consola **utilizando la función `print()`**.

Conversión de un resultado de cálculo a cadena:

```
radio = 5
area = 3.14 * (radio ** 2)
cadena_area = str(area)
print("El área del círculo es: " + cadena_area)
```

```
El área del círculo es: 78.5
```

Concatenación de números y cadenas:

En Python, puedes **combinar números con otras cadenas utilizando** el operador de concatenación (+) sin la necesidad de convertir explícitamente los números en cadenas.

Concatenación de números y cadenas:

```
nombre = "Teresa"  
edad = 27  
saludo = "Hola, mi nombre es " + nombre + " y tengo " + str(edad) + " años."  
print(saludo)
```

```
Hola, mi nombre es Teresa y tengo 27 años.
```

Programas interactivos simples:

1. Suma de dos números enteros:

```
numero1 = int(input("Ingrese el primer número entero: "))  
numero2 = int(input("Ingrese el segundo número entero: "))  
suma = numero1 + numero2  
print("La suma es:", suma)
```

```
Ingrese el primer número entero: 12  
Ingrese el segundo número entero: 26  
La suma es: 38
```

En este ejemplo estamos convirtiendo a entero lo que el usuario escribe a través del teclado gracias a la función `input()`.

Una vez convertidos a entero (los datos ingresados por el usuario) realizamos la suma y mostramos los resultados por pantalla.

Programas interactivos simples:

2. Cálculo del área de un rectángulo:

```
base = float(input("Ingrese la longitud de la base del rectángulo: "))
altura = float(input("Ingrese la altura del rectángulo: "))
area = base * altura
print("El área del rectángulo es:", area)
```

```
Ingrese la longitud de la base del rectángulo: 25
Ingrese la altura del rectángulo: 15
El área del rectángulo es: 375.0
```

En este ejemplo estamos convirtiendo a punto flotante lo que el usuario escribe a través del teclado gracias a la función `input()`.

Una vez convertidos a punto flotante (los datos ingresados por el usuario) realizamos el cálculo de área y mostramos los resultados por pantalla.

Programas interactivos simples:

3. Conversión de grados Celsius a Fahrenheit:

```
celsius = float(input("Ingrese la temperatura en grados Celsius: "))  
fahrenheit = (celsius * 9/5) + 32  
print("La temperatura en grados Fahrenheit es:", fahrenheit)
```

```
Ingrese la temperatura en grados Celsius: 19  
La temperatura en grados Fahrenheit es: 66.2
```



A practicar...

Ejercicios:

De acuerdo al contenido visto en clases, desarrolla scripts en lenguaje Python para los siguientes ejercicios:

1. Calcular la suma de 4 números con decimales ingresados por el usuario.
2. Cálculo del área de un rectángulo solicitando los datos al usuario.
3. Conversor de monedas (Dólar a CLP) solicitando el valor actual del dólar y la cantidad de dolares al usuario.
4. Conversor de longitud (Pies a Metros) solicitando los datos al usuario.
5. Generador de correo electrónico con formato nombre.apellido@correo.cl, solicitando los datos al usuario.
6. Solicitar al usuario un número y un porcentaje, para mostrar por pantalla el número correspondiente al porcentaje del número ingresado.
7. Solicitar al usuario el precio de un producto y el porcentaje de descuento, para mostrar por pantalla el precio a pagar después de aplicar el descuento



Ejercicios:

De acuerdo al contenido visto en clases, desarrolla scripts en lenguaje Python para los siguientes ejercicios:

1. Calcular la suma de 4 números con decimales ingresados por el usuario:

```
numero1 = float(input("Ingrese el primer número: "))
numero2 = float(input("Ingrese el segundo número: "))
numero3 = float(input("Ingrese el tercer número: "))
numero4 = float(input("Ingrese el cuarto número: "))

suma = numero1 + numero2 + numero3 + numero4
print("La suma es:", suma)
```

```
Ingrese el primer número: 25.2
Ingrese el segundo número: 17.9
Ingrese el tercer número: 75.4
Ingrese el cuarto número: 15.6
La suma es: 134.1
```



Ejercicios:

De acuerdo al contenido visto en clases, desarrolla scripts en lenguaje Python para los siguientes ejercicios:

2. Cálculo del área de un rectángulo solicitando los datos al usuario:

```
base = float(input("Ingrese la longitud de la base del rectángulo: "))
altura = float(input("Ingrese la altura del rectángulo: "))
area = base * altura
print("El área del rectángulo es:", area)
```

```
Ingrese la longitud de la base del rectángulo: 25
Ingrese la altura del rectángulo: 15
El área del rectángulo es: 375.0
```



Ejercicios:

De acuerdo al contenido visto en clases, desarrolla scripts en lenguaje Python para los siguientes ejercicios:

3. Conversor de monedas (Dólar a CLP) solicitando el valor actual del dólar y la cantidad de dolares al usuario:

```
valor_dolar = float(input("Ingrese el valor actual del dolar: "))  
cantidad_dolares = float(input("Ingrese la cantidad de dólares: "))  
cantidad_clp = int(cantidad_dolares * valor_dolar)  
print("Equivalente en CLP: $", cantidad_clp)
```

```
Ingrese el valor actual del dolar: 849.23  
Ingrese la cantidad de dólares: 5  
Equivalente en CLP: $ 4246
```



Ejercicios:

De acuerdo al contenido visto en clases, desarrolla scripts en lenguaje Python para los siguientes ejercicios:

4. Conversor de longitud (Pies a Metros) solicitando los datos al usuario:

```
pies = float(input("Ingrese la longitud en pies: "))  
metros = pies * 0.3048  
print("Equivalente en metros: ", metros)
```

```
Ingrese la longitud en pies: 100  
Equivalente en metros: 30.48
```



Ejercicios:

De acuerdo al contenido visto en clases, desarrolla scripts en lenguaje Python para los siguientes ejercicios:

5. Generador de correo electrónico con formato nombre.apellido@correo.cl, solicitando los datos al usuario:

```
nombre = input("Ingrese su nombre: ")
apellido = input("Ingrese su apellido: ")
correo = nombre.lower() + "." + apellido.lower() + "@correo.cl"
print("Su correo electrónico es:", correo)
```

```
Ingrese su nombre: Teresa
Ingrese su apellido: Rojas
Su correo electrónico es: teresa.rojas@correo.cl
```



La función **lower()** convierte a **minúsculas** una cadena y **upper()** convierte a **mayúsculas** una cadena.

Ejercicios:

De acuerdo al contenido visto en clases, desarrolla scripts en lenguaje Python para los siguientes ejercicios:

6. Solicitar al usuario un número y un porcentaje, para mostrar por pantalla el número correspondiente al porcentaje del número ingresado:

```
numero = float(input("Ingrese el total: "))  
porcentaje = float(input("Ingrese el porcentaje a calcular: "))  
resultado = numero * (porcentaje / 100)  
print("El", porcentaje, "% de", numero, "es:", resultado)
```

```
Ingrese el total: 850  
Ingrese el porcentaje a calcular: 10  
El 10.0 % de 850.0 es: 85.0
```



Ejercicios:

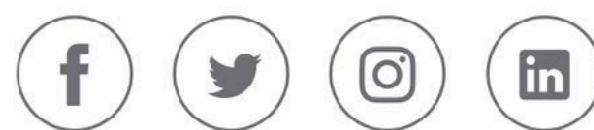
De acuerdo al contenido visto en clases, desarrolla scripts en lenguaje Python para los siguientes ejercicios:

7. Solicitar al usuario el precio de un producto y el porcentaje de descuento, para mostrar por pantalla el precio a pagar después de aplicar el descuento:

```
precio = float(input("Ingrese el precio original del producto: "))
descuento = float(input("Ingrese el porcentaje de descuento: "))
precio_descuento = precio - (precio * (descuento / 100))
print("El precio con descuento es:", precio_descuento)
```

```
Ingrese el precio original del producto: 3500
Ingrese el porcentaje de descuento: 10
El precio con descuento es: 3150.0
```





inacap.cl