

# Variables

Conceptualización, representación,  
codificación y operaciones básicas  
Del lenguaje





# Contenido

- Variables de Python
  - Declaración de variable y asignación de valores
  - Errores al declarar una variable
  - Asignación múltiple
  - Asignación de un solo valor a varias variables
  - Asignar varios valores a varias variables



# Contenido

- Tipos de datos integrados
- Manipular números
  - Operadores Aritméticos
- Orden en las operaciones
  - Paréntesis (P)
  - Exponenciación (E)



# Contenido

- Multiplicación (M), División (D), Suma (A), Resta (S)
- Misma precedencia
- Ejercicios



# Contenido

- Multiplicación (M), División (D), Suma (A), Resta (S)
- Misma precedencia
- Ejercicios





# Contenido

## Variables de Python

- Variable es un nombre que se usa para referirse a la ubicación de la memoria. La variable de Python también se conoce como un identificador y se usa para mantener el valor.
- En Python, no necesitamos especificar el tipo de variable porque Python es un lenguaje de inferir y lo suficientemente inteligente como para obtener el tipo de variable.
- Los nombres de las variables pueden ser un grupo de letras y dígitos, pero deben comenzar con una letra o un guion bajo.



# Contenido

## Declaración de variable y asignación de valores

- Python no nos obliga a declarar una variable antes de usarla en la aplicación. Nos permite crear una variable en el momento requerido.
- No necesitamos declarar explícitamente variables en Python. Cuando asignamos cualquier valor a la variable, esa variable se declara automáticamente.
- El operador igual (=) se utiliza para asignar valor a una variable.
- En Python, las variables son un nombre simbólico que es una referencia o puntero a un objeto. Las variables se utilizan para denotar objetos con ese nombre.



# Contenido

Si se desea nombrar una variable, se deben seguir las siguientes reglas:

- El nombre de la variable debe de estar compuesto por MAYUSCULAS, minúsculas, dígitos, y el carácter \_ (guion bajo).
- El nombre de la variable debe comenzar con una letra.
- El carácter guion bajo es considerado una letra.
- Las mayúsculas y minúsculas se tratan de forma distinta (un poco diferente que en el mundo real - Alicia y ALICIA son el mismo nombre, pero en Python son dos nombres de variable distintos, subsecuentemente, son dos variables diferentes).
- El nombre de las variables no pueden ser igual a alguna de las palabras reservadas de Python (se explicará más de esto pronto).
- Una variable se crea cuando se le asigna un valor. A diferencia de otros lenguajes de programación, no es necesario declararla.





# Contenido

## Ejemplos

```
var = "3.7.1"  
print(var)  
var = 1.2  
print(var)  
var = var + 1  
print(var)  
var = 100  
print(var)  
var = 200 + 300  
print(var)
```

## Ejemplos

```
var1 = 1  
var2 = var1  
print(var2, var1)  
  
x = 1  
x = x * 2  
x *= 2  
print(x)  
  
oveja = 25  
oveja = oveja + 1  
oveja += 1  
print(oveja)
```

## Ejemplos

```
i = 1  
j = 2  
i = i + 2 * j  
i += 2 * j  
print(i, j)  
  
var = 100  
var = var / 2  
var /= 2  
print(var)  
  
rem = 5  
rem = rem % 10  
rem %= 10  
print(rem)
```

## Ejemplos

```
i = 1  
j = 2  
i = i + 2 * j  
i += 2 * j  
print(i, j)  
  
var = 100  
var = var / 2  
var /= 2  
print(var)  
  
rem = 5  
rem = rem % 10  
rem %= 10  
print(rem)
```

## Ejemplos

```
j = 50  
x = 100  
var = 55  
rem = 99  
i = 65  
j = j - (i + var + rem)  
j -= (i + var + rem)  
x = x ** 2  
x **= 2  
print(j, x)
```



# Contenido

## Errores al declarar una variable

```
var 1 = 120
```

```
@var1 = 120
```

```
1var = 120
```

```
var-1 = 120
```



# Contenido

## Asignación múltiple

- Python nos permite asignar un valor a múltiples variables en una sola declaración, lo que también se conoce como asignaciones múltiples.
- Podemos aplicar múltiples asignaciones de dos maneras, ya sea asignando un solo valor a múltiples variables o asignando múltiples valores a múltiples variables.



# Contenido

## Asignación entre variables

```
var1, var2, var3 = 1, 2, 3
```

```
print(var1)
```

```
print(var2)
```

```
print(var3)
```

```
var1, var2, var3 = var3, var2, var1
```

```
print(var1)
```

```
print(var2)
```

```
print(var3)
```

```
var1 = var2 = var3 = 200
```

```
print(var1, var2, var3)
```



# Contenido

## Tipos de datos integrados

- Python tiene varios tipos de datos integrados , como números (enteros, flotantes, números complejos), cadenas, listas, tuplas y diccionarios.
- Hoy veremos entero y flotantes (decimales).
- Python soporta dos tipos de números - enteros y números de punto flotante (decimales).
  - Los números enteros son: - 1, 2, 22, 476,-99999
  - Los float o decimales son: - 1.0, 2.22, 22.098, 476.1,-99999.9





# Contenido

Para definir un entero, usa la siguiente sintaxis:

Para definir un float (decimal), se usa la siguiente sintaxis:

```
mientero = 1  
print(mientero)  
  
print(type(mientero))
```

Para definir un float (decimal), se usa la siguiente sintaxis:

```
mifloat = 1.0  
  
print(mifloat)  
print(type(mifloat))
```



# Contenido

## Manipular números Operadores Aritméticos

Tal como cualquier otro lenguaje de programación, los operadores de sumas, restas, multiplicaciones y divisiones pueden ser usadas con números.

Operación	Resultado
+	Suma
-	Resta
*	Multiplicación
/	División
//	División entera
**	Potencia



# Contenido

## Algunos ejemplos

```
numero1 = 10  
numero2 = 9.99  
numero3 = numero1 + numero2  
print(numero1, numero2, numero3)
```

## Algunos ejemplos

```
numero1 = 10  
numero2 = 10  
numero3 = numero1 + numero2  
print(numero1, numero2, numero3)
```



# Contenido

## Algunos ejemplos

```
x = 8 - 5  
print(x)  
x = 2 * 6  
print(x)  
x = 12 / 3  
print(x)  
x = 12 // 3  
print(x)  
X = 7 % 3  
print(x)  
Cuadrado = 7 ** 2  
print(x)
```

## Algunos ejemplos

```
cubico = 2 ** 3  
print(cubico)  
  
numero1 = 10  
numero2 = 10  
numero3 = numero1 + numero2  
print(numero1, numero2, numero3)
```



# Contenido

## Orden en las operaciones

- Cuando en una expresión aparece más de un operador, el orden de evaluación depende de las reglas de precedencia. Para los operadores matemáticos, Python sigue la convención matemática. El acrónimo PEMDAS es una forma útil de recordar las reglas:





# Contenido

## Paréntesis (P)

- Los paréntesis tienen la mayor precedencia y pueden ser usados para forzar una expresión a evaluar en el orden que se desee. Como las expresiones entre paréntesis se evalúan primero
- por ejemplo

```
x = 2 * (5 - 1)
```

```
print(x)
```

```
y = (1 + 1) ** (5 - 2)
```

```
print(y)
```



# Contenido

## Exponenciación (E)

- La exponenciación tiene la siguiente precedencia más alta. Así que  $x =$

```
2 ** 1 + 1
```

```
print(x)
```

```
y = 3 * 1 ** 3
```

```
print(y)
```



# Contenido

## Multiplicación (M), División (D), Suma (A), Resta (S)

- La multiplicación y la división tienen la misma precedencia, que es mayor que Suma y resta, que también tienen la misma precedencia.

```
x = 2 * 3 - 1  
print(x)
```



# Contenido

## Misma precedencia

- Los operadores con la misma precedencia se evalúan de izquierda a derecha.
- Así que la expresión 5-3-1 es 1, no 3, porque el 5-3 ocurre primero y luego el 1 es restado de 2.

```
x = 5 - 3 - 1  
print(x)
```



# Contenido

## Ejercicios rápidos

¿Que imprime el siguiente programa ?

```
x = 43
```

```
x = x + 1
```





# Contenido

Calcule el promedio de las siguientes variables

```
var1 = 1.0
```

```
var2 = 4.5
```

```
var3 = 5.5
```

```
var4 = 2.5
```

```
promedio = (var1 + var2 + var3 + var4)/4
```

```
print("El promedio de esos números es "+str(promedio))
```

```
print("El promedio de esos números es ", round(promedio))
```

```
print("El promedio de esos números es ", round(promedio, 2))
```



# Contenido

Calcule el área y perímetro de un cuadrado

Formulas:

lado = 10

area = lado \* lado

perimetro = lado + lado + lado + lado

print(lado)

print(perimetro)



El futuro digital  
es de todos

MinTIC



# Contenido

**GRACIAS!!!**

**POR COMPLETAR TU PRIMERA  
CLASE!!**