

Informatorio Chaco | 2024 | Data Analytics

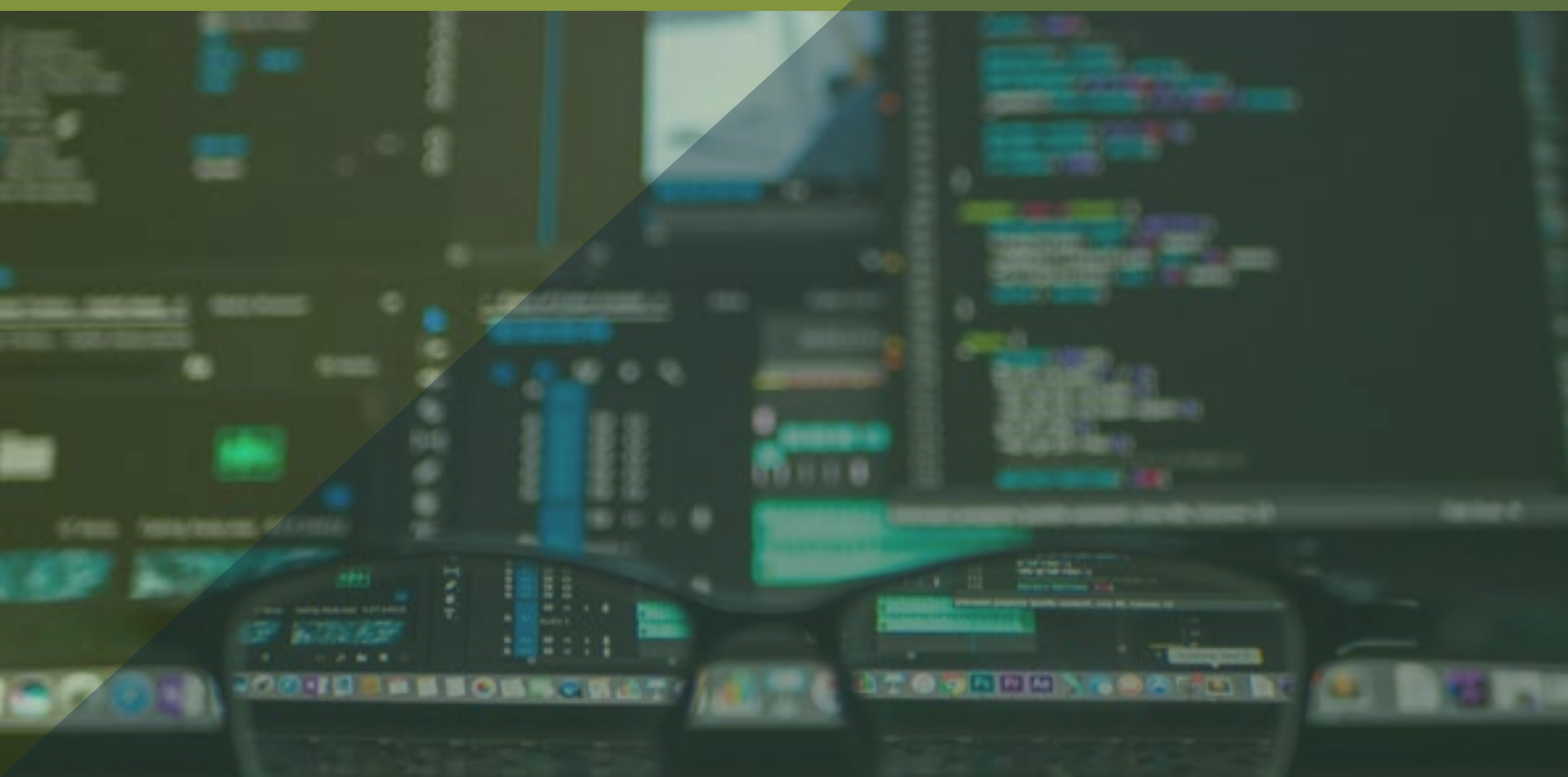
Clase 1

VARIABLES Y OPERADORES, ESTRUCTURAS DE DATOS, ESTRUCTURAS DE CONTROL

Variables y operadores | Sintaxis | Tipos de datos | Estructuras de datos | Estructuras de control | Ejemplos.

módulo

2




Variables y Operadores

Variables

Las variables en Python se utilizan para almacenar datos que pueden ser manipulados durante la ejecución del programa. Python es un lenguaje de tipado dinámico, lo que significa que no es necesario declarar el tipo de la variable antes de asignarle un valor.

Sintaxis básica:

python

 Copiar código

```
nombre_variable = valor
```

Tipos de Datos

- ▷ **int**: Números enteros.
- ▷ **float**: Números decimales.
- ▷ **str**: Cadenas de texto.
- ▷ **bool**: Valores booleanos (True, False).

Ejemplo:

python

 Copiar código

```
numero = 10  
texto = "Hola, mundo"
```

Operadores:

ARITMÉTICOS

+ : Suma.
- : Resta.
* : Multiplicación.
/ : División.
% : Módulo.
** : Exponenciación.
// : División entera.

*Ver ejemplo en
imagen 3.*

DE COMPARACIÓN

== : Igual a.
!= : Distinto de.
> : Mayor que.
< : Menor que.
>= : Mayor o igual que.
<= : Menor o igual que.

*Ver ejemplo en
imagen 4.*

LÓGICOS

and : Y lógico.
or : O lógico.
not : Negación lógica.

*Ver ejemplo en
imagen 5.*

Imagen 3 (aritméticos).

```
python Copiar código

a = 10
b = 3
print(a + b) # Salida: 13
print(a - b) # Salida: 7
print(a * b) # Salida: 30
print(a / b) # Salida: 3.3333333333333335
print(a % b) # Salida: 1
print(a ** b) # Salida: 1000
print(a // b) # Salida: 3
```

Imagen 4 (de comparación).

```
python Copiar código

a = 10
b = 3
print(a == b) # Salida: False
print(a != b) # Salida: True
print(a > b) # Salida: True
print(a < b) # Salida: False
print(a >= b) # Salida: True
print(a <= b) # Salida: False
```

Imagen 5 (lógicos).

```
python Copiar código

a = True
b = False
print(a and b) # Salida: False
print(a or b) # Salida: True
print(not a) # Salida: False
```


Estructuras de Datos

Listas

Las listas son colecciones ordenadas y mutables de elementos. Se crean utilizando corchetes `[]`.

Sintaxis básica:


python

 Copiar código

```
mi_lista = [1, 2, 3, 4]
```

Acceso a elementos:

python

 Copiar código

```
print(mi_lista[0]) # Salida: 1
```

Modificación de elementos:


python

 Copiar código

```
mi_lista[1] = 10
```

Métodos comunes:

python

 Copiar código

```
mi_lista.append(5)  
mi_lista.remove(3)  
mi_lista.pop()  
mi_lista.sort()  
mi_lista.reverse()
```

Tuplas


Las tuplas son colecciones ordenadas e inmutables de elementos. Se crean utilizando paréntesis ().

Inmutabilidad:

► Las tuplas no se pueden modificar después de su creación.

Sintaxis básica:


python

 Copiar código

```
mi_tupla = (1, 2, 3)
```

Acceso a elementos:

python

 Copiar código

```
print(mi_tupla[0]) # Salida: 1
```

Diccionarios

Los diccionarios son colecciones desordenadas de pares clave-valor. Se crean utilizando llaves {}.

Sintaxis básica:


python

 Copiar código

```
mi_dict = {'clave1': 'valor1', 'clave2': 'valor2'}
```

Acceso a valores:


python

 Copiar código

```
print(mi_dict['clave1']) # Salida: valor1
```

Modificación de valores:


python

 Copiar código

```
mi_dict['clave1'] = 'nuevo_valor'
```

Métodos comunes:

python

 Copiar código

```
claves = mi_dict.keys()
valores = mi_dict.values()
items = mi_dict.items()
```


Estructuras de Control

Condicionales

Las estructuras condicionales permiten ejecutar diferentes bloques de código en función de condiciones.

Sintaxis básica:

python

 Copiar código

```
if condicion:
    # Código a ejecutar si la condición es verdadera
elif otra_condicion:
    # Código a ejecutar si la otra condición es verdadera
else:
    # Código a ejecutar si ninguna condición anterior es verdadera
```

Ejemplo:

python

 Copiar código

```
a = 10
if a > 5:
    print("Mayor que 5")
elif a == 5:
    print("Igual a 5")
else:
    print("Menor que 5")
```


Bucles

Los bucles permiten repetir un bloque de código varias veces.

Bucle for

Sintaxis básica:


python

 Copiar código

```
for elemento in secuencia:
    # Código a ejecutar para cada elemento
```

Ejemplo:

python

 Copiar código

```
for i in range(5):
    print(i)
```

Bucle While

Sintaxis básica:

python

 Copiar código

```
while condicion:  
    # Código a ejecutar mientras la condición sea verdadera
```

Ejemplo:

python

 Copiar código


```
a = 0  
while a < 5:  
    print(a)  
    a += 1
```

Comprensiones de Listas

Las comprensiones de listas proporcionan una forma concisa de crear listas.

Sintaxis:


python

 Copiar código

```
nueva_lista = [expresion for elemento in iterable if condicion]
```

Ejemplo:

python

 Copiar código

```
cuadrados = [x**2 for x in range(10) if x % 2 == 0]
```