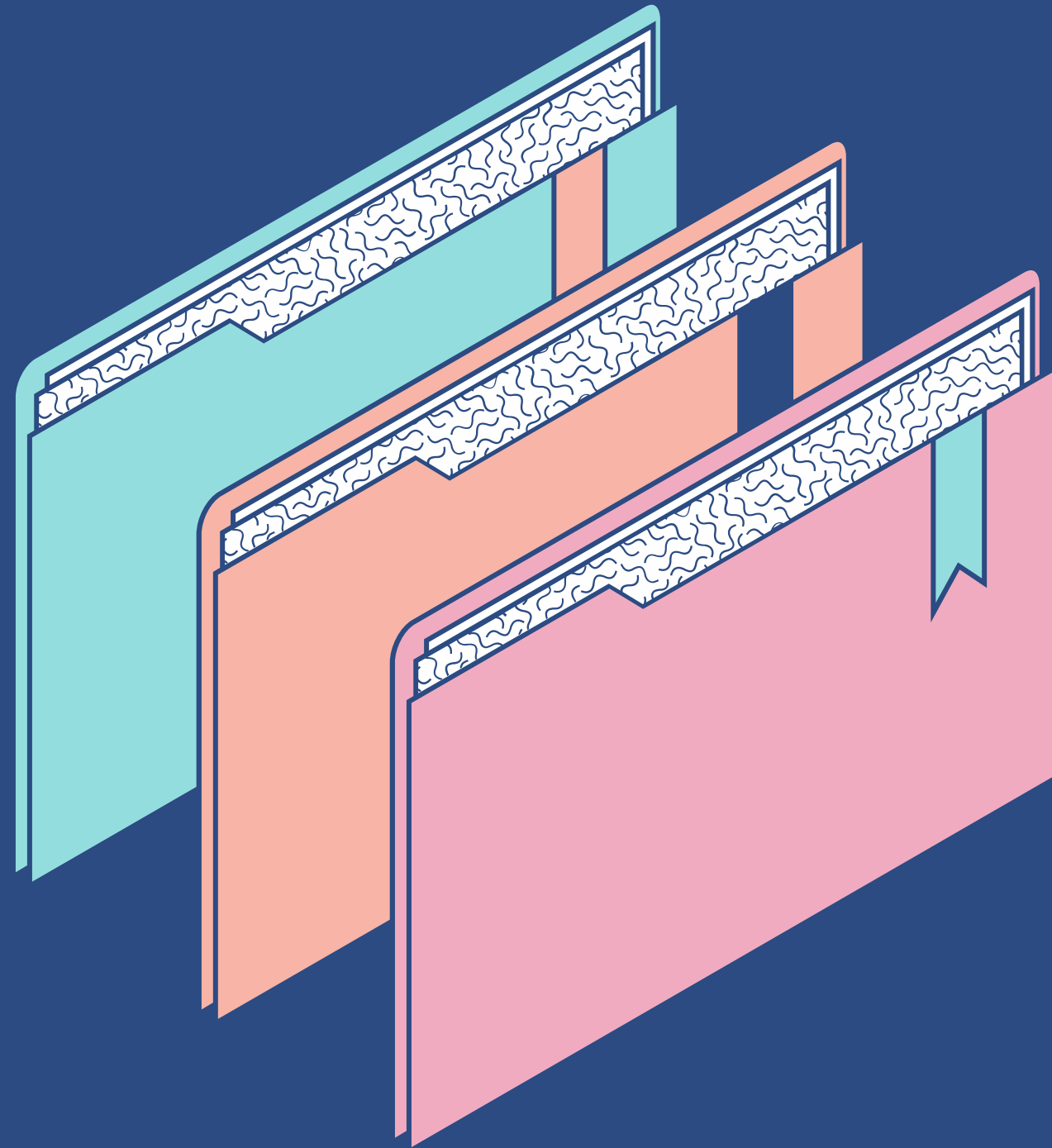




CLASE 14 DE JUNIO

Sintáxis básica, variables y expresiones condicionales

Python desde cero - NineHub



Contenido

- Tipos de Datos Básicos
- Declaración y Uso de Variables
- Operaciones Básicas
- If, Elif, Else
- Operadores de comparación Lógicos

Conjunto de caracteres de Python

- Letras: de **A** hasta **Z**; de **a** hasta **z**
- Dígitos: de **0** a **9**
- Símbolos especiales: **=, -, +, *, /**, etc.
- Espacios en blanco: Barra espaciadora, tab, enter (nueva linea)
- Otros caracteres: Todos los caracteres en ASCII o Unicode pueden ser procesados.

Variables

Una variable es un nombre dado a un ubicación de memoria en un programa.

Asignación de variables: *variable* = *valor*

- nombre = “Sandra”
- edad = 29
- precio = 49.99

EJEMPLO Cómo usar variables

```
1 nombre = "Sandra"  
2 print("Mi nombre es", nombre)
```

```
Mi nombre es Sandra
```

```
1 edad = 24  
2 edad = edad + 10  
3 print(edad)
```

```
34
```

Reglas de identificadores

(nombres para variables, funciones, etc.)

1. Los identificadores pueden ser una combinación de letras mayúsculas y minúsculas, dígitos o un guion bajo (_). Por lo tanto, **miVariable**, **variable_1**, **variable_for_print** son todos identificadores válidos en Python.
2. Un identificador no puede comenzar con un dígito. Así que mientras **variable1** es válido, **1variable** no lo es.
3. No podemos usar símbolos especiales como **!, #, @, %, \$**, etc., en nuestro identificador.
4. Un identificador puede tener cualquier longitud.

Tipos de datos

- **Enteros:**
 - Positivos, negativos, 0
- **Cadena de texto (String)**
 - "Sandra", "hola mundo"
- **Flotante (Float)**
 - 3.99, 2.509528392, 9.0
- **Booleano (Boolean)**
 - True, False
- **Ninguno (None)**
 - *a = None*

EJEMPLO Tipos de datos

```
1 nombre = "Sandra"
2 edad = 21
3 precio = 4.5
4 es_domingo = False
5 v1 = None
6 |
7 print(type(nombre))
8 print(type(edad))
9 print(type(precio))
10 print(type(es_domingo))
11 print(type(v1))
```

```
<class 'str'>
<class 'int'>
<class 'float'>
<class 'bool'>
<class 'NoneType'>
```


Keywords (Palabras reservadas)

| | | | | |
|--------|----------|---------|----------|--------|
| False | await | else | import | pass |
| None | break | except | in | raise |
| True | class | finally | is | return |
| and | continue | for | lambda | try |
| as | def | from | nonlocal | while |
| assert | del | global | not | with |
| async | elif | if | or | yield |

(Podría haber alguna distinta según la versión de Python)

Comentarios en Python

```
#Comentarios de una  
linea
```

```
"""  
Comentarios de  
varias  
lineas  
"""
```

Tipos de Operadores

Un operador es un símbolo que realiza una cierta operación entre operandos.

- **Operadores Aritméticos:**
 - $+$, $-$, $*$, $/$, $\%$, $**$
- **Operadores Relacionales / de Comparación**
 - $==$, $!=$, $>$, $<$, $>=$, $<=$
- **Operadores de Asignación**
 - $=$, $+=$, $-=$, $*=$, $/=$, $\%=$, $**=$
- **Operadores Lógicos**
 - not , and , or

Conversión de tipos

```
a, b = 1, 2.0
```

```
suma = a + b
```

```
# sí funciona
```

```
a, b = 1, "2"
```

```
suma = a + b
```

```
# error
```

Conversión de tipos

```
a, b = 1, "2"
```

```
c = int(b)
```

```
suma = a + c
```

```
# sí funciona
```

EJERCICIOS:

- Escribe un programa para ingresar 2 números e imprimir su suma.
- Escribe un programa para ingresar el lado de un cuadrado e imprimir su área.

Expresiones Condicionales y Operadores de comparación Lógicos



Expresiones booleanas

UNA EXPRESIÓN BOOLEANA ES AQUELLA QUE DA COMO RESULTADO VERDADERO (TRUE) O FALSO (FALSE). EL TIPO DE DATO BOOLEANO NO SON CADENAS.



Las expresiones booleanas nos permiten saber si algo es verdadero o no. Puede ser utiles para comparar o decidir si algo se ejecuta en el codigo o no.

```
>>> 5==5
True
>>> 4==3
False
```

```
>>> a = 3
>>> b = 4
>>> c = a**2 + b**2
>>> if c == 25:
...     print("el valor de c es 25")
...
el valor de c es 25
>>> if c != 25:
...     print("el valor de c no es 25")
...
>>> |
```

Operadores de Comparación

LOS OPERADORES DE COMPARACIÓN SON EXPRESIONES BOOLEANAS, ESTOS PERMITEN COMPARAR VARIABLES O DATOS.

Listado operadores de comparación

- **$x == y$** : x es igual que y
 - **$x != y$** : x es diferente de y
 - **$x > y$** : x es mayor que y
 - **$x >= y$** : x es mayor o igual que y
 - **$x < y$** : x es menor que y
 - **$x <= y$** : x es menor o igual que y
-



Operadores Lógicos

LOS OPERADORES LÓGICOS SON OTRO TIPO DE EXPRESIONES BOOLEANAS. ESTOS PERMITEN EVALUAR CONDICIONES. ESTOS PUEDEN SER “AND”, “OR” Y “NOT”

| x | y | x and y | x or y |
|----------|----------|----------------|---------------|
| True | True | True | True |
| True | False | False | True |
| False | True | False | True |
| False | False | False | False |

Analicemos el siguiente código

```
>>> x = 15
>>> x > 14 and x < 25
True
>>> x == 15 and x < 15
False
>>> x > 15 or x < 14
False
>>> x > 14 or x < 15
True
```

Ejecuciones Condicionadas

Permiten verificar condiciones y tomar acción dependiendo de si se cumplen las condiciones o no. En python se utiliza if para determinar si una condicion se cumple o no, apoyandose de las expresiones booleanas.

Un ejemplo puede ser, si usted tiene mas de 61 puntos en un curso, entonces el curso es aprobado. En código esto se vería de la siguiente manera.

```
>>> calificacion = 65
>>> if calificacion > 61:
...     print("Usted ha ganado el curso")
...
Usted ha ganado el curso
>>>
```

“If” también puede indicar dos posibilidades.

- Si se cumple la condición, se ejecuta alguna línea de código.
- De lo contrario, se puede ejecutar otra línea contemplando el hecho que no se cumple la condición.
- Para condiciones alternativas se utiliza la sentencia “else”.

```
>>> calificación = 49
>>> if calificación > 61:
...     print("Ha ganado el curso")
... else:
...     print("Ha perdido el curso")
...
Ha perdido el curso
```

A veces es necesario poder evaluar mas de dos posibles condiciones. Para ello se utilizan las condiciones encadenadas con la sentencia “elif” es una abreviación de *else if*.

```
1 calificacion = 94
2 if calificacion < 61:
3     print("Ha perdido el curso")
4 elif calificacion >= 80:
5     print("Ha ganado el curso con honores")
6 else:
7     print("Ha ganado el curso")
```

```
>>> %Run -c $EDITOR_CONTENT
    Ha ganado el curso con honores
>>>
```

Consideraciones de las condiciones encadenadas.

Solo una de las condiciones será ejecutada. De acuerdo con la condición que sea verdadera. Estas son evaluadas en orden, con la primera que sea verdadera. Las otras condiciones pueden ser verdaderas, pero como vienen después en el código ya no se ejecutarán. Se pueden agregar tantos elif como se desee y el uso de else al final es opcional. Dependerá del estilo de programación de cada uno.

```
calificacion = 94
if calificacion >= 61:
    print("Ha ganado el curso")
elif calificacion >= 80:
    print("Ha ganado el curso con honores")
else:
    print("Ha perdido el curso")
```

```
>>> %Run -c $EDITOR_CONTENT

Ha ganado el curso
```


Resolvamos el siguiente problema tomando en cuenta lo aprendido en esta clase y la clase anterior.

PROBLEMA:

EN UNA COMUNIDAD LAS AUTORIDADES LOCALES DESEAN AUTOMATIZAR EL PROCESO DE VERIFICACIÓN DE SI UNA PERSONA ES LEGIBLE PARA VOTAR O NO EN LAS PRÓXIMAS ELECCIONES. LAS LEYES ELECTORALES ESTIPULAN QUE UNA PERSONA ES CAPAZ DE VOTAR SI CUMPLE CON LO SIGUIENTE:

1. TENER AL MENOS 18 AÑOS DE EDAD
2. SER CIUDADANO REGISTRADO DE LA COMUNIDAD
3. NO TENER RESTRICCIONES LEGALES QUE LE IMPIDAN VOTAR, ES DECIR NO HABER SIDO CONDENADO POR ALGÚN DELITO EN LA COMUNIDAD.

PARA AYUDAR A LAS AUTORIDADES LOCALES SE TE PIDE QUE DESARROLLES UN PROGRAMA E PYTHON PARA VERIFICAR LE LEGIBILIDAD DE UNA PERSONA PARA VOTAR. TOMA EN CUENTA LAS CONSIDERACIONES ANTERIORES, RECIBE COMO ENTRADA DEL USUARIO LA EDAD, SU CIUDADANIA Y SI FUE CONDENADO POR ALGÚN DELITO O NO. EL PROGRAMA DEBE REGRESAR SI LA PERSONA ES CAPAZ DE VOTAR O NO.