

PROGRAMACIÓN II

Aprendiendo Python

Que hacemos hoy

1

**Clasificación de los
lenguajes**

2

El IDE

3

Variables y constantes

4

Tipos de datos

5

Conversiones

6

**Nivel inicial
Entradas y Salidas**

La semana anterior

- Print
- Variables
 - Tipado Dinámico
 - Tipos de datos
 - ¿Qué son las constantes?

Comentarios

- De una sola línea

- Multilínea

```
# esto es un comentario
```

```
''' esto es un comentario  
De muchas líneas'''
```

Importante: los comentarios en la misma línea del código deben separarse con dos espacios en blanco. Luego del # debe ir un solo espacio en blanco

Palabras para identificar tareas



TODO

Esto es Algo por hacer



FIXME

Esto debe corregirse



estamos y a dónde vamos ...

1

¿Qué es un
lenguaje de
programación?

2

Lenguaje de la máquina
Variables

3

Condicionales

4

Bucles

Algoritmos

Cronograma

- Uso de la instrucción input
- Uso potenciado de Print
- Operadores Aritméticos
- Instrucción de Decisión
 - If
 - If –else
 - Elif
- Entorno de la Variable
- Import + librerías

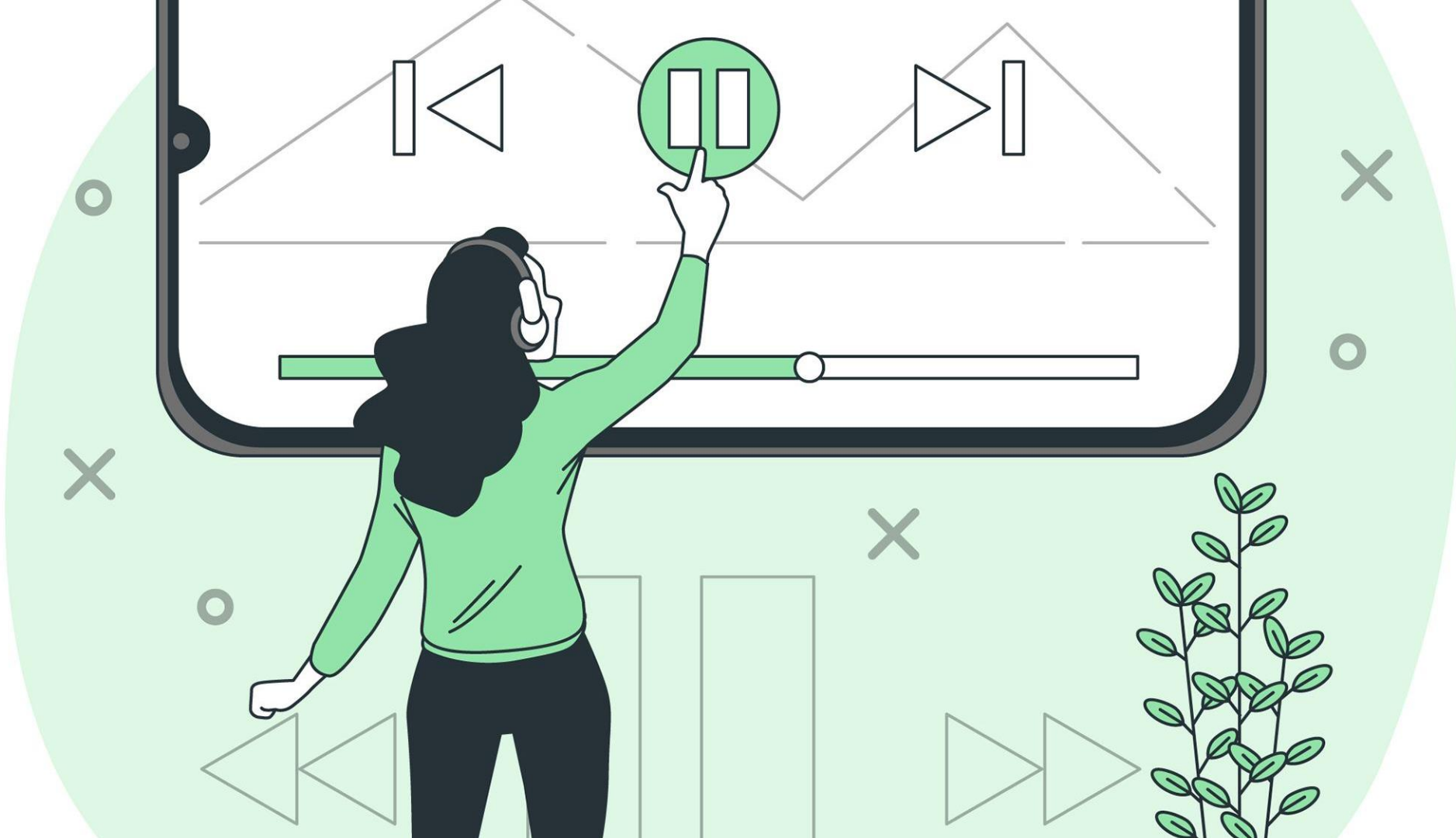
Operadores Aritméticos

- + suma
- - resta
- / División
- * Multiplicación
- % Módulo
- ** Potencia
- // División de Enteros

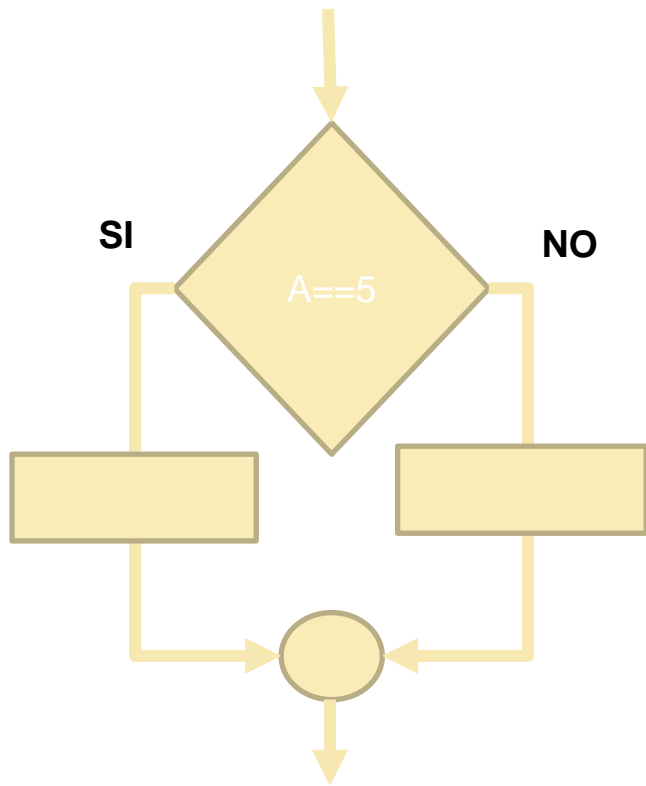
Algo más con el print

- Formateo de String
- Concatenar
- El operador end=
- El operador sep=



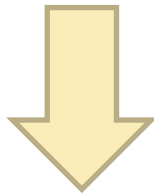


Instrucciones de decisión



La condición

- [Valor] [operador relacional] [valor]



Bool

Operadores relacionales

==

IGUAL

- Usar el doble igual para representar una comparativa de igualdad. Este operador retorna true cuando los dos operadores son iguales

!=

DISTINTO

- Devuelve true cuando los valores son diferentes

<

MENOR

- Si la condición es $A < B$ devuelve true si A es menor que B

>

MAYOR

- Si la condición es $A < B$ devuelve true si A es mayor que B

<=

MENOR IGUAL

- Si la condición es $A < B$ devuelve true si A es menor e igual que B

>=

MAYOR IGUAL

- Si la condición es $A < B$ devuelve true si A es mayor e igual que B

¿Por qué usamos el IF?



¿La puerta solo puede estar abierta?

Instrucción IF: Un ejemplo con código ...

```
numero=5  
#acá decimos que si numero es igual a 5  
if numero==5:  
    print("El número es 5")  
    print("esta es la última línea dentro del if")  
print("Esta línea está fuera del if")
```


Cuándo hay dos opciones

CASO 1



CASO 2



- Podemos ver que la Puerta tiene 2 estados: Abierto o Cerrado

El IF ... ELSE: Un ejemplo con código ...

```
numero=5
#acá decimos que si numero es igual a 5
if numero==5:
    print("El número es 5")
else:
    print("El número no es 5")
print ("Esta línea está fuera del if")
```

y... Cuándo hay mas de dos estados posibles ☹



- Podemos ver que la Puerta tiene 3 estados: Abierto, Cerrado, y entreabierto



El if anidado



Instrucción ELIF

```
numero=0
#acá decimos que si numero es igual a 5
if numero>0:
    print("El número es mayor a cero")
elif numero==0:
    print ("El número es igual a cero")
else:
    print("El número no es menor a cero")
print ("Esta línea está fuera del if")
```

IF anidado

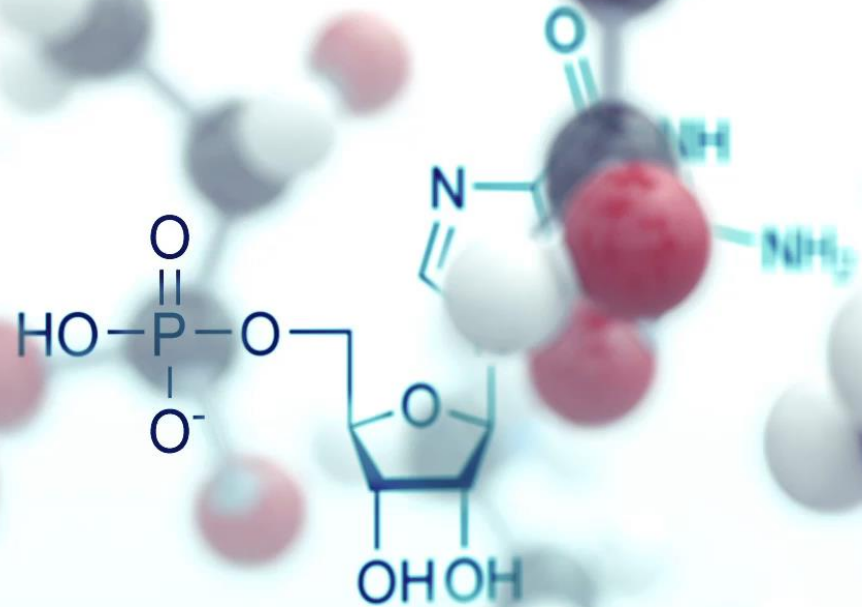
```
# El numero es mayor, igual o menor a cero?
numero=0
if numero==0:
    print("Es igual a cero")
else:
    if numero>0:
        print("Es mayor a cero")
    else:
        print("Es menor a cero")
```


Otra mirada del IF

Variable= "Salida True" **if** (condición) **else** "salida false"

```
numero=5  
#acá decimos que si numero es igual a 5  
respuesta= "El número es 5" if (numero==5) else "El número no es 5"  
print (respuesta)
```


Tratando de usar pocos if



Operadores lógicos

and \rightarrow se puede decir Y

or \rightarrow se puede decir o

not – se puede decir no

Operador lógico AND

A and B		
A	B	A and B
T	T	T
T	F	F
F	T	F
F	F	F

Operador lógico OR

A or B		
A	B	A or B
T	T	T
T	F	T
F	T	T
F	F	F

Operador lógico NOT

Not A		
A	not A	
T	F	
F	T	

Grupos de Operadores

- Como pasa con los operadores aritméticos, los () también se usan

(True and False) or true
False or true
true

Grupos de Operadores

((True and False) or (not true)) and true



(False or (not true)) and true



(False or False) and true



(False) and true



False



EXPLORER

OPEN EDITORS

- ControladorAlumno.php controladores

CRUDPDO

- controladores
 - ControladorAlumno.php**
 - ControladorBD.php
 - Paginador.php
- css
- fotos
- modelos
- nbproject
- recursos
- script
- sql
- utilidades
- vistas
- leeme.txt
- dirs.php
- favicon.png
- index.php

OUTLINE

ControladorAlumno.php x

```
1 <?php
2
3 /*
4  * To change this license header, choose License Headers in Project Properties.
5  * To change this template file, choose Tools | Templates
6  * and open the template in the editor.
7  */
8
9 /**
10  * Description of ControladorAlumno
11  *
12  * @author link
13  */
14 require_once MODEL_PATH . "Alumno.php";
15 require_once CONTROLLER_PATH . "ControladorBD.php";
16
17 class ControladorAlumno {
18
19     // Variable instancia para Singleton
20     static private $instancia = null;
21
22     // constructor--> Private por el patrón Singleton
23     private function __construct() {
24         //echo "Conector creado";
25     }
26
27     /**
28      * Patrón Singleton. Obtiene una instancia del Manejador de la BD
29      * @return instancia de conexion
30      */
31     public static function getControlador() {
32         if (self::$instancia == null) {
33             self::$instancia = new ControladorAlumno();
34         }
35         return self::$instancia;
```



**Nos vemos la
próxima clase**

¡ Qué tengan una Buena semana!