AVANCE 0: En clases definiremos el LP con el que van a trabajar, el alcance del proyecto, vocabulario y sintaxis a validar. De manera general todos los grupos validarán lo siguiente según su LP asignado:

**Alcance del proyecto:**

Nuestro proyecto analizara la forma de la declaración de la variable y estructuras de control, además se manejara de una manera adecuada el vocabulario de la sintaxis del lenguaje de programación en este caso el lenguaje Python.

* **Definir la escritura correcta de variables y su inicialización.**

Las variables pueden ser escritas con cualquier carácter alfanumérico, y el carácter especial “\_”. Las variables tienen que empezar con un carácter alfabético o el carácter especial “\_”.

Y respecto a su inicialización debe estar precedida de su dato, no se aceptan variables sin su valor correspondiente.

Ejemplo:

1. VARIABLE = 12
2. Variable = 1
3. V\_araible = 2
4. v\_ariable = 3
5. a , b = 5, 12
6. a = b = 10

* **Tipos de datos primitivos: enteros, flotantes, booleanos**

|  |  |
| --- | --- |
| Tipo de Dato | Ejemplo |
| Enteros | Variable = 1 |
| Flotantes | Variable = 1.2 |
| Booleanos | Variable = False ó True |

* **Tipos de datos estructurados: str, listas, tuplas, conjuntos, etc.**

|  |  |
| --- | --- |
| Tipo de Dato Estructurado | Ejemplo |
| Str(String) | 1. Cadena = “string” 2. Cadena = ‘string’ |
| Lista | 1. Lista = [ 1, 2, 3 ] 2. lista = [“A”,”B”,”C”] 3. lista = [2.1 , 3.5, 2.3] |
| Tupla | 1. Tupla = (2,3) 2. tupla = (2,5,3,…,) |
| Conjunto | 1. Conjunto = {1, 2, 3} 2. conjunto = {“A”,”B”,”C”} 3. conjunto = {2.1 , 3.5, 2.3} |
| Diccionario | Diccionario = { “Clave1” : “Valor1”, “Clave2” : “Valor2”} |

* **Definir las operaciones matemáticas y condiciones de acuerdo a los tipos de datos mencionados.**

|  |  |
| --- | --- |
| Operación | Ejemplo |
| Adición | a = 4 + 5 |
| Sustracción | d = 4 – 5 |
| Multiplicación | c = 4 \* 5 |
| División | b = 4 / 5  b = 4//5 división con resultado entero |
| Exponencial | e = 4 \*\* 5 |
| Mod | f = 4 % 5 |

* **Métodos impresión, lectura de datos.**

1. **Imprimir**

Se utiliza la función print(), la cual recibe cualquier objeto y lo convierte en string, y muestra por pantalla.

Print(“Hello World”) -> Hello World

1. **Lectura de datos.**
2. **Inputs del usuario.**

Se utiliza la funcion input(), la cual recibe input del usuario y lo guarda en un string. Esta función puede recibir como parámetro un mensaje que se le mostrara al usuario.

1. **Archivos.**

Para leer archivos, se tiene que almacenar el archivo con

f = open(“nombrearchivo”, “r”)

Una vez hecho esto, se puede utilizar las varias funciones de lectura, como f.read(), f.readline().

* **Definir 3 estructuras de control básicas.**

1. **if** ( condición ):

orden a ejecutar.

En condición puede ir cualquier dato o función que pueda devolver un tipo de dato booleano.

Y en Orden puede ir cualquier cosa incluso otra condición o otra estructura.

Ejemplo:

if (variable == 2):

Print(“Es igual”)

1. **while** ( condición ):

orden a ejecutar.

En condición puede ir cualquier dato o función que pueda devolver un tipo de dato booleano.

Y en Orden puede ir cualquier cosa incluso otra condición o otra estructura Ejemplo:

while(variable = = True ):

print(“infinito”)

1. **for** elemento **in** (lista, conjunto,tupla):

**sentencia.**

Elemento es una variable.

Puede contener una lista o tipo de dato estructurado.

Sentencia puede existir otra estructura**.**

Ejemplo:

For i in range(10):

Print(“For ejecutandose”)

* **Para datos estructurados definir al menos 2 métodos.**

Para listas.

Lista.append(elemento)

Lista.remove(elemento)

* **Creación de funciones.**

**def** nombre\_funcion ( parámetro\_1, parámetro\_2,… parámetro\_N)

puede o no tener “**return”**

Nombre de la Función puede ser cualquiera siempre y cuando cumpla estándares de variables al igual que los parámetros.

Y la sentencia puede retornar un valor o solo ejecutar estructuras de control o imprimir.

Ejemplo:

def función(a,b,c):

z = a+b+c

print(z)

return z

**Operaciones Lógicas**

|  |  |
| --- | --- |
| Operación Lógica | Ejemplo |
| and(“y”) | Variable = True  Variable2= False  Variable and Variable2 |
| or(“o”) | Variable = True  Variable2= False  Variable or Variable2 |
| not(“no”) | Variable = False  not Variable |

OPERADORES DE COMPARACION

|  |  |
| --- | --- |
| Operación Lógica | Ejemplo |
| = = (“igual a”) | Variable = 1  Variable2= 1  Variable = = Variable2 |
| ! = (“no igual a”) | Variable = True  Variable2= False  Variable ! = Variable2 |