**PROGRAMACIÓN**

* Python es un lenguaje de alto nivel.
* Alto nivel se conecta con el software (se acerca al lenguaje humano) y el de bajo de nivel se conecta con la arquitectura (lenguaje máquina).
* String
* Ntn
* Float entero y decimales
* Printt en la pantalla lo que está entre comillas
* Se recomienda usar minúsculas y ser ordenado
* Nombre edad altura, son variables para el computador.
* nombre= "Juan"  
  edad = "23"  
  altura = 1.75  
  print (nombre, edad, altura)

-Se asigna la fariable Juan, luego se colocan las variables….. se puede cambiar Juan (practico)

-La clave es edad, altura, nombre y la caja es considerada como nombre, edad, altura.

- Lo entre comilla es lo que aparece en pantalla, en este caso Juan

Ejercicio 2

nombre = "Loreto"  
print(type(nombre))

* **Si coloco type dentro es para indicar el tipo de variable… Este indica el tipos de variable.**
* Reglas para nombrar variables, no se pueden usar palabras claves del sistema, no pueden ser mayúsculas.
* x = 5  
  x = 10  
  print(x)

Si ambas variables tienen letra X. Python reconoce la última.

suma = 5 + 4  
resta = 5 - 3  
print(suma,resta)

* Se tengo cajas de suma y resta, print debe decir “suma y resta”, separadas por una coma.
* Python entiende que dos iguales son igual, a diferencia del humano que entiende que es solo un =
* print(5>3)  
  print(2!=2)
* El soigno de exclamación con el igual, Es distinto
* Aquí sale falso porque el 2 con el 2 son iguales.

a = 5  
b = 3  
resultado = a > b and a !=b  
print(resultado)

* en and o en or varia el tema de verdadero o falso, se debe analizar y aplicar as reglas de verdad.

EJERCICIO

**nombre = "python"  
edad = 25  
altura = 1.75  
  
print(nombre , edad , altura)  
print(type(nombre))  
print(nombre , edad , altura)  
print(type(edad))  
print(nombre , edad , altura)  
print(type(altura))**

RESULTADO

"C:\Users\PCAIO-012\PycharmProjects\Clase repaso prueba\.venv\Scripts\python.exe" "C:\Users\PCAIO-012\PycharmProjects\Clase repaso prueba\PruebaPrg.py"

python 25 1.75

<class 'str'>

python 25 1.75

<class 'int'>

python 25 1.75

<class 'float'>

Process finished with exit code 0

declarar ciertas variables y entregar el tipo de variable, con nombre edad y altura da stn porque es una variable y 1075 porque es float.

EJERCICIO

**x = 10  
y = 3  
  
#operaciones  
  
suma = x + y  
resta = x - y  
multiplicacion = x \* y  
division = x / y  
potencia = x \*\* y  
print (suma, resta, multiplicacion, division, potencia)**

RESULTADO

**13 7 30 3.3333333333333335 1000**

**Process finished with exit code 0**

* Pregunta de prueba, para imprimir los resultados siempre se debe cerrar con un print.

mensaje = "Bienvenido al curso de programación"  
numero = 100  
pi = 3.14  
  
print(mensaje,numero,pi)

En el print es mejor escribir todo en el paréntesis de print, y la manera menose eficiente hacerlo, es separado… se debe ocupar la menor cantidad de espacio posible.

EJERCICIO

**x = 10  
y = 4  
  
#suma  
suma = x + y  
print("suma",suma)  
  
#resta  
resta = x - y  
print("resta",resta)  
  
#division  
division = x/y  
print("division",division)  
  
#multiplicacion  
multiplicacion = x\*y  
print("multiplicacion",multiplicacion)  
  
#potencia  
potencia = x\*\*y  
print("potencia",potencia)**

RESULTADO

"C:\Users\PCAIO-012\PycharmProjects\Clase repaso prueba\.venv\Scripts\python.exe" "C:\Users\PCAIO-012\PycharmProjects\Clase repaso prueba\PruebaPrg.py"

suma 14

resta 6

division 2.5

multiplicacion 40

potencia 10000

Process finished with exit code 0

EJERCICIO

**def saludar():  
 print("hola,¿como estas?")  
saludar()**

RESULTADO

**hola,¿como estas?**

Process finished with exit code 0

* Funciones es con “def”, es para llamar funciones
* Y lo que esta entre paréntesis es el argumento, aquí al imprmir podré escribir lo que quiera.

EJERCICIO

**def(saludar\_persona(nombre)):  
 print(f"hola.{nombre},bienvenido")  
   
#LLamar a la funcion  
saludar\_persona("Loreto")**

* Argumentos es nombre
* F se usa para hacer la unión de los string (una cadena con la información hola, nombre…)

EJERCICIO

**def sumar\_numeros(num1,num2):  
 resultado = num1 + num2  
 print(f"la suma de {num1} y {num2} es {resultado}")  
  
sumar\_numeros(5,3)**

RESULTADO

**la suma de 5 y 3 es 8**

**Process finished with exit code 0**

EJERCICIO

**def multiplicar\_por\_dos(numero):**

**resultado = numero \* 2**

**print(f"{numero} multiplicado por 2 es: {resultado}")**

**multiplicar\_por\_dos(8)**

RESULTADO

**8 multiplicado por 2 es: 16**

* Va e llave porque son eslavones

EJERCICIO

**def area\_rectangulo(ancho,alto):  
 return ancho\*alto  
  
resultado = area\_rectangulo(5,10)  
print(f"El area del rectangulo es: {resultado}")**

RESULTADO

**El area del rectangulo es: 50**

EJERCICIO

**def area\_triangulo(base,altura):  
 return base\*altura/2  
  
resultado = area\_triangulo(6,2)  
print(f"El area del triangulo es: {resultado}")**

RESULTADO

**El area del triangulo es: 6.0**

* Return es la multiplicación de eso.
* El resultado guarda el área del rectángulo.

EJERCICIO

def convertir\_a\_fahrenheit(celsius):  
 return celsius \*9/5 +32  
  
resultado = convertir\_a\_fahrenheit(25)  
print(f"30 grados celsius son {resultado} grados fahrenheit")

RESULTADO

30 grados celsius son 77.0 grados fahrenheit

* Grados fareomngeit se vocvierte la función y se hace la llamada return
* Lego caja de resultado de onversion para luego imprimir en pantalla para unir eslavones.

EJERCICIO

def contar\_letra(palabra.letra):  
 return palabra.count(letra)  
  
resultado = contar\_letra("programacion","o")  
print(f"la letra o aparece {resultado} veces en 'programacion'")

**ARREGLAR EL ULTIMO**

* Una basse de datos es
* El campo es el apellido, los nombres ( es específico)
* El archivo es todo lo que conforma la base de datos
* Necesitamos una base de datos y se utiliza la palabra archivo.
* Representación en el modelado .. entidad con atributos .. la entidad es lo que sale arriba como título.
* Clave primaria permite identificar los registros de nuestra tabla.
* Clave foránea se da cuando le prest ala clave primaria a otra tabla.
* Entidades; Clientes productos pedidos.
* Aributos:
* cliente (id cliente, nombre dirección y correo electrónico)
* Producto: código producto , tipo
* Pedido: id pedido, dirección, precio
* Int es enetero , varchar es para textos
* 255 es por hecadecimal

create table usuarios(

ID\_usuario int primary key,

nombre varchar(100),

direccion varchar(255),

telefono varchar(15)

);

Create table libros(

ISBN varchar(13) primary key,

titulo varchar(255),

autor varchar(100),

año\_publicacion INT

);

create table prestamos(

id\_prestamo int primary key,

fecha\_prestamo date,

fecha\_devolucion date,

id\_usuario int,

ISBN varchar(13),

foreign key (ID\_usuario) references usuarios(ID\_usuarios),

foreign key (ISBN) references libros(ISBN)

);

El asterisco dice que seleccione algo de la tabla usuario. De la biblioteca

El punto es para acortar camino

El join es la llamada de la función

select (usuario.nombre, libro.titulo, prestamos. fecha\_prestamo)

from prestamos

join usuarios on pretamos.ID\_usuarios = usuarios.ID\_usuario

join libros on pretamos.ISBN = libros\_ISBN

as es como

count contar

ISBN es al id del libro

select libros.titulo, count(prestamos.ID\_prestamos) as veces\_prestado

from prestamos

join libros on prestamos.ISBN = libros.ISBN

group by libros.titulo;

preguntas que s una entidad