

- 1) Realizar un programa que solicite la carga por teclado de dos números, si el primero es mayor al segundo informar su suma y diferencia, en caso contrario informar el producto y la división del primero respecto al segundo.

```
num1=int(input("Ingrese primer valor:"))
num2=int(input("ingrese segundo valor:"))

if num1>num2:
    suma=num1 + num2
    resta=num1 - num2
    print("La Suma es: ", suma, "\nLa resta es: ", resta, end="****")
else:
    produ=num1 * num2
    divi=num1 / num2
    print("El producto es: ", produ, "\nLa División es: ", divi, end="****")
```

+Str(suma)

```
num1=int(input("Ingrese primer valor:"))
num2=int(input("Ingrese segundo valor:"))
print("La suma es: " , num1+num2)
```

- 2) Se ingresan tres notas de un alumno, si el promedio es mayor o igual a siete mostrar un mensaje "Promocionado".

```
num1=int(input("Ingrese 1° NOTA:"))
num2=int(input("ingrese 2° NOTA:"))
num3=int(input("ingrese 3° NOTA:"))
prome=(num1+num2+num3)/3
if prome >= 7:
    print ("Has Promocionado")
,
```

**3)** Confeccionar un programa que pida por teclado tres notas de un alumno, calcule el promedio e imprima alguno de estos mensajes:

Si el promedio es  $\geq 7$  mostrar "Promocionado".

Si el promedio es  $\geq 4$  y  $< 7$  mostrar "Regular".

Si el promedio es  $< 4$  mostrar "Reprobado".

```
#Opcion 1
nota1=int(input("Ingrese Nota 1:"))
nota2=int(input("Ingrese Nota 2:"))
nota3=int(input("Ingrese Nota 3:"))

promedio=(nota1+nota2+nota3)/3

if promedio >= 7:
    print("Promocionado")
elif promedio >= 4:
    print("Regular")
else:
    print("Reprobado")

#Opcion 2
nota1=int(input("Ingrese Nota 1:"))
nota2=int(input("Ingrese Nota 2:"))
nota3=int(input("Ingrese Nota 3:"))

promedio=(nota1+nota2+nota3)/3

if promedio >= 7:
    print("Promocionado")
else:
    if promedio >= 4:
        print("Regular")
    else:
        print("Reprobado")
```

## Explicación de la lógica del algoritmo

### Ingresamos las tres notas

La instrucción `input( )` en la versión 3 de Python o superior por defecto toma tipos de datos *string* por eso utilizamos la función `int()`

### Calculamos el Promedio

Si quisieramos que el promedio tuviera decimales las variables tendrían que `float()` en lugar de `int()`

### Condicionales

Si se cumple que el promedio es mayor o igual a 7 va a mostrar el mensaje "Promocionado".

Sino se cumple (*else*) puedo poner el `else` y luego otro `if` como en la opción 2 o usar *elif* que es un sentencia de Python que une el `else` con el `if`

Como ya sé que es menor a 7 porque salió por el `else` no es necesario poner en la condición  $< 7$  and  $\geq 4$  no estaría mal pero sería redundante. Lo óptimo es tratar de usar la menor cantidad de líneas de código posible.

Si el promedio es mayor o igual a 4 mostrara "Regular" sino o sea es menor a 4 mostrará "Reprobado"

## Definición de comentarios en el código fuente

Un programa en Python puede definir además del algoritmo propiamente dicho una serie de comentarios en el código fuente que sirvan para aclarar los objetivos de ciertas partes del programa.

Tengamos en cuenta que un programa puede requerir mantenimiento del mismo en el futuro. Cuando hay que implementar cambios es bueno encontrar en el programa comentarios sobre el objetivo de las distintas partes del algoritmo, sobretodo si es complejo.

Existen dos formas de definir comentarios en Python:

Comentarios de una **sola línea**, se emplea el **caracter #**:

```
#definimos tres contadores
```

```
conta1=0
```

```
conta2=0
```

```
conta3=0
```

Todo lo que disponemos después del caracter # no se ejecuta

Comentarios de **varias líneas**: **"""**

```
""" Definimos tres contadores  
    que se muestran si son distintos a cero"""
```

```
conta1=0
```

```
conta2=0
```

```
conta3=0
```

Se deben utilizar tres comillas simples o dobles seguidas al principio y al final del comentario.

# Estructuras repetitivas

Hasta ahora hemos empleado estructuras SECUENCIALES y CONDICIONALES.  
*Estructuras de control de flujo (condicionales y Repetitivas)*

Existe otro tipo de estructuras tan importantes como las anteriores que son las estructuras **REPETITIVAS o BUCLES**

**Una estructura repetitiva permite ejecutar una instrucción o un conjunto de instrucciones varias veces.**

Una ejecución repetitiva de sentencias se caracteriza por:

- La sentencia o las sentencias que están dentro del bucle se repiten.
- El test o prueba de condición antes de cada repetición, que motivará que se repitan o no las instrucciones.

# Estructura repetitiva while

## Funcionamiento

En primer lugar se verifica la **condición**, si la misma resulta **verdadera** se ejecutan las operaciones que indicamos por la rama del Verdadero.

A la rama del verdadero la graficamos en la parte inferior de la condición.

Una línea al final del bloque de repetición la conecta con la parte superior de la estructura repetitiva.

En caso que la condición sea **Falsa** continúa por la rama del Falso y sale de la estructura repetitiva para continuar con la ejecución del algoritmo.

El bloque se repite **MIENTRAS** la condición sea **Verdadera**.

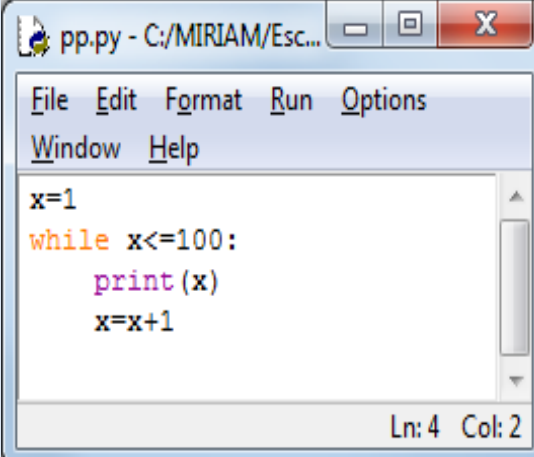
## *While condición:*

Es importante notar que seguido de la palabra clave **while** disponemos la **condición** y finalmente los **dos puntos**.

Todo el código contenido en la estructura repetitiva debe estar indentado

***Realizar un programa que muestre en pantalla los números del 1 al 100***

Iniciamos una variable con el valor 1, luego imprimimos la variable, incrementamos nuevamente la variable y así sucesivamente.



The image shows a screenshot of a Python IDE window titled "pp.py - C:/MIRIAM/Esc...". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The main text area contains the following Python code:

```
x=1
while x<=100:
    print(x)
    x=x+1
```

The status bar at the bottom right indicates "Ln: 4 Col: 2".

# Combinación while if

Algoritmo para calcular la raíz cuadrada controlando que el nro ingresado no sea negativo.  
Se pueden realizar 2 intentos

```
*Python 3.8.2 Shell*
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print ("Programa de cálculo de raíz cuadrada")
numero=int(input("Introduce un número:"))
intentos=0
while numero<0:
    print("No se puede hallar la raíz de número negativo")
    if intentos==2:
        print("Te has quedado sin intentos")
        break;
    numero=int(input("Introduce un número:"))
    if numero<0:
        intentos=intentos+1
if intentos < 2:
    solucion=numero**0.5
    print("La raíz cuadrada de " + str(numero) + " es " + str(solucion))
```

*Para calcular la raíz también podríamos usar `math.sqrt(nro)`. Para lo cual tendríamos que importar el módulo `math`, ya lo veremos en clases posteriores*

Usamos el **Break** para salir de la estructura repetitiva.  
Break simplemente termina el bucle actual y continua con la ejecución de la siguiente instrucción.

**Continue** ignora esa iteración del bucle y sigue con la otra vuelta de bucle. Al aparecer un Continue en Python, este regresa al comienzo del bucle, ignorando todos los estamentos que quedan en la iteración actual del bucle e inicia la siguiente iteración.

**Pass** como su nombre lo indica es una operación nula, o sea que no pasa nada cuando se ejecuta. Se utiliza cuando se requiere por sintaxis una declaración pero no se quiere ejecutar ningún comando o código. También se utiliza en lugares dónde el código irá finalmente, pero no ha sido escrita todavía (utilizándolo como un relleno temporal, hasta que se escriba el código final).

### **Diferencia entre continue y pass**

La diferencia es que tal como lo indica su nombre continue termina la iteración actual, pero continua con el ciclo, volviendo al inicio del bucle en la siguiente iteración.

En cambio pass simplemente no hace nada y pasa a la siguiente instrucción.



1- Desarrollar un programa que permita la carga de 12 valores por teclado y nos muestre posteriormente la suma de los valores ingresados y su promedio. El resultado del promedio es un valor real es decir con coma. Si queremos que el resultado de la división solo retorne la parte entera del promedio debemos utilizar el operador **//**:

**promedio=suma//12**

```
x=1
total=0
while x <=12:
    nume=int(input("Ingrese número: "))
    total=total+nume
    x=x+1
promedio=total/12
print("El promedio es: ", promedio)
```

2- Desarrollar un programa que nos permita ingresar 10 número y nos diga si cada número es par o impar. Para lo cual utilizar el operador % que nos devuelve el resto de la División.

```
x=1
while x <=10:
    nume=int(input("Ingrese número: "))
    if nume%2==0:
        print("El número ingresado es par ", nume)
    else:
        print("El número ingresado es impar ", nume)
    x=x+1
```

**3-** Un postulante a un empleo, realiza un test de capacitación, se obtuvo la siguiente información: cantidad total de preguntas que se le realizaron y la cantidad de preguntas que contestó correctamente. Se pide confeccionar un programa que ingrese los dos datos por teclado e informe el nivel del mismo según el porcentaje de respuestas correctas que ha obtenido, y sabiendo que:

Nivel máximo: Porcentaje  $\geq 90\%$ .

Nivel medio: Porcentaje  $\geq 75\%$  y  $< 90\%$ .

Nivel regular: Porcentaje  $\geq 50\%$  y  $< 75\%$ .

Fuera de nivel: Porcentaje  $< 50\%$ .

**4-** Escribir un programa que solicite ingresar n notas de alumnos y nos informe cuántos tienen notas mayores o iguales a 7 y cuántos menores. Cuando ingresa nota=0 termina la ejecución.

**5-** Ingresan las alturas n personas y se calcula el promedio. Preguntar cuántos datos se ingresarán

**6-** En una empresa trabajan n empleados cuyos sueldos oscilan entre \$100 y \$500, realizar un programa que lea los sueldos que cobra cada empleado e informe cuántos empleados cobran entre \$100 y \$300 y cuántos cobran más de \$300. Además el programa deberá informar el importe total que gasta la empresa en sueldos al personal.

**7-** Se necesita realizar un control de edad de ingreso al Sistema de una empresa. Mientras la edad sea entre 18 y 65 pueden acceder al sistema caso contrario mostrar Mensaje de Acceso Denegado.

**8-** Una planta que fabrica perfiles de hierro posee un lote de n piezas.

Confeccionar un programa que pida ingresar por teclado la cantidad de piezas a procesar y luego ingrese la longitud de cada perfil; sabiendo que la pieza cuya longitud esté comprendida en el rango de 1.20 y 1.30 son aptas. Imprimir por pantalla la cantidad de piezas aptas que hay en el lote.