

... continuación de operadores ...

1. Operadores de asignación

... continuación de Control de flujo

- 2. Bucles o sentencias iterativas
- 3. Bucles While
- 4. Bucles FOR
- 5. Repaso y prácticas



1. Operadores de asignación

```
In [1]: a = 0
```

```
In [2]: a += 1 # suma en asignación  
a
```

equivale a `a = a + 1`

```
Out[2]: 1
```

`a = 10`

```
In [8]: a -= 10 # resta en asignación
```

`a`

equivale a `a = a - 10`

```
Out[8]: 0
```

```
In [9]:
```

```
a = 5
```

```
a *= 2 # producto en asignación
```

`a`

equivale a `a = a * 2`

```
Out[9]: 10
```

```
In [10]: a /= 2 # división en asignación  
a
```

```
Out[10]: 5.0
```

```
In [11]: a %= 2 # módulo en asignación
```

```
In [12]: a
```

```
Out[12]: 1.0
```

```
In [13]: a **= 10
```

```
In [14]: a
```

```
Out[14]: 1.0
```

```
In [ ]: a = 5
```

Práctica P01

Calcula mentalmente y comprueba los resultados parciales y el resultado final.

$a = 7$

$a += 3$

$a /= 5$

$a **= 4$

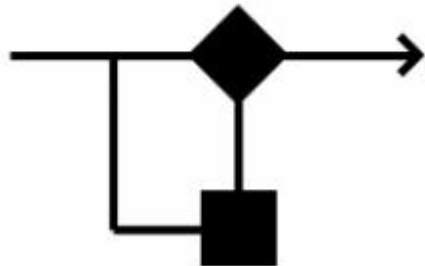
$a -= 20$

...continuando con Control de Flujo

Iteraciones

- Bucles while
- Bucles for

Iterativas: Para
repetir un bloque de
instrucciones



2. Bucles o sentencias iterativas

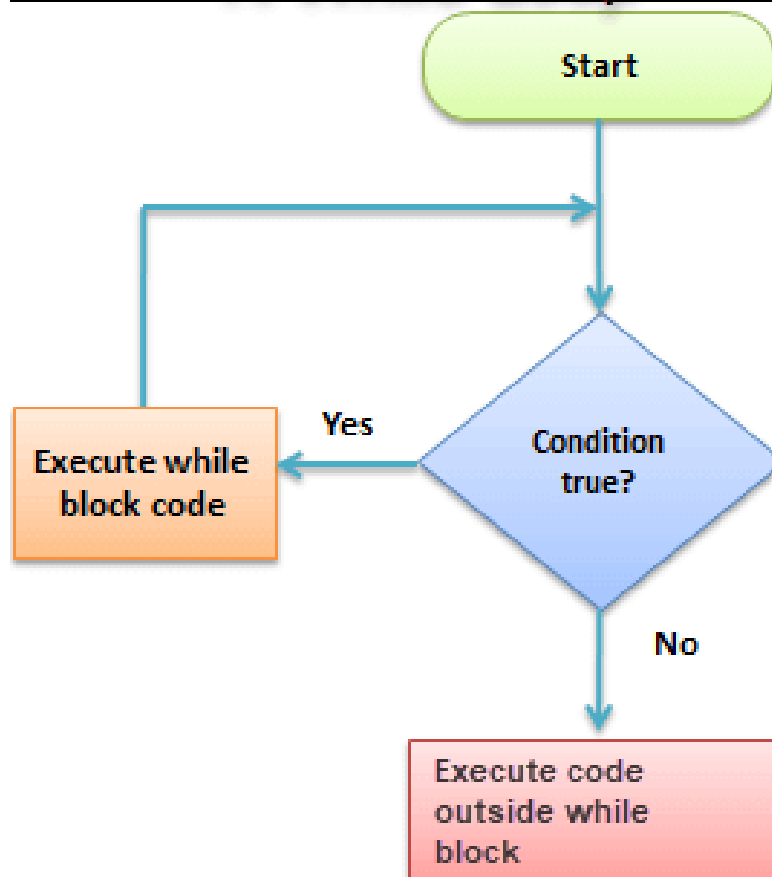
¿Qué es la iteración?

- Iterar es realizar una determinada acción varias veces
- Cada vez que se repite la acción se denomina iteración

data	Proveïdor	Import	Concepte
17/04/2018	SOLA Pintura	49,00	pintura blanca
19/04/2018	BAUHAUS	32,89	pintures i coles
20/04/2018	BAUHAUS	208,05	campana + estanteria
20/04/2018	SOLA Pintura	12,95	pintura esmalt pissarra

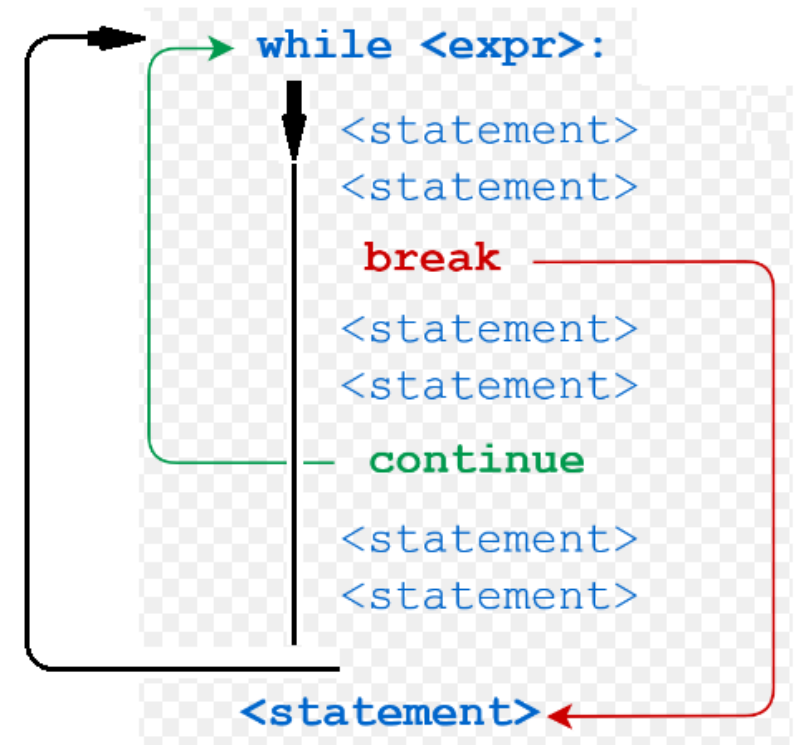
3. Bucles While (while = mientras)

R While Loop



```
c = 0
while c < 10 :
    c+=1
    print("c = ",c)

print("completado")
```



while (while = mientras)

```
c = 0
while c <=5 :
    c+=1
    print("c vale ",c)

print("completado")
```

Mientras c sea menor o igual que 5 hacer

- $c = c + 1$
- imprime el valor de c

Imprime "completado"

La magia de las iteraciones

Para encontrar un elemento, el ordenador debe recorrer los registros y compararlos hasta encontrar el que se busca

Importancia crucial del **valor inicial** (se establece antes de empezar a iterar)

E importancia también de la **condición final** que detiene la iteración

```
In [2]: c = 0
while c <= 5:
    c+=1
    print("c vale",c)
else:
    print("Se ha completado toda la iteración y c vale",c)
```

c vale 1

c vale 2

c vale 3

c vale 4

c vale 5

c vale 6

Se ha completado toda la iteración y c vale 6

cláusula *break* = rompe el bucle

```
In [3]: c = 0
while c <= 5:
    c+=1
    if (c==2):
        print("Rompe el bucle cuando c vale",c)
        break
    print("c vale",c)
else:
    print("Se ha completado toda la iteración y c vale",c)
```

c vale 1

Rompe el bucle cuando c vale 2

cláusula *continue* = continúa siguiente

```
c = 0
while c <= 5:
    c+=1
    if (c==4):
        # print("Continuamos con la siguiente iteración",c)
        continue
    print("c vale",c)
else:
    print("Se ha completado toda la iteración y c vale",c)
```

Cuidado con los bucles infinitos !!!

Dejan saturado el procesador

EXEMPLE : Calcula la suma dels termes anteriors a un nombre base que demana per pantalla (Exemple 4 -> 4 + 3 + 2 + 1 resultat = 10)

```
print ("DADO UN NÚMERO BASE CALCULAR LA SUMA DEL ")
print (" NÚMERO MÁS TODOS LOS ANTERIORES POSITIVOS")
base = input("numero base : ")
base = int(base)
suma = 0
while base > 0 :
    print ("dentro while ", base , suma)
    suma = suma + base
    base = base -1

print ("La suma es : ", suma)
```

Práctica P02-1: while

variables

```
char opcion
char semaforo # considera iguales mayúsculas y minúsculas
```

inicio

```
ESCRIBIR : "Opciones Semaforo "
ESCRIBIR : " R.Poner en Rojo "
ESCRIBIR : " V.Poner en Verde"
ESCRIBIR : " ?.Interrogar Semáforo"
ESCRIBIR : " X.Salir del Programa"
LEER opcion
MIENTRAS (opcion != "X")
    SI (opcion == "R")
        ESCRIBIR "Pasando a Rojo... "
        semaforo <- "Rojo"
    SINO
        SI (opcion == "V")
            ESCRIBIR "Pasando a Verde... "

            semaforo <- "Verde"
        SINO
            SI (opcion == "?")
                ESCRIBIR semaforo
            FIN-SI
        FIN-SI
    FIN-SI
    LEER opcion
FIN_MIENTRAS
```

fin



Haz el programa
de este algoritmo

Práctica P02-2: while

1. Tenemos la lista `marcas = ["toyota", "renault", "mercedes"]`

Realiza un programa que muestre el siguiente Menú de opciones

(1. Consultar 2.Crear Nuevo, 3.Eliminar, 0.Salir)

y pida la `opcion`

Ayuda

`list.append()`

`list.remove()`

- Si `opcion == 1`, muestra los elementos de la lista
- Si `opcion == 2`, pide marca y la guarda en la lista
- Si `opcion == 3` pide marca y la elimina de la lista
- Si `opcion == 0` termina el programa
- En otro caso muestra en mensaje “**opción errónea**” y vuelve a pedir la opción

Avanzado, que no duplique marcas de coches, ni falle al intentar borrar un elemento que no existe. Para ello comprueba si existe o no ese elemento en la lista antes de proceder a añadir o quitar.

Ayuda

If elemento in lista :

Práctica P02-3: while

2 Número mayor (while)

Escriba un programa que pida dos números enteros. El programa pedirá de nuevo el segundo número mientras no sea mayor que el primero. El programa terminará escribiendo los dos números.

```
Escriba un número: 6
Escriba un número mayor que 6: 6
6 no es mayor que 6. Inténtelo de nuevo: 1
1 no es mayor que 6. Inténtelo de nuevo: 8

Los números que ha escrito son 6 y 8.
Programa terminado
```

3. Números positivos

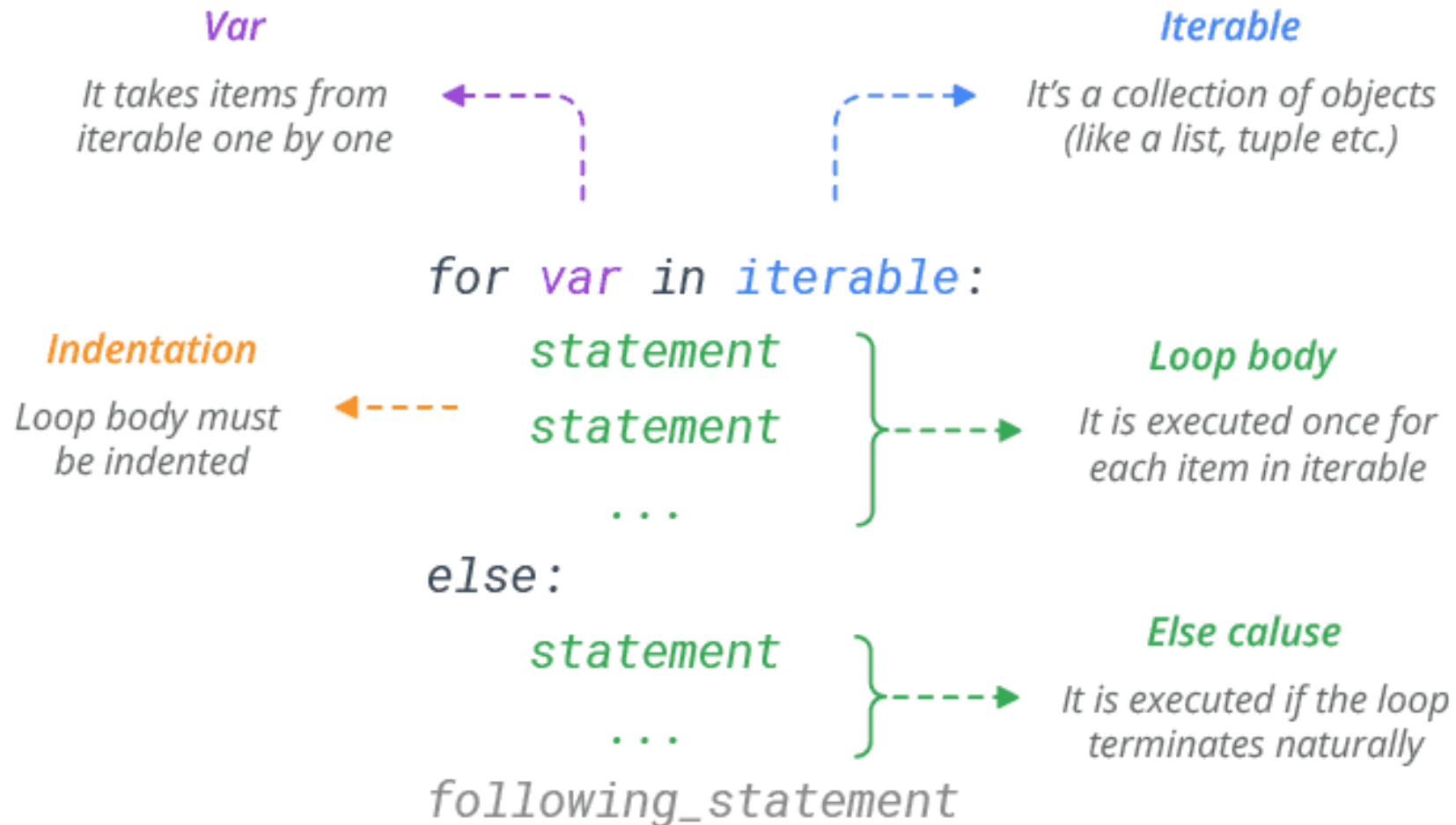
Escriba un programa que pida la cantidad de números positivos que se tienen que escribir y a continuación pida números hasta que se haya escrito la cantidad de números positivos indicada.

```
Escriba la cantidad de números positivos a escribir: 3

Escriba un número: 10
Escriba otro número: 2
Escriba otro número: -5
Escriba otro número: 2

Ha escrito 4 números, 3 de ellos positivos.
Programa terminado
```

4. Bucles FOR (for = para)



Bucles FOR (for = para)

El For de Python está pensado para colecciones

```
dias = ["lunes", "martes", "miércoles", "jueves", "viernes", "sábado", "domingo"]
indice = 0
while indice < len(dias):
    print(dias[indice])
    indice+=1
```

Simplifica mucho el recorrido de listas

lunes
martes
miércoles
jueves
viernes
sábado
domingo

```
dias = ["lunes", "martes", "miércoles", "jueves", "viernes", "sábado", "domingo"]

for dia in dias: # Para [variable] en [lista]
    print(dia)
```

lunes
martes
miércoles
jueves
viernes
sábado
domingo


```
numeros = [1,2,3,4,5,6,7,8,9,10]
for numero in numeros :
    print(numero*10)
```

10
20
30
40
50
60
70
80
90
100



Python For Loop - Essentials...

for in seq

```
for theBird in birds:
    print(theBird )
else:
    print("...")
```

```
for iter in vowels:
    print("char:", iter)
```

```
for iter in range(0, 3):
    print("iter: %d" % (iter))
```

```
In [3]: for numero in numeros:
        numero *= 10
```

```
In [4]: numeros
```

```
Out[4]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
In [5]: indice = 0
        numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
        for numero in numeros:
            numeros[indice] *= 10
            indice+=1
        numeros
```

```
Out[5]: [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

```
In [6]: numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
        for indice,numero in enumerate(numeros):
            numeros[indice] *= 10
        numeros
```

```
Out[6]: [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

```
cadena = "Hola amigos"
for character in cadena:
    print(character)
```

H
o
l
a

a
m
i
g
o
s

```
for i,c in enumerate(cadena):
    cadena[i] = "*"
-----
```

```
TypeError                                Traceback (most recent call last)
<ipython-input-9-8ba888c46579> in <module>()
      1 for i,c in enumerate(cadena):
----> 2     cadena[i] = "*"

TypeError: 'str' object does not support item assignment
```

Práctica P03: for

1. Sea la colección de alumnos alus = ["ana", "joan", "rosa", "antonio"], pide una nota para cada uno y guárdalas en una lista.
2. Imprime la lista de todas las personas con una columna para el nombre y otra para la calificación

FOR sin colección

Funcionamiento del **FOR** similar al de otros lenguajes.

```
for i in range(10):  
    print(i)
```

0
1
2
3
4
5
6
7
8
9

La instrucción **range** admite
(desde, hasta, incremento)

```
range(10)
```

```
range(0, 10)
```

```
for i in [0,1,2,3,4,5,6,7,9]:  
    print(i)
```

0
1
2
3
4
5
6
7
9

Práctica P04: tabla verdad

3) **Avanzado** : Realiza en programa para que imprima la tabla de verdad de relaciones 2 términos y 1 operador solicitados por pantalla.

Ejemplo de entrada :

Introduzca 2 letras separadas por comas : p,q

Introduzca un operador (or / and) : and

Salida :	p	q	p and q
	0	0	0
	0	1	0
	1	0	0
	1	1	1

Pista : `cadena.split(",")`
convierte una cadena en
una lista, excluyendo los
separadores.

Soluciones ejercicios

#P02 1. Pedir un número mayor

```
num1 = int (input("Escriba un número: ") )
print("Escriba un número mayor que " + str(num1)+ ":")
num2 = int (input())

while num2 <= num1 :
    print(num2, "no és mayor que", num1, ".Intentelo de nuevo : ")
    num2 = int(input())

print("Los números que ha escrito son ", num1 , "y", num2)
print("Programa terminado")
```

#Solución P02-1 : Semáforo

```
opcion = ""
semaforo = "Rojo"

#----- Inicio
print("Opciones Semaforo ")
print("\t R.Poner en Rojo ")
print("\t V.Poner en Verde ")
print("\t ?.Interrogar Semáforo")
print("\t X.Salir del Programa")

opcion = input ("Opcion :")

while opcion != "X" :
    if opcion == "R" :
        print("pasando a Rojo ...")
        semaforo = "Rojo"
    elif opcion == "V" :
        print("pasando a Verde ...")
        semaforo = "Verde"
    elif opcion == "?" :
        print ("semaforo : " , semaforo)
    opcion = input ("Opcion :")
```


#Solución P02 1. Marcas de coches

```
marcas =["toyota", "renault", "mercedes"]

print("<1. Consultar    2.Crear Nuevo,    3.Eliminar,    0.Salir>")
while True :
    opcion = input("OPCION : ")
    if opcion.isdigit() :
        opcion = int(opcion)
        if opcion == 1 :
            print(marcas)
        elif opcion == 2 or opcion == 3 :
            marca = input("marca :").lower()
            if opcion == 2 :
                if marca not in marcas :
                    marcas.append(marca)
            else:
                if marca in marcas :
                    marcas.remove(marca)
        elif opcion == 0 :
            break
        else :
            print("opcion errónea")
    else :
        print("opcion errónea")
```

#Solución P02-3.2 : Números positivos

```
print("Escriba la cantidad de números positivos a escribir : ")
nums = int(input())
i=0
p=0
while i < nums :
    if i == 0 :
        print ("Escriba un número: ")
    else :
        print("Escriba otro número")
    n = int(input())
    if n >= 0 : p+=1
    i+=1
print("Ha escrito ", nums, "numeros, ", p , "de ellos son
positivos")
print ("programa terminado")
```

#Solución P03 : for

```
alus = ["ana", "joan", "rosa", "antonio"]
notas = []
for alu in alus :
    nota = input ("Entre la nota de " + alu)
    notas.append(nota)

for i,alu in enumerate(alus) :
    print (alu, notas[i])
```

#Solución P04 : tabla de verdad con dos letras y un operador

```
lletres = input ("Entri dues lletres separades per comes : ")
opera = input ("Entri un operador (and, or): ")

columnes = lletres.split(",")

print ( columnes[0], columnes[1], columnes[0] + " " + opera + " " + columnes[1])
print ( "- - - - -")
for i in range (0, 2) :
    for j in range (0, 2) :

        if opera == "and" : # Calcula operació
            calc = i and j
        else :
            calc = i or j

        print ( i, j, " ", calc)
```

Resultado :

```
Entri dues lletres separades per comes : p,q
Entri un operador (and, or): and
p q p and q
- - - - -
0 0    0
0 1    0
1 0    0
1 1    1
```

5. Repaso

```
n = 0 # Asignación de 0 en n
while n < 10: # Expresión relacional n < 10, que devuelve True
    if (n % 2) == 0: # Expresión aritmética y expresión relacional
        print(n, 'es un número par')
    else:
        print(n, 'es un número impar')
    n += 1 # expresión aritmética n = n + 1 equivalente a operación en asignación n+=1
```

Y todo lo aprendido

Bases de la programación

- Instrucciones básicas
- Variables y tipos de datos
- Operadores y expresiones
- Sentencias condicionales
- Sentencias iterativas

Practicas P05

- 1) Realiza un programa que sume todos los números enteros pares desde el 0 hasta el 100 (excluido)

Sugerencia : Puedes utilizar la funciones *sum()* y *range()* para hacerlo más fácil. El tercer parámetro en la función *range(inicio, fin, salto)* indica un salto de números, pruébalo.

- 2) Control Parking. Crea una lista vacía llamada `cars[]`.

Muestra el siguiente menú y pide la opción

“ opcion (1.Entra coche, 2.Sale coche, 3.Listar coches, 0.Salir): “

- Si la **opción** es 1 = pide la matrícula y guárdalo en la lista
`cars.append (matricula)`
- Si la **opción** es 2 = pide la matrícula y elimínala de la lista.
- Si la **opción** es 3 = Muestra la lista de coches que no han salido.

Sigue pidiendo datos hasta que la opción introducida sea 0.

algoritmo **control_parking**

variables

caracter opcion
cadena matricula
lista cars []

inicio

opcion = ""

mientras (opcion <> "0")

 escribir "Opcion (1.Entra coche, 2.Sale coche, 3.Listar coches, 0.Salir) : "

 leer (opcion)

 #----- Analiza opcion : 0 salir

 si (opcion == "0")

 interrumpir

 fin_si

 #----- Analiza opcion : 1 entra , 2 sale

 si (opcion == "1" or opcion == "2")

 escribir "Entre matrícula: "

 leer (matrícula)

 si (opcion == "1")

 cars = cars + matrícula

 sino

 cars = cars - matrícula

 fin_si

 fin_si

 #----- Analiza opcion : 3 Listar

 si (opcion == "3")

 imprimir cars

 fin_si

fin_mientras

fin

3) **Avanzado**: Realiza en programa para que imprima la tabla de verdad de 3 términos y 2 operadores solicitados por pantalla en forma de cadena y usando “,” como separador.

Ejemplo de entrada :

Introduzca 3 letras separadas por comas : p, q, r

Introduzca 2 operadores separados por comas(or / and) : or, and

p	q	r		p	or	q	and	r
0	0	0	=	0				
0	0	1	=	0				
0	1	0	=	0				
0	1	1	=	1				
1	0	0	=	0				
1	0	1	=	1				
1	1	0	=	0				
1	1	1	=	1				

Soluciones P05– Ejercicio 1

```
suma = 0
for i in range (0, 100, 2) :
    suma += i

print ("La suma es ", suma)
```

```
print ("suma:",
      sum(range(0,100,2)))
```

Solución P05– Ejercicio 2

```
cars =[]

print ("MENU PARKING COCHES 1.Entra, 2.Sale, 3.Lista, 0.Salir")

bSalir = False
while bSalir == False :
    opcion = int(input("Introduce opcion:"))
    if opcion == 0 :
        bSalir = True
        continue
    elif opcion == 3 :
        print (cars)
    else :
        matricula = input ("Introduzca la matrícula: ")
        if opcion == 1:
            cars.append(matricula)
        else :
            cars.remove(matricula)
```

```
cars =[]

print ("MENU PARKING COCHES 1.Entra, 2.Sale, 3.Lista,
0.Salir")

bSalir = False
while bSalir == False :
    opcion = int(input("Introduce opcion:"))
    if opcion == 0 :
        bSalir = True
        continue
    elif opcion == 3 :
        print (cars)
    else :
        matricula = input ("Introduzca la matrícula: ")
        if opcion == 1:
            cars.append(matricula)
        else :
            cars.remove(matricula)

#-----
```

Texto para copiar

Solución P05 – Ejercicio 3 – Tabla de Verdad

```
lletres = input ("Entri 3 lletres separades per comes : ")
operadors = input ("Entri dos operadors separats per comes (and, or): ")

columnes = lletres.split(",")
opers = operadors.split(",")

print ( columnes[0], columnes[1], columnes[2], columnes[0] + " " + opers[0] + " " +
        columnes[1]+ " " + opers[1] + " " + columnes[2])
print ( "- - - -----")

for i in range (0, 2) :
    for j in range (0, 2) :
        for k in range (0, 2) :
            if opers[0] == "and" : # Calcula operació
                calc = i and j
            else :
                calc = i or j
            if opers[1] == "and" : # Calcula operació
                calc = calc and k
            else :
                calc = calc or k

            print ( i, j,k, " ", calc)
```

Cumple las leyes de precedencia de los operandos lógicos ?.

Arréglalo por favor

Texto para copiar

```
lletres = input ("Entri 3 lletres separades per comes : ")
operadors = input ("Entri dos operadors separats per comes (and, or): ")

columnes = lletres.split(",")
opers = operadors.split(",")

print ( columnes[0], columnes[1], columnes[2], "(" + columnes[0] + " " + opers[0] + " " +
        columnes[1]+ ") " + opers[1] + " " + columnes[2])
print ( "- - - -----")

for i in range (0, 2) :
    for j in range (0, 2) :
        for k in range (0, 2) :
            if opers[0] == "and" : # Calcula operació
                calc = i and j
            else :
                calc = i or j
            if opers[1] == "and" : # Calcula operació
                calc = calc and k
            else :
                calc = calc or k

            print ( i, j,k, " ", calc)
```

Solución P05 – Tablas de multiplicar

```
#----- FUNCIONES

def imprimir_tabla (num) :
    print ("--- Tabla del", num , "---")
    for i in range (1, 11) :
        print ( num , " x " , i , " = " , num * i)
    print ("0-----")

#----- Proceso principal

bSalir = False

print ("MENU OPCIONES  quit, all, o número de tabla a imprimir")
while bSalir == False :
    opcion = input("Introduzca opción :")
    if opcion == "quit" :
        bSalir = True
    elif opcion == "all" :
        for i in range (1, 10) :
            imprimir_tabla (i)
    else :
        numero = int (opcion)
        if numero >= 0 and numero <= 100 :
            imprimir_tabla (numero)

#----- fin del bucle
```

Texto para copiar

```
#----- FUNCIONES
```

```
def imprimir_tabla (num) :  
    print ("--- Tabla del", num , "--")  
    for i in range (0, 10) :  
        print ( num , " x " , i , " = " , num * i)  
    print ("-----")
```

```
#----- Proceso principal
```

```
bSalir = False
```

```
print ("MENU OPCIONES  quit, all, o número de tabla a imprimir")  
while bSalir == False :  
    opcion = input("Introduzca opción :")  
    if opcion == "quit" :  
        bSalir = True  
    elif opcion == "all" :  
        for i in range (1, 10) :  
            imprimir_tabla (i)  
    else :  
        numero = int (opcion)  
        if numero >= 0 and numero <= 100 :  
            imprimir_tabla (numero)  
#----- fin del bucle
```