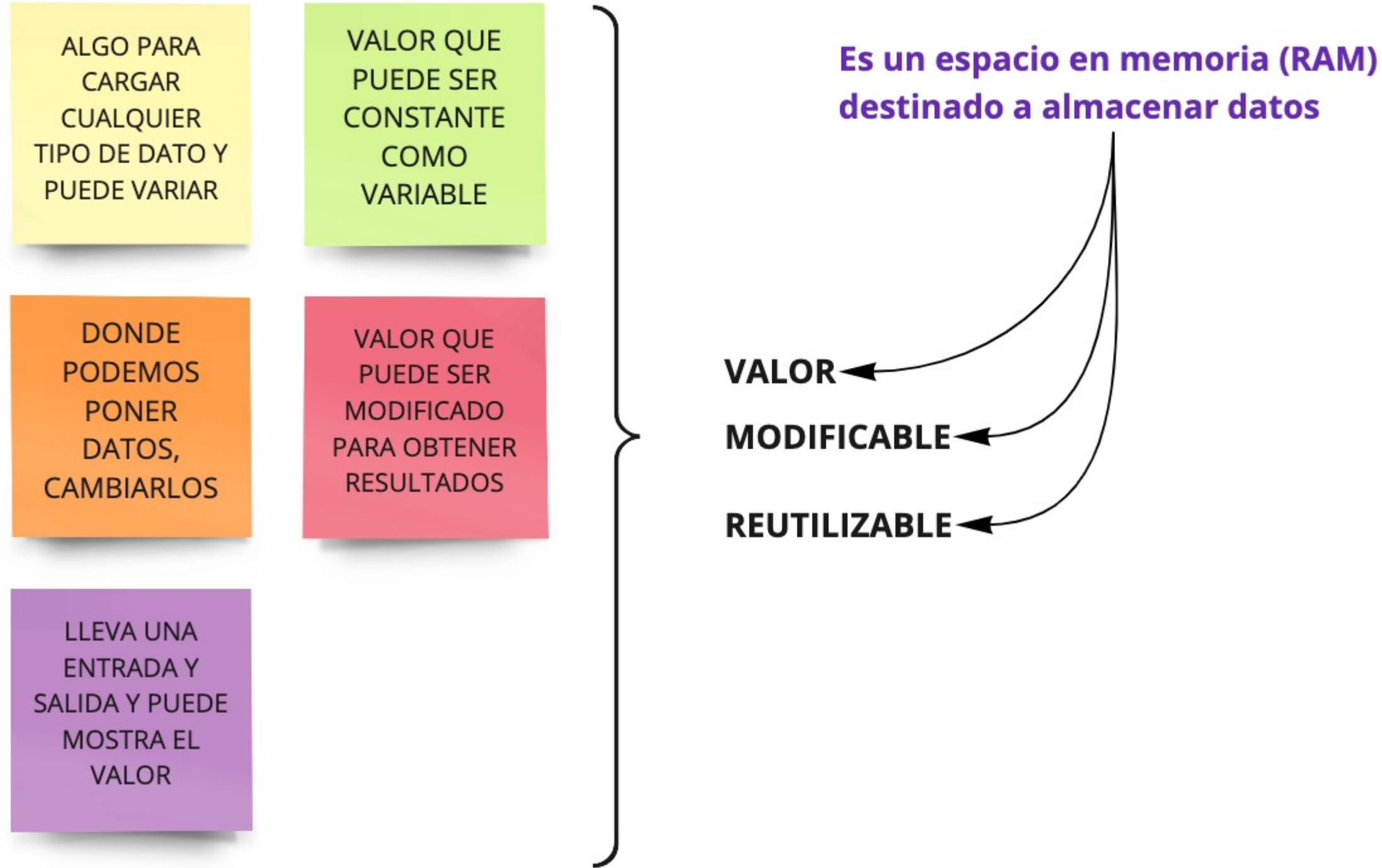
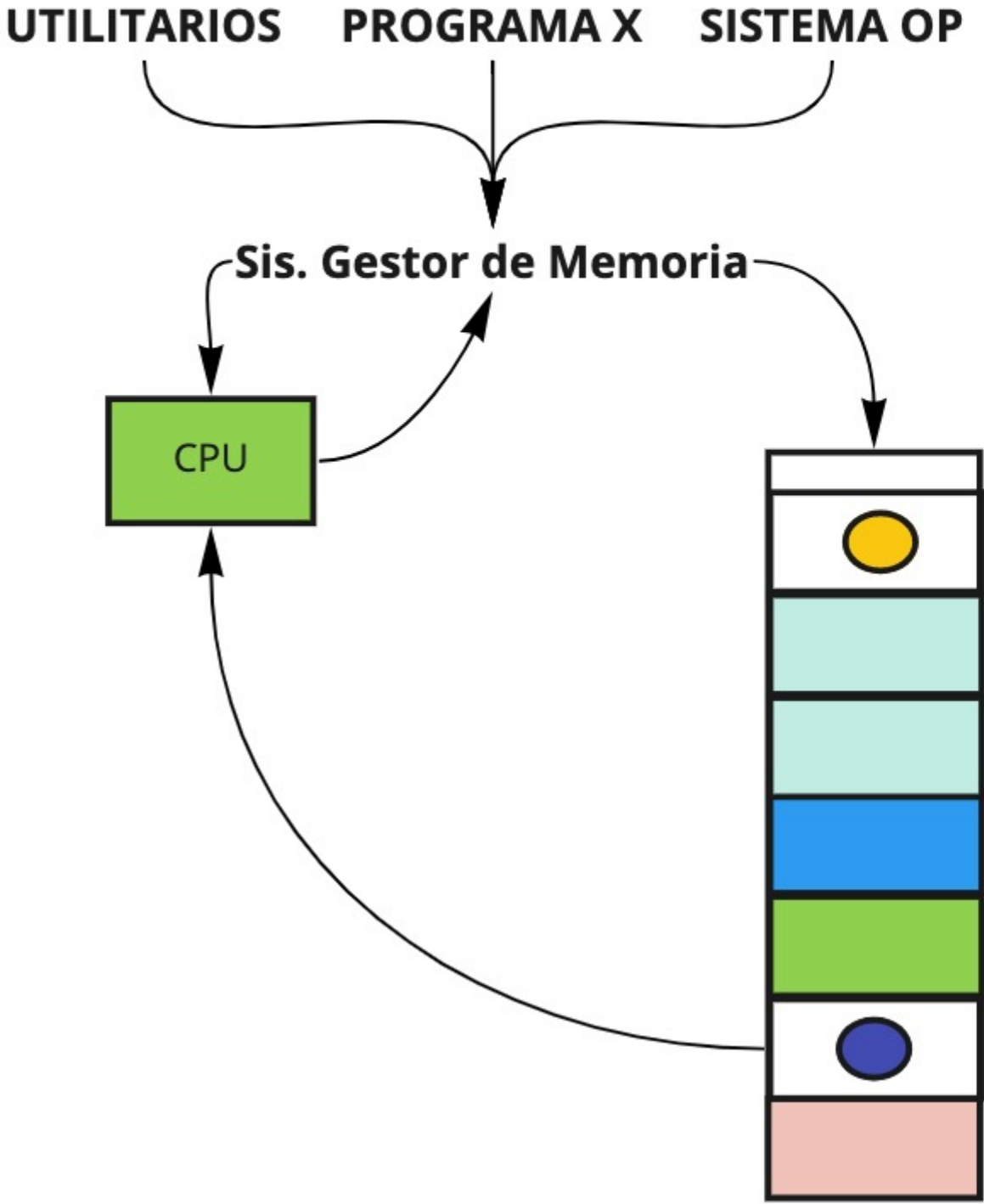


VARIABLES DEFINICIÓN



COMPORTAMIENTO PC



Palabras reservadas

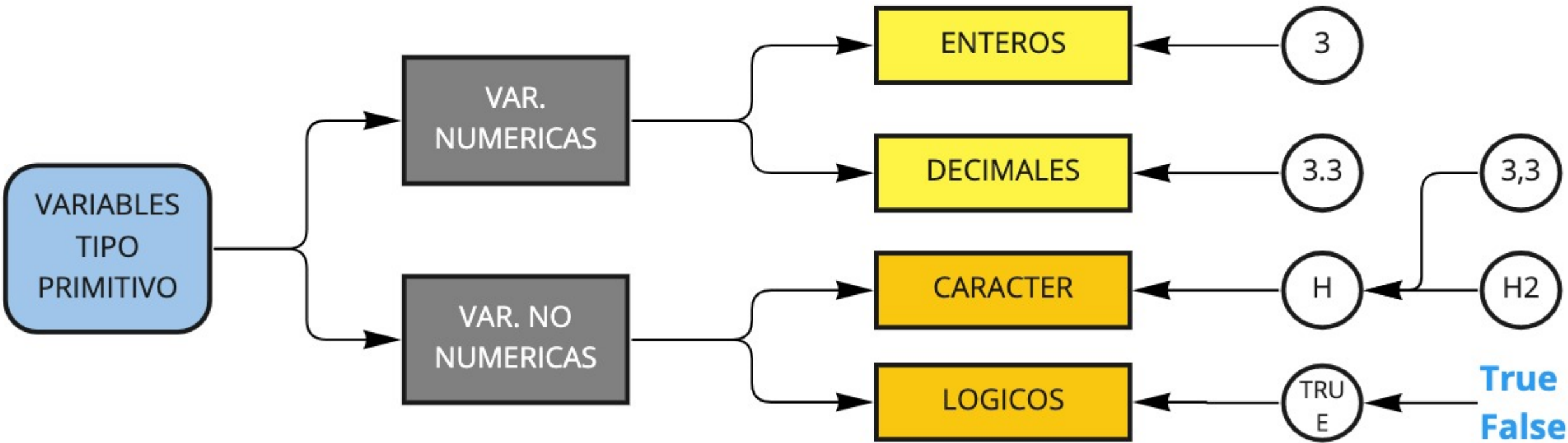
False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

DECLARAR VARIABLES EN PYTHON

numCasa = 435

Nombre Var Operador Valor

ESQUEMA DE TIPOS DE DATO Y VARIABLE



SANA PRÁCTICA DEF. VARIABLES

Representativa

X1 —porque despues olvidamos el propósito de la variable ✗
X1 - X2 - Xn —porque se confunde la variable ✗
sumPasajeBus ✓

No empiezan con Números

2Carros ✗
carros2 ✓

uso de estadares

Camel

Pascal

_numero_casa
Numero_Casa
numeroCasa
NUMEROCASA

Los nombre de variables no puedes ser palabras reservada ✗

inicializar variables

var = 0
var = '' ✓

OPERADORES LOGICOS

AND
OR
NOT

TRUE
FALSE

OP1	OP2	RESULTADO
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

AND
CONJUNCIONES

OP1	OP2	RESULTADO
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

OR
DISYUNCIONES

OP1	OP2	RESULTADO	NEGADO
TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	TRUE	FALSE
FALSE	TRUE	TRUE	FALSE
FALSE	FALSE	FALSE	TRUE

OR — NOT

OP1	OP2	RESULTADO	NEGADO
TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	TRUE

AND — NOT

Tablas de verdad

IF - ELIF - ELSE



Son decisiones basadas en condiciones que se deben tomar a nivel código.

Esas desiciones se aplican con operadores lógicos.

```
if op1 == 2:  
    print ('Hola')
```

```
if op1 == 2 and op2 < 5:  
    print ('Hola')
```

op1 = 2
op2 = 7



```
if op1 == 2 and op2 < 5:  
    print ('Hola')
```

op1 = 2
op2 = 5



```
if op1 == 2 and op2 < 5:  
    print ('Hola')
```

op1 = 2
op2 = 3



```
if op1 == 2 or op2 < 5:  
    print ('Hola')
```

op1 = 2
op2 = 7



```
if op1 == 2 or op2 < 5:  
    print ('Hola')
```

op1 = 2
op2 = 5



```
if op1 == 2 or op2 < 5:  
    print ('Hola')
```

op1 = 2
op2 = 3



```
if not (op1 == 2 or op2 < 5):  
    print ('Hola')
```

op1 = 2
op2 = 3



```
if not (op1 == 2) or op2 < 5:  
    print ('Hola')
```

op1 = 2
op2 = 3



```
if op1 == 2 or op2 < 5:  
    print ('Hola')  
else:  
    print ('Adios')
```

op1 = 1
op2 = 5



```
if op1 == 2 or op2 < 5:  
    print ('Hola')  
elif op1 < 3 and op2 <= 5:  
    print ('aca estoy')  
else:  
    print ('Adios')
```

op1 = 1
op2 = 5



```
if op1 == 2 or op2 < 5:  
    print ('Hola')  
elif op1 < 3 and op2 <= 5:  
    print ('aca estoy')  
else:  
    print ('Adios')
```

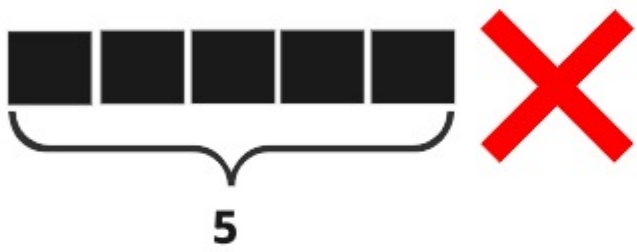
op1 = 7
op2 = 5



CICLOS

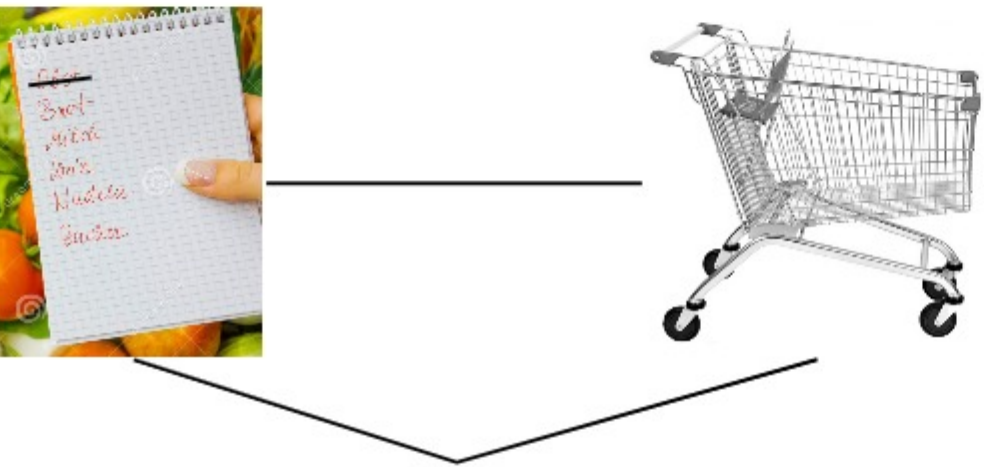
while (mientras)

Se usa cuando no se conoce cuantas iteracciones se van a ejecutar.



while condicionLogica :
 print (....)

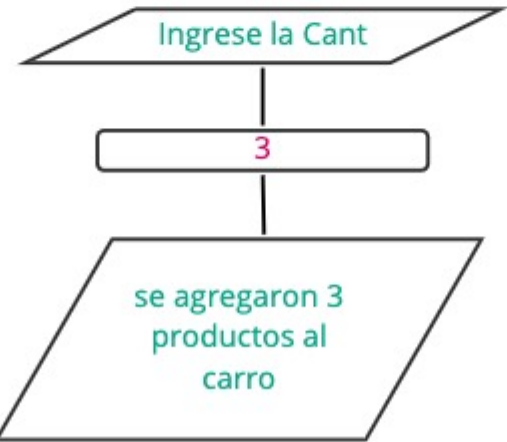
mientras se cumpla la condición haga



Productos = 10
cant = 0

```
while productos > 0 :  
    print ('ingrese cantidad')  
    cant = int(input())  
    if cant > 1:  
        print ('se agregaron', cant, 'productos al carro')  
        productos = productos - cant  
    else:  
        print ('No hay suficientes productos')
```

Producto	cant
10	0
7	3
3	5
0	3



for (para)

Se usa cuando se conoce cuantas iteracciones se van a ejecutar.

