

# Algoritmos

## Algoritmo.

Condicionales

Ciclos - (for, while)

Listas

Matrices

Diccionarios

Archivos

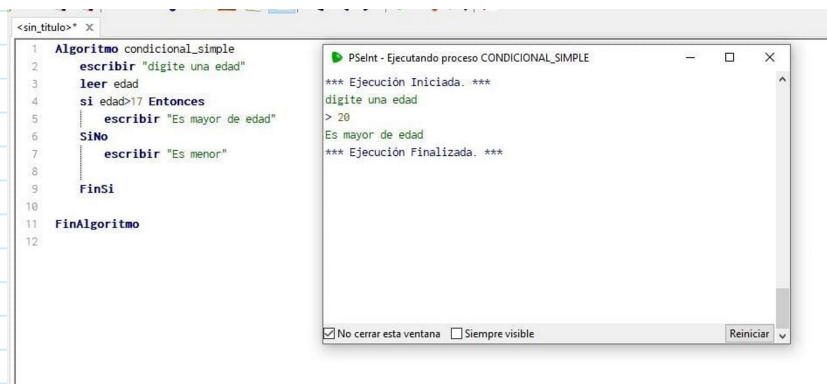
**Que es un algoritmo:** término "algoritmo" también se utiliza en el contexto de la informática y la tecnología. En este sentido, un algoritmo se refiere a un conjunto preciso de instrucciones o reglas que se utilizan para resolver un problema o realizar una tarea en un programa de computadora.

Los algoritmos en el ámbito de la informática son fundamentales, ya que son la base para el desarrollo de software y la resolución de problemas computacionales. Pueden abordar desde tareas simples, como ordenar una lista de números, hasta procesos complejos, como el análisis de grandes conjuntos de datos.

En resumen, en el contexto de la informática, un algoritmo es una secuencia lógica y precisa de pasos que se utilizan para resolver problemas o realizar operaciones en un entorno digital.

## Ejercicios en pSeint.

- **Hacer un algoritmo que permita saber si una persona es mayor de edad.**



```
<sin_titulo>* X
1 Algoritmo condicional_simple
2   escribir "digite una edad"
3   leer edad
4   si edad>17 Entonces
5       escribir "Es mayor de edad"
6   SiNo
7       escribir "Es menor"
8   FinSi
9
10
11 FinAlgoritmo
12
```

PSeint - Ejecutando proceso CONDICIONAL\_SIMPLE

\*\*\* Ejecución Iniciada. \*\*\*

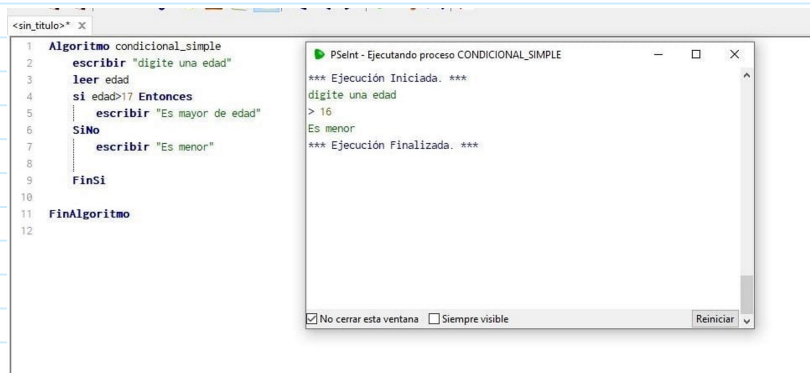
digite una edad

> 20

Es mayor de edad

\*\*\* Ejecución Finalizada. \*\*\*

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar



```
<sin_titulo>* X
1 Algoritmo condicional_simple
2   escribir "digite una edad"
3   leer edad
4   si edad>17 Entonces
5       escribir "Es mayor de edad"
6   SiNo
7       escribir "Es menor"
8   FinSi
9
10
11 FinAlgoritmo
12
```

PSeint - Ejecutando proceso CONDICIONAL\_SIMPLE

\*\*\* Ejecución Iniciada. \*\*\*

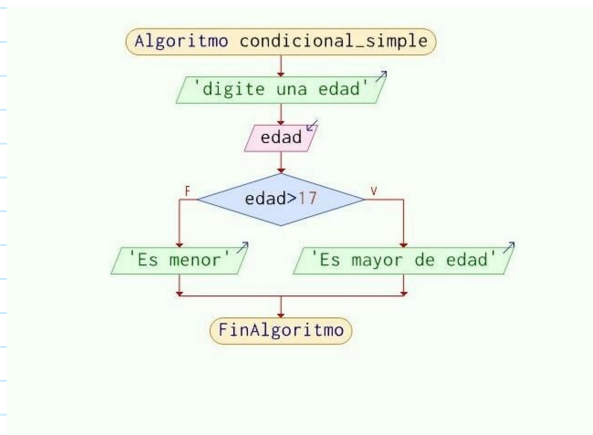
digite una edad

> 16

Es menor

\*\*\* Ejecución Finalizada. \*\*\*

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar



➤ **Hacer un algoritmo que permita saber si un numero es par o impar.**

```

1 Algoritmo condicional_simple
2   escribir "escribir numero"
3   leer num
4   si num %2 == 0 Entonces
5     escribir "su numero es par"
6   sino
7     escribir "No es par"
8   FinSi
9
10
11
12
13 FinAlgoritmo
14

```

PSeInt - Ejecutando proceso CONDICIONAL\_SIMPLE

```

*** Ejecución Iniciada. ***
escribir numero
> 8
su numero es par
*** Ejecución Finalizada. ***

```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

```

1 Algoritmo condicional_simple
2   escribir "escribir numero"
3   leer num
4   si num %2 == 0 Entonces
5     escribir "su numero es par"
6   sino
7     escribir "No es par"
8   FinSi
9
10
11
12
13 FinAlgoritmo
14

```

PSeInt - Ejecutando proceso CONDICIONAL\_SIMPLE

```

*** Ejecución Iniciada. ***
escribir numero
> 5
No es par
*** Ejecución Finalizada. ***

```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

**15/05/2024**

## **VARIABLE.**

Es un nombre que se utiliza para hacer referencia a un valor almacenado en la memoria de la computadora.

En Python, las variables se crean mediante asignación, es decir, se asigna un valor a un nombre utilizando el operador de asignación.

## Principales tipos de variables en Python:

Enteros("int")

Flotantes ('Float')

Cadenas de texto ('str')

Booleanos ('bool')

Listas('list')

Tuplas ('tuple') 2 métodos (.index) y (.count)

Conjuntos ('set')

En Python, una variable puede cambiar de tipo durante la ejecución del programa. Por ejemplo, puedes asignar un entero a una variable y luego asignar una cadena de texto a la misma variable en una línea posterior.



## Operador de concatenación de cadenas de caracteres.

La forma más simple de concatenar dos cadenas en Python es utilizando el operador de concatenación +

## Operadores lógicos o booleanos.

A la hora de operar con valores booleanos, tenemos a nuestra disposición los operadores **and**, **or** y **not**.

OPERADOR	DESCRIPCIÓN	USO
and	Devuelve True si ambos operandos son True	a and b
or	Devuelve True si alguno de los operandos es True	a or b
not	Devuelve True si alguno de los operandos False	not a

**Operador and:** El operador and evalúa si el valor a la izquierda y el de la derecha son True, y en el caso de ser cierto, devuelve True. Si uno de los dos valores es False, el resultado será False.

**Operador or:** El operador or devuelve True cuando al menos uno de los elementos es igual a True. Es decir, evalúa si el valor a la izquierda o el de la derecha son True.

**Operador not:** Y por último tenemos el operador not, que simplemente invierte True por False y False por True.

### Operadores de comparación.

Los operadores de comparación se utilizan, como su nombre indica, para comparar dos o más valores. El resultado de estos operadores siempre es True o False.

Operador	Descripción
>	Mayor que. <code>True</code> si el operando de la izquierda es estrictamente mayor que el de la derecha; <code>False</code> en caso contrario.
>=	Mayor o igual que. <code>True</code> si el operando de la izquierda es mayor o igual que el de la derecha; <code>False</code> en caso contrario.
<	Menor que. <code>True</code> si el operando de la izquierda es estrictamente menor que el de la derecha; <code>False</code> en caso contrario.
<=	Menor o igual que. <code>True</code> si el operando de la izquierda es menor o igual que el de la derecha; <code>False</code> en caso contrario.
==	Igual. <code>True</code> si el operando de la izquierda es igual que el de la derecha; <code>False</code> en caso contrario.
!=	Distinto. <code>True</code> si los operandos son distintos; <code>False</code> en caso contrario.

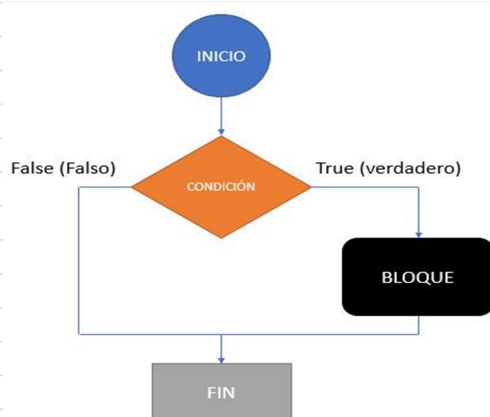
### Sentencias Condicionales IF,ELIF,ELSE.

Se trata la **estructura de control if ... elif ... else ...**

Estas construcciones permiten condicionar la ejecución de uno o varios bloques de **sentencias** para el cumplimiento de una o varias condiciones.

#### Condicional Simple:

Ejecuta un bloque de instrucciones cuando la **proposición** (condición) es **verdadera**; si es falsa, no hace nada. En inglés "if" significa "si" (condición).



#### Condicional Doble:

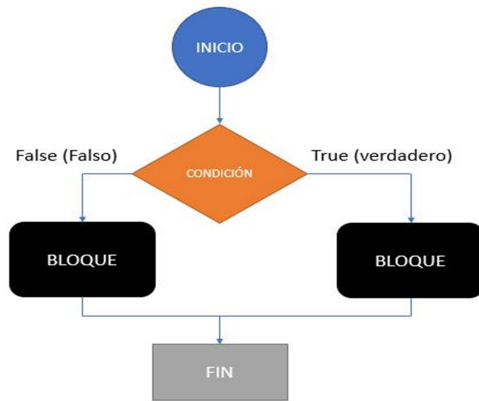
La estructura de control if ... else... permite que un programa ejecute unas instrucciones **cundo se cumple una condición** y otras instrucciones cuando no se cumple es condición. En inglés "if" significa "si" (Condición) y "else" significa "si no".

### IF (Condición):

Ejecuta la acción principal, \*\*cuando la condición se cumple\*\*

### ELSE :

Ejecuta la acción secundaria, \*\*cuando la condición no se cumple\*\*



### Condicional Múltiple.

La estructura de control IF...elif...elif... permite la ejecución de **múltiples condiciones de forma jerárquica**, es decir, si no se cumple la primera condición se evalúa la siguiente condición y así sucesivamente.

Este tipo de estructura puede o no acabar en una sentencia ELSE, dependerá de la tarea a controlar.

### IF (condición1):

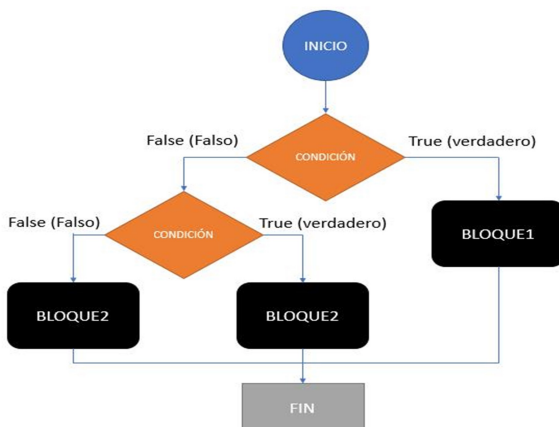
Ejecuta acción cuando se cumpla la condición1.

### ELIF(condición2):

Ejecuta acción cuando se cumpla la condición2.

### ELSE:

Ejecuta acción cuando ninguna condición se cumple.



### Condicional Anidada.

Una sentencia condicional **puede contener a su vez otra sentencia anidada.**

Es decir dentro de cada bloque de ejecución se implementa otra estructura de control que contendrá sus propios bloques de ejecución.

IF (condición1):

IF(condición anidada):

Se ejecuta cuando se cumple la condición anidada

ELSE:

si la condición anidada no se cumple

ELSE:

Ejecuta acción cuando ninguna condición se cumple