



Institución Universitaria
Acreditada en Alta Calidad

Digital II

Docente:

Yomin Estiven Jaramillo Múnera

yominjaramillo2283@correo.itm.edu.co

Ingeniero biomédico

Msc. en automatización y control industrial

Somos Innovación Tecnológica con

Sentido Humano



Alcaldía de Medellín

Variables

- Un lenguaje de programación, necesita poder acceder a la información, y tener la capacidad de modificar la información. Esto se logra mediante el uso de variables.
- Las **variables** son espacios reservados en la **memoria** del computador que pueden cambiar de contenido a lo largo de la ejecución de un programa.
- Las variables guardan **valores** (los datos de la aplicación).

Cada valor tiene un tipo, por ejemplo:

- 1 es un número entero.
- 2.5 es un número decimal.
- 'Hola' es un texto.

Tipos de variables

Tipos de datos en Python



Variables

Para utilizar una variable en un programa, es necesario declararla.

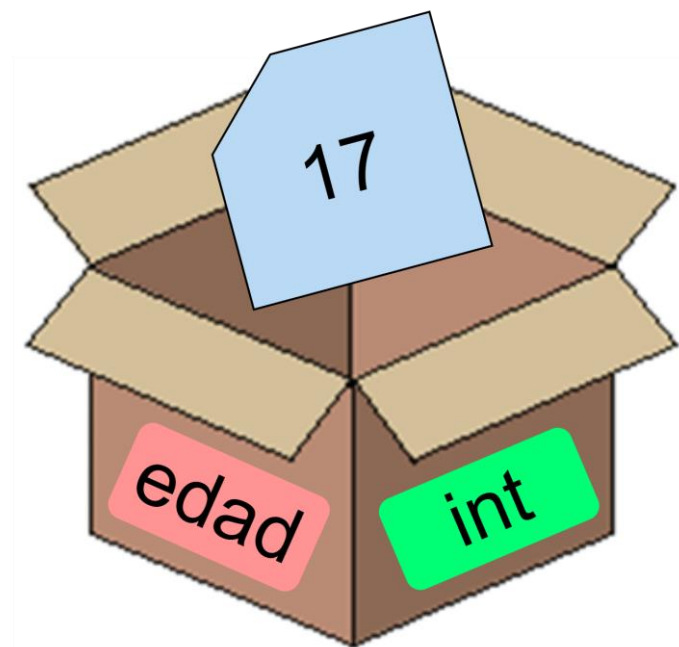
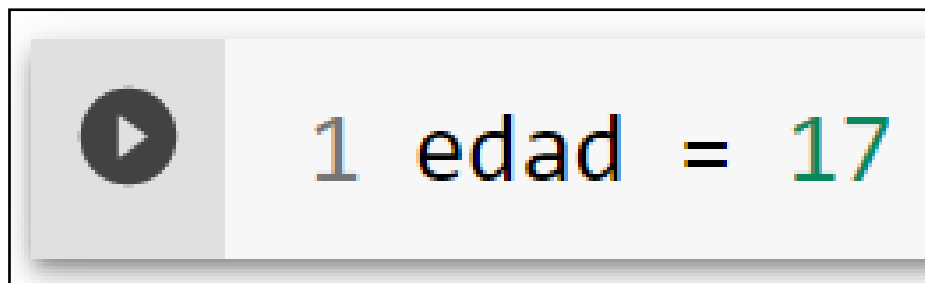
- Debemos **darle un nombre** utilizando letras y números sin espacios.
- **Se coloca el operador “=”**.
- **Al lado derecho se coloca el valor** que se quiere asignar a la variable.

nombreVariable = valorVariable

Variabales


$$1 \text{ edad} = 17$$

Ejemplo



Reglas para declarar una variable

- Existen 3 reglas para declarar variables en Python:
 - **Sólo puede contener letras, números y guiones bajos.**
 - **Debe iniciar con una letra o con un guion bajo.**
 - **No se puede usar una palabra reserva de Python.**

Palabras reservadas

abstract	continue	finally	int	public	throw
assert	default	float	interface	return	throws
boolean	do	for	long	short	transient
break	double	goto	native	static	true
byte	else	if	new	strictfp	try
case	enum	implements	null	super	void
catch	extends	import	package	switch	volatile
class	false	inner	private	synchronized	
const	final	instanceof	protected	this	while

Ejemplos

Para mostrar por pantalla el valor de una variable basta con invocar la función ***print()*** de Python, y **dentro de los paréntesis colocar el nombre de la variable.**

Declarar una variable e imprimir su valor

```
1 edad = 17
2 print(edad)
```

```
1 edad = 17
2 print(edadd)
```

```
17
```

Ejemplos



1 nombre = "Maria"

2 edad = 17

3 estatura = 1.60

Ejemplos

```
▶ 1 nombre = "Maria"  
2 edad = 17  
3 estatura = 1.60  
4  
5 print(estatura)  
6 print(nombre)  
7 print(edad)
```

```
↳ 1.6  
   Maria  
   17
```


Ejemplos

Declarar y modificar el valor y tipo de variables

```
▶ 1 precioProducto = 150
  2 print(precioProducto)
  3
  4 precioProducto = 250.5
  5 print(precioProducto)
```

⇨ 150
250.5

⇨ <class 'int'>
<class 'float'>

Entrada y salida

```
▶ 1 print("Ingrese su nombre:")  
2 nombre = input()  
3 print("Su nombre es:")  
4 print(nombre)
```

```
▶ 1 nombre = "Maria"  
2 edad = 17  
3  
4 print("Nombre: ")  
5 print(nombre)  
6 print("Edad: ")  
7 print(edad)
```

Condicionales

- Los programas se ejecutan siguiendo un orden determinado.
 - La ejecución inicia en la primera línea de código y continúa hacia abajo.
- Esto quiere decir que no tenemos una respuesta diferente o un orden diferente cuando se ingresan diferentes entradas o valores.



Variables booleanas

- Hasta este punto hemos trabajado con variables y valores *int*, *float* y *str*. Estos 3 tipos de variables permiten almacenar una cantidad bastante amplia de posibles valores.
- Las variables booleanas (*bool*) **sólo permiten almacenar 2 valores**. Verdadero (***True***) o Falso (***False***). Normalmente *True* se asocia a “sí” y *False* a “no”.
- Estas variables y valores **son generalmente utilizados para evaluar condiciones o comparaciones** (*tal como veremos más adelante*).

Operadores de comparación

- Los **operadores de comparación** permiten **comparar 2 valores** y **evaluar el resultado de esa comparación en un sólo valor booleano**.
- Python proporciona seis operadores de comparación. Veamos la siguiente tabla:

Operador	Significado
==	Igual a
!=	Diferente de
<	Menor que
>	Mayor que
<=	Menor o igual a
>=	Mayor o igual a

Ejemplo

Suponga que se declara una variable llamada **lado** con valor de 5.

lado = 5

Comparación	Resultado
lado < 0	<input type="text"/>
lado >= 2	<input type="text"/>
lado != 5	<input type="text"/>
lado == 5	<input type="text"/>
lado > -3	<input type="text"/>
lado <= 120	<input type="text"/>

Operadores lógicos

Los valores booleanos (True y False), también se pueden comparar entre ellos.

Similar a como funcionan los operadores de comparación, **los operadores lógicos permiten comparar valores booleanos y evaluar el resultado de las comparaciones en un solo valor booleano.**

Python proporciona tres operadores lógicos.

and (y)

or (o)

not (negación)

Operadores lógicos

Tabla de verdad del operador *and*

Expresión	Evalúa a
True <i>and</i> True	True
True <i>and</i> False	False
False <i>and</i> True	False
False <i>and</i> False	False

Tabla de verdad del *or*

Expresión	Evalúa a
True <i>or</i> True	True
True <i>or</i> False	True
False <i>or</i> True	True
False <i>or</i> False	False

Tabla de verdad del *not*

Expresión	Evalúa a
<i>not</i> True	False
<i>not</i> False	True

Ejemplos

Expresión	Resultado
(edad == 18) and (nombre == "Paola")	<input type="text"/>
(edad > 20) and (nombre == "Paola")	<input type="text"/>
(edad <= 60) and (nombre != "Rosa")	<input type="text"/>
(edad != 18) and (nombre == "Paola")	<input type="text"/>

Expresión	Resultado
(edad == 18) or (nombre == "Paola")	<input type="text"/>
(edad > 20) or (nombre == "Rosa")	<input type="text"/>
(edad <= 60) or (nombre != "Rosa")	<input type="text"/>
(edad != 18) or (nombre == "Paola")	<input type="text"/>

Expresión	Resultado
not(mayorDeEdad)	<input type="text"/>
not(edad > 20)	<input type="text"/>
not(edad <= 60)	<input type="text"/>
not(edad == 18)	<input type="text"/>

Condicionales

- Los condicionales o sentencias condicionales son grupos de instrucciones que se pueden ejecutar o no en función del valor de una condición.
- Existen varios tipos de sentencias condicionales, entre las cuales se encuentran:
 - *If-then* (Si – Entonces)
 - *If-then-else* (Si – Entonces – Si no)
 - *If-then-elif* (Si – Entonces – Si no, Si)

Estructura *if-then*

Una estructura if-then se compone de las siguientes partes:

1. Se utiliza la palabra reservada de Python ***if***.
2. Se define una **condición o expresión** que debe evaluar a un valor booleano. Esa condición o expresión se puede o no colocar entre paréntesis (son opcionales).
3. Se agregan **dos puntos (:)**.
4. Se coloca la **instrucción (o instrucciones) que se ejecutarían si la condición se evalúa a *True***. Esa instrucción (o instrucciones) se deben colocar como **bloques de código indentados**. Es decir, mover de izquierda a derecha mediante una tabulación.

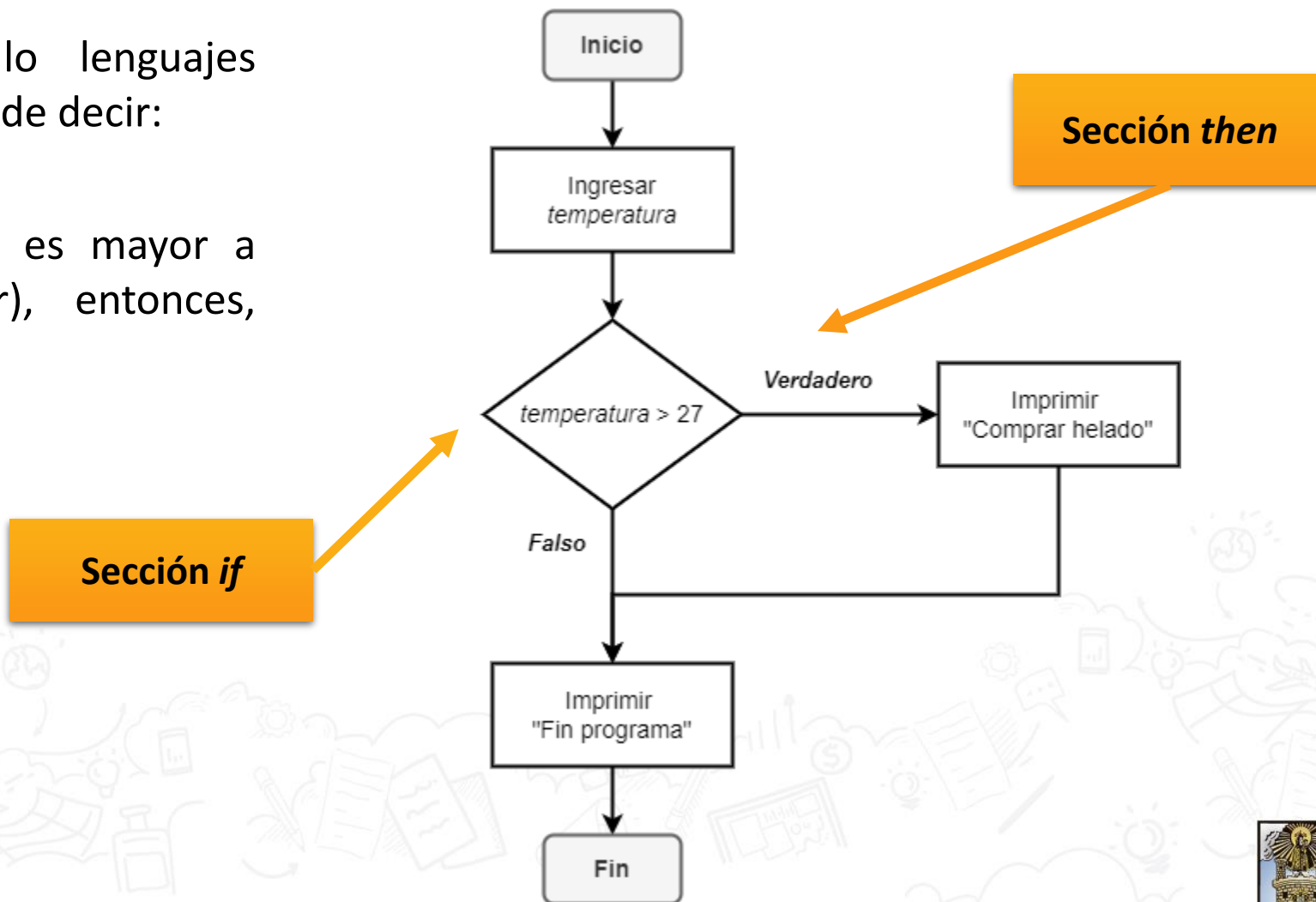
Estructura if-then

```
if (condición):  
    print("Cuerpo del if")  
    print("Cuerpo del if")  
  
print("Ya no hace parte del cuerpo del if")
```

Ejemplo

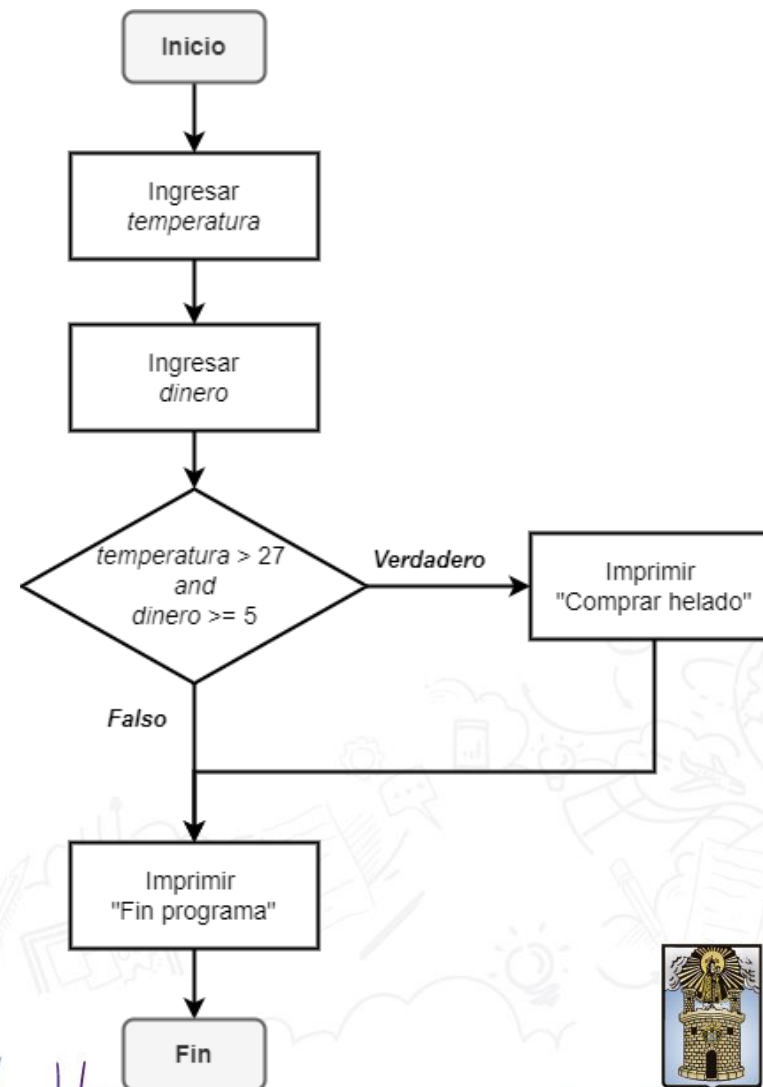
Ejemplo: similar a lo lenguajes naturales donde se puede decir:

- Si la temperatura es mayor a 27°C (hace calor), entonces, compro un helado.



Ejercicio

- Ahora suponga que debemos pedirle al usuario por pantalla tanto la *temperatura* como la cantidad de *dinero* que tiene.
- Y que imprima por pantalla “Comprar helado” si y solo si la *temperatura* es mayor a 27 (hace calor) y el *dinero* mayor o igual a 5 (tiene dinero para comprar el helado).
- Antes de finalizar el programa se deberá imprimir un mensaje “Fin programa”.





Institución Universitaria
Acreditada en Alta Calidad

¡Gracias!

Somos Innovación Tecnológica con *Sentido Humano*



Alcaldía de Medellín