

TC1028

Pensamiento Computacional para Ingeniería

Componentes de un programa y tipos de datos

Tecnológico de Monterrey





Componentes de un programa

1. Identificador
2. Variable
3. Constante
4. Instrucciones



A decorative graphic on the left side of the slide. It features a large cyan hexagon in the center containing the number '1'. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and cyan. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and radiating lines. The entire graphic is set against a dark blue background.

1

Identificador



Identificador

Son nombres dados a variables, constantes y funciones de un programa. Se forman con la combinación de letras, números y otros símbolos. El primer carácter debe ser una letra.

Ejemplos: PI, vocales, x, i, etc.



A decorative graphic on the left side of the slide. It features a large cyan hexagon in the center containing the number '2'. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and cyan. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and radiating lines. The entire graphic is set against a dark blue background.

2

Variable



Variable

Es un identificador que puede cambiar durante la ejecución de un programa.

Ejemplos:

- $x = 1$
- $y = 2$
- $i = 1$
- $x = x + y$
- $i = i + 1$



A decorative graphic on the left side of the slide. It features a large cyan hexagon in the center, surrounded by several smaller hexagons in various shades of blue and cyan. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and radiating lines, and a speech bubble icon.

3

Constante



Constante

Es un identificador al que se le asigna un valor fijo; es decir, no cambia durante la ejecución del programa. Puede ser un número, un carácter o una lista de caracteres.

Ejemplos:

`PUERTO = 3307`

`USUARIO = "root"`

`PASSWORD = "123456"`



A decorative graphic on the left side of the slide. It features a large cyan hexagon in the center containing the number '4'. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and cyan. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and radiating lines. The entire graphic is set against a dark blue background.

4

Instrucciones




Instrucciones

Unidad ejecutable más pequeña de un programa.
Las instrucciones controlan el flujo u orden de ejecución.

Ejemplos: and, break, continue, elif, else, for, if, import, is, not, or, print, return, while, with, etc.





Reglas y convenciones de nombres para variables y constantes (Python)

- ❖ El **primer carácter no** puede ser un **número** o dígito.
- ❖ Inicie con letra o guion bajo _ (El resto puede ser letra, número o guion bajo _)
NOTA: Los nombres que comienzan con guion bajo (_simple, __o doble) se reservan para variables con significado especial.
- ❖ No use **símbolos especiales** como !, @, #, \$, %, etc.
- ❖ Los nombres de las variables pueden tener la combinación de letras en minúsculas (a - z) o MAYÚSCULAS (A - Z) o dígitos (0 - 9) o un guion bajo (_). Por ejemplo:
 - ☐ snake_case
 - ☐ MACRO_CASE
 - ☐ camelCase
- ❖ No pueden usarse como identificadores, las **palabras reservadas**.





Palabras reservadas Python

Existen ciertas palabras que tienen significado especial para el intérprete de Python. Estas no pueden utilizarse para nombrar variables o constantes.

Algunas palabras reservadas son las siguientes:

- ❖ and
- ❖ break
- ❖ continue
- ❖ def
- ❖ elif
- ❖ else
- ❖ for
- ❖ from

- ❖ if
- ❖ import
- ❖ in
- ❖ is
- ❖ not
- ❖ or
- ❖ print
- ❖ return

- ❖ while
- ❖ with



Actividad grupal

Señala si el nombre de las siguientes **variables** es **correcto/incorrecto**:

1

- | | |
|--------------------------------|------------------------------------|
| 1. <code>_num_Alumno</code> | 8. <code>nombreProceso</code> |
| 2. <code>#exterior</code> | 9. <code>correo_electronico</code> |
| 3. <code>5Telefono</code> | 10. <code>practica2</code> |
| 4. <code>Radio</code> | 11. <code>nombre del perro</code> |
| 5. <code>direccion_casa</code> | 12. <code>teléfono</code> |
| 6. <code>miPerro</code> | 13. <code>numero-lista</code> |
| 7. <code>nombre_Pila</code> | 14. <code>while</code> |



Tipos de datos

Son los valores que puede tomar una variable.

Los tipos de datos son los siguientes:

- Numéricos
- Alfanuméricos
- Lógicos



Tipos de datos

- ❖ **Datos numéricos:** Pueden representarse de dos formas:
 - ❑ **Números enteros**, los cuales no tienen componentes fraccionarios y pueden ser positivos o negativos
 - ❑ **Números reales**, pueden tener cifras decimales y pueden ser positivos o negativos.
- ❖ **Datos alfanuméricos:** Son datos que contienen caracteres no numéricos ya sean letras, caracteres especiales (.,='+) o los dígitos mismos.
- ❖ **Datos Lógicos:** Podemos hablar de otro tipo de datos llamado “booleano”, el cual sólo puede tomar uno de dos valores : verdadero o falso.



Tipos de datos

Numéricos

Tipo de dato	Python	C
Entero	<code>num = 40</code>	<code>int num = 40;</code>
Real	<code>numReal = 45.2</code>	<code>float numReal = 45.2;</code>

Como puede ver en **Python**, a diferencia de muchos otros lenguajes, no se declara el **tipo de la variable** al crearla.



Tipos de datos

Alfanuméricos

Tipo de dato	Python	C
Caracter	letra = 'c'	char letra = 'a';
Cadena de caracteres	c = "Hola mundo" c = "12345"	char c[10] = "Hola mundo";

Como puede ver en **Python**, a diferencia de muchos otros lenguajes, no se declara el **tipo de la variable** al crearla.

Tipos de datos

Lógicos o booleanos

Tipo de dato	Python	C
Booleano	<code>x = True</code> <code>x = 5 > 3</code>	<code>bool x = True;</code>
Booleano	<code>y = False</code> <code>y = 3 > 5</code>	<code>bool y = False;</code>

Como puede ver en **Python**, a diferencia de muchos otros lenguajes, no se declara el **tipo de la variable** al crearla.

Tipos de datos

Python ve diferente la variables si tiene mayúsculas o minúsculas:

Ejemplo:

En este ejemplo, las dos variables son distintas

Nombre = "juan"
nombre = "pedro"

Thonny

- Instala **Thonny**: <https://thonny.org/>
- **Thonny** es un entorno de desarrollo integrado (IDE) para el lenguaje **Python** diseñado para principiantes.





Tipos de datos

Python

- Entero (**Integer**)
- Real (**Float**)
- Lógico o booleano (**Boolean**)
- Cadena de caracteres (**String**)



Actividad grupal



2

- Inicializa 5 variables con diferentes tipos de datos (entero, real, booleano, string)
- Con la instrucción **type** averigua el tipo de dato detectado.

Ejemplo:

```
1 A = 2
2 print(type(A))
```

Shell ×

```
>>> %Run tipoA.py
<class 'int'>
```

Actividad grupal



3

Escribe un algoritmo para verificar si un precio dado por el usuario es válido o no lo es, para ser válido debe ser un valor positivo o cero.

Casos de prueba:

precio	Resultado
10	El precio es válido
0	El precio es válido
-3	El precio no es válido

Actividad grupal

Algoritmo

3

1. Pedir el **precio**
2. Si el precio es mayor o igual a cero
Escribir **“El precio es válido”**
SiNo
Escribir **“El precio no es válido”**



4

Actividad grupal

Crear un algoritmo que dados tres números encuentre el más pequeño.

Casos de prueba:

radio	área
10, 23, 1	1
200, -23, 9	-23
0, 0, 0	0

Actividad grupal

Algoritmo

4

1. Pedir a, b, c
2. Si $(a < b \text{ y } a < c)$
 menor = a
 SiNo
 Si $(b < a \text{ y } b < c)$
 menor = b
 SiNo
 menor = c
3. Escribir menor

Actividad grupal

Definir el **algoritmo** y utiliza **Thonny** para diseñar el **programa** en **Python**:

- ❖ Pedir al usuario el radio.
- ❖ Calcular el área de un círculo a partir de su radio con la fórmula:
 $\text{area} = \text{PI} * \text{radio}^2$
- ❖ Guarda el nombre de tu archivo en Python: **areaCirculo.py**

Casos de prueba:

radio	área
10	314.1592
5	78.5398
-5	78.5398

Actividad grupal

Algoritmo

5

1. Definir el valor de PI como 3.141592
2. Pedir el radio
3. $\text{area} = \text{PI} * \text{radio} * \text{radio}$
4. Escribir (area)

Actividad grupal

Programa

5

```
areaCirculo.py x
1  # Programa que calcula el área de un círculo
2  """ 1. Definir el valor de PI como 3.141592
3        2. Pedir el radio
4        3. area = PI * radio*radio
5        4. Escribir (area) """
6
7  PI = 3.141592
8  radio = float(input("Introduce el valor del radio: "))
9  area = PI * radio * radio
10 print ("El area del circulo es:", area)

Shell x

>>> %Run areaCirculo.py

Introduce el valor del radio: 10
El area del circulo es: 314.1592
```

Th

Actividad grupal

Definir el **algoritmo** y utiliza **Thonny** para diseñar el **programa** en **Python**:

- ❖ Un estudiante desea conocer el área de un triángulo a partir de la base y la altura.
- ❖ Usar la fórmula: **$\text{area} = b * h / 2$**
- ❖ Guarda el nombre de tu archivo en Python: **areaTriangulo.py**

Casos de prueba:

base	altura	área
10	2	10
25	15	187.5
5.5	2.8	7.69999

Actividad grupal

Definir el **algoritmo** y utiliza **Thonny** para diseñar el **programa** en **Python**:

- ❖ Convierta el precio de un producto de pesos a dólares, si se tiene el tipo de cambio del dólar y el precio en pesos del producto. El resultado debe mostrar “El precio del producto en dólares es:” X .
- ❖ Guarda el nombre de tu archivo en Python: **pesosDolares.py**

Casos de prueba:

tipoCambio	precioPesos	precioDolares
19.5	100	5.128205
20	350.5	17.525
21.2	230	10.849056

Actividad grupal

8

Definir el **algoritmo** y utiliza **Thonny** para diseñar el **programa** en **Python**:

- ❖ Un alumno desea conocer la calificación final de su materia de Programación.
- ❖ La rúbrica de esta materia se compone de la siguiente manera:

• Parcial 1	20%
• Parcial 2	35%
• Proyecto final	15%
• Examen final	30%
- ❖ Guarda el nombre de tu archivo en Python: **calificacion.py**



Gracias

