Fundamentos de Programación con Python

Unidad II
Elementos básicos

Docente: T.S.U Gerardo Alí Ferraro Schelijasch
gerferr83@soltecferr.com
https://soltecferr.com

Unidad II - Elementos básicos

Resumen

- Definición de Variable.
- Definición de Constante.
- Técnicas de Naming.
- Tipos de datos.
    - Numéricos.
        - Enteros.
        - Flotantes o reales
        - Complejos
    - Booleanos
    - Cadena de caracteres

Unidad II - Elementos básicos

Resumen

- Operadores
  - Operadores aritméticos.
  - Operadores relacionales.
  - Operadores lógicos.
  - Operadores de asignación
- Comentarios.
- UML.

Unidad II - Elementos básicos

Variables:

Representa un espacio en la memoria con un nombre asociado que puede contener un valor que puede cambiar durante la ejecución de un programa. Las variables permiten a los programadores escribir código que puede trabajar con datos dinámicos y cambiar su comportamiento en función de diferentes condiciones y entradas.

a = 2
b = 3
c = ‘Hola’
d = ‘mundo’

Unidad II - Elementos básicos

Características de las variables:

- **Nombre:** El identificador único que se utiliza para referirse a la variable. Debe seguir las reglas de nomenclatura del lenguaje de programación
- **Tipo de Dato:** Determina qué tipo de datos puede almacenar la variable, como enteros, flotantes, cadenas de texto, booleanos, etc.
- **Valor:** El dato actual almacenado en la variable.
- **Alcance:** El contexto dentro del cual la variable es accesible. Puede ser local (dentro de una función o bloque de código) o global (accesible en todo el programa).
- **Tiempo de Vida:** El periodo durante el cual la variable existe en la memoria. Una variable local existe solo durante la ejecución de la función o bloque donde fue definida, mientras que una variable global existe durante la ejecución del programa completo.

Unidad II - Elementos básicos

Constante:

una constante es un tipo de variable cuyo valor no puede cambiar durante la ejecución del programa. Una vez que se asigna un valor a una constante, este valor permanece fijo y no se puede modificar. Las constantes se utilizan para representar valores que son conocidos y que deben permanecer inalterables a lo largo de la ejecución del programa, proporcionando claridad y evitando errores accidentales de modificación.

En Python:

```
PI = 3.14159
GRAVITY = 9.8
MAX_CONNECTIONS = 100
```

aunque no hay una palabra clave específica para definir constantes, se sigue una convención de nomenclatura para indicar que ciertas variables deben tratarse como constantes. La convención consiste en usar letras mayúsculas para los nombres de las constantes.

Unidad II - Elementos básicos

Convenciones de nombres de variables:

- Los nombres de variables pueden contener letras, números y guiones bajos.
- Los nombres de variables deben comenzar con una letra o un guion bajo.
- Los nombres entre variables distinguen entre mayúsculas y minúsculas.
- Los nombres de las variables deben ser breves pero descriptivos.

Técnicas de naming:

Las técnicas de naming (nombrado) son prácticas y convenciones utilizadas para asignar nombres a variables, funciones, clases y otros elementos en el código fuente. Un buen esquema de nombrado mejora la legibilidad, mantenibilidad y comprensión del código. A continuación se describen algunas técnicas y convenciones comunes para el nombrado en programación.

Técnicas de Naming:

Los técnicas de naming mas usadas para los identificadores en programación son:

- Camel Case
- Pascal Case
- Snake Case
- Kebab Case

Camel Case:

Cada palabra en el nombre comienza con una letra mayúscula, excepto la primera palabra. Utilizado comúnmente en JavaScript y Java para variables y funciones.

Ejemplo:
TotalAmount = 10
BaseCuadrado = 12

Técnicas de Naming:

Pascal Case:

Similar a CamelCase, pero la primera palabra también comienza con una letra mayúscula. Utilizado para nombres de clases en muchos lenguajes.

Ejemplo:

```
class CalculadorCientifica
class ArtistaCine
```

Técnicas de Naming:

Snake Case:

Las palabras se separan con guiones bajos y están en minúsculas. Común en Python para variables y funciones.

```
total_amount = 12
def area_triangulo:
    pass
```

Kebab Case:

Similar a Snake_Case, pero utiliza guiones en lugar de guiones bajos. Es común en nombres de archivos y URLs.

Unidad II - Elementos básicos

Técnicas de Naming:

Snake Case:

Las palabras se separan con guiones bajos y están en minúsculas. Común en Python para variables y funciones.

```
total_amount = 12
def area_triangulo:
    pass
```

Kebab Case:

Similar a Snake_Case, pero utiliza guiones en lugar de guiones bajos. Es común en nombres de archivos y URLs.

Unidad II - Elementos básicos

Tipos de datos:

Los tipos de datos definen un conjunto de valores que tienen una serie de características y propiedades determinadas. En Python existen los siguientes tipos de datos:

Simples y compuestos:

Estos tipos de datos representan valores individuales y son los bloques de construcción básicos de cualquier lenguaje de programación.

Numéricos:
Los tipos de datos numéricos se utilizan para representar y manipular valores numéricos.

Unidad II - Elementos básicos

Tipos de datos:

Numéricos:

- **Enteros:**

Son aquellos tipos de datos que se utilizan para indicar los números enteros, positivos o negativos, sin parte decimal.

a = 4

- Punto Flotante (Floating Point)

Representan números que tienen una parte decimal.

a = 4.55
b = 6.50

Tipos de datos:

Números complejos:

Representan numeros que tienen una parte real y una parte imaginaria.

num_complejo = 1 + 2j

Booleanos:

En programación, los tipos de datos booleanos se utilizan para representar valores de verdad, es decir, pueden tomar solo uno de dos valores: verdadero o falso. Estos valores son fundamentales para la lógica de control en la programación, como en las declaraciones condicionales y los bucles.

Unidad II - Elementos básicos

Tipos de datos:

Booleanos:

- Representa un valor de verdad.
- En la mayoría de los lenguajes de programación, los valores booleanos se representan como true y false.

String o cadenas de caracteres:

Son variables que sirven para representar texto mediante una “cadena de caracteres”.

a = ‘hola mundo’

Unidad II - Elementos básicos

Operadores:

Los operadores son símbolos o palabras reservadas que se utilizan para realizar operaciones sobre uno o más operandos (valores o variables). Los operadores son fundamentales en la programación, ya que permiten manipular datos y realizar cálculos, comparaciones, y otras tareas.

Tipo de operadores:

- **Operadores aritméticos.**

Son aquellos que nos permiten realizar las diferentes operaciones aritméticas.

suma +,multiplicación *, división /, resta - , % modulo, cociente //, potencia**.

Unidad II - Elementos básicos

Operadores:

- **Operadores aritméticos. x = 10 , y = 3**

Operador	Nombre	Ejemplo
+	Suma	x + y = 13
-	Resta	x - y = 7
*	Multiplicación	x * y = 30
/	División	x/y = 3.333
%	Módulo	x%y = 1
**	Exponente	x ** y = 1000
//	Cociente	3

Unidad II - Elementos básicos

Operadores:

- **Operadores de asignación:**

Los operadores de asignación o assignment operators nos permiten realizar una operación y almacenar su resultado en la variable inicial.

=, +=, -= entre otros

Se usan normalmente cuando queremos asignar un valor a una variable.

Unidad II - Elementos básicos

Operadores:

- **Operadores de asignación:**

Operador	Ejemplo	Equivalente
=	x=7	x=7
+=	x+=2	x=x+2 = 7
-=	x-=2	x=x-2 = 5
*=	x*=2	x=x*2 = 14
/=	x/=2	x=x/2 = 3.5
%=	x%=2	x=x%2 = 1
//=	x//=2	x=x//2 = 3
**=	x**=2	x=x**2 = 49
&=	x&=2	x=x&2 = 2
|=	x|=2	x=x|2 = 7
^=	x^=2	x=x^2 = 5
>>=	x>>=2	x=x>>2 = 1
&lt&lt=	x&lt&lt=2	x=x&lt&lt2 = 28

Unidad II - Elementos básicos

Operadores:

- **Operadores Relacionales:**

Los operadores relacionales, o también llamados comparison operators nos permiten saber la relación existente entre dos variables. Se usan para saber si por ejemplo un número es mayor o menor que otro. Dado que estos operadores indican si se cumple o no una operación, el valor que devuelven es True o False.

Ejemplo:

==, >=, <=, <,>

Operador	Nombre	Ejemplo
==	Igual	x == y = False
!=	Distinto	x != y = True
>	Mayor	x > y = False
&lt	Menor	x &lt y = True
>=	Mayor o igual	x >= y = False
&lt=	Menor o igual	x &lt y = True

Unidad II - Elementos básicos

Operadores:

- **Operadores lógicos:**

Los operadores lógicos o logical operators nos permiten trabajar con valores de tipo booleano. Un valor booleano o bool es un tipo que solo puede tomar valores True o False. Por lo tanto, estos operadores nos permiten realizar diferentes operaciones con estos tipos, y su resultado será otro booleano.

Operador	Nombre	Ejemplo
and	Devuelve True si ambos elementos son True	True and True = True
or	Devuelve True si al menos un elemento es True	True or False = True
not	Devuelve el contrario, True si es Falso y viceversa	not True = False

Unidad II - Elementos básicos

Comentarios en Python:

En Python, los comentarios son líneas de texto dentro del código que no se ejecutan como parte del programa. Sirven para documentar el código, explicando su propósito, funcionamiento, y cualquier otra información que sea útil para entenderlo. Los comentarios son esenciales para mantener el código legible y fácil de mantener, especialmente en proyectos grandes o cuando múltiples desarrolladores están involucrados.

- Comentarios de una sola linea:

Se indican con el simbolo # ejemplo:

#suma de dos numeros

Unidad II - Elementos básicos

Comentarios en Python:

- Comentarios de multiples lineas:

Normalmente se hace usando el simbolo “”” “”” o ‘’’ ‘’’

Ejemplo

“””

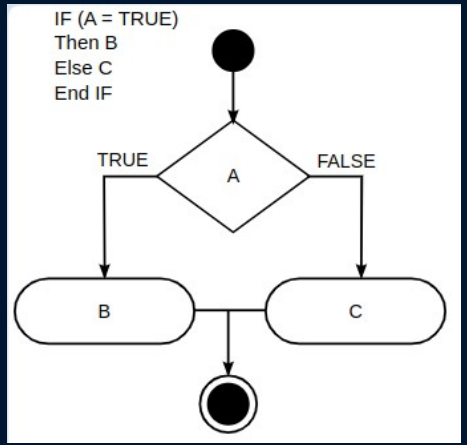
Esto es un comentario

“””

Unidad II - Elementos básicos

UML:

UML, que significa "Lenguaje de Modelado Unificado" (Unified Modeling Language), es un lenguaje de modelado estándar ampliamente utilizado en ingeniería de software para especificar, visualizar, construir y documentar los artefactos de los sistemas de software. UML proporciona una forma estandarizada de visualizar el diseño de un sistema.



Términos de la licencia.