

Variables

Tipos de datos



IITA 2023

Introducción

CONCEPTOS INICIALES

- **Software**: Conjunto de programas que realizan alguna tarea dentro del sistema. Constituyen la parte **lógica** del mismo, y al ser INTANGIBLE no se deterioran.
- **Hardware**: Elementos tecnológicos TANGIBLES del sistema, están sujetos a fallas y por lo tanto se deterioran con el uso. Es posible repararlos



Introducción

CONCEPTOS INICIALES

- ¿Qué es un “Lenguaje”?

Conjunto de símbolos, reglas y notaciones utilizado para comunicarse

- ¿Y cuando hablamos de “Lenguajes de Programación”?

Conjunto de símbolos, reglas y notaciones que se utilizan para comunicarse con la **COMPUTADORA**. Expresan la estructura y la lógica de un programa

Tienen **SINTAXIS** y **SEMÁNTICA**

- ¿Un programa?

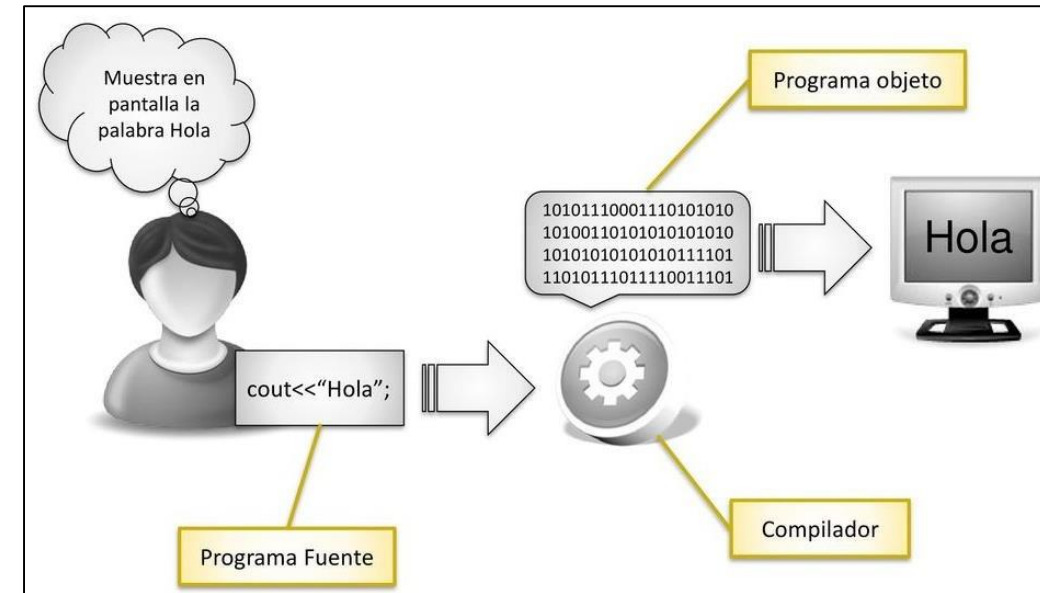
Conjunto ORDENADO de instrucciones escritas en un lenguaje determinado, que le indican a la computadora cómo realizar una determinada tarea

↓
¿POR QUÉ?



Programa

- El programador teclea instrucciones en un editor, que es un programa parecido a un simple procesador de palabras. Estas instrucciones son almacenadas en archivos denominados **programas fuente** (código fuente). Si los programadores necesitan hacer cambios al programa posteriormente, vuelven a correr el editor y cargan el programa fuente para modificarlo.
- El proceso de conversión de programas fuente a programas objeto se realiza mediante un programa denominado compilador. El compilador toma un programa fuente y lo traduce a **programa objeto** y almacena este último en otro archivo.



Programa

→ **Programa fuente.** Es el programa escrito en alguno de los lenguajes y que no ha sido traducido al lenguaje de la máquina, es decir, el programa que no está en código de máquina y que por lo tanto no puede ser ejecutable

→ **Programa objeto.** Es aquel programa que se encuentra en lenguaje máquina y que ya es ejecutable por ésta.

→ **Compilador.** Es un programa que traduce un lenguaje de alto nivel al lenguaje máquina. Un programa compilado indica que ha sido traducido y está listo para ser ejecutado

→ **Intérprete.** Traductor de lenguajes de programación de alto nivel. Los intérpretes ejecutan un programa línea por línea. El programa siempre permanece en su forma original (programa fuente) y el intérprete proporciona la traducción al momento de ejecutar cada una de las instrucciones



Python

(Guido van Rossum, principio de los '90)

- Lenguaje de alto nivel
 - Existen lenguajes de bajo (ceranos a la arq. del HW) y alto nivel (ceranos a prgs y us)
- Lenguaje interpretado
 - PVM (Python Virtual Machine)
- Multiparadigma
 - Admite diferentes 'tipos' de programación
 - Algunos paradigmas de programación son: (en verde los que admite Python)
 - Programación asíncrona
 - **Imperativo**
 - Lógico
 - **Funcional**
 - Declarativo
 - Estructurado
 - Dirigido por eventos
 - Modular
 - Orientado a aspectos
 - **Orientado a objetos**
 - Con restricciones

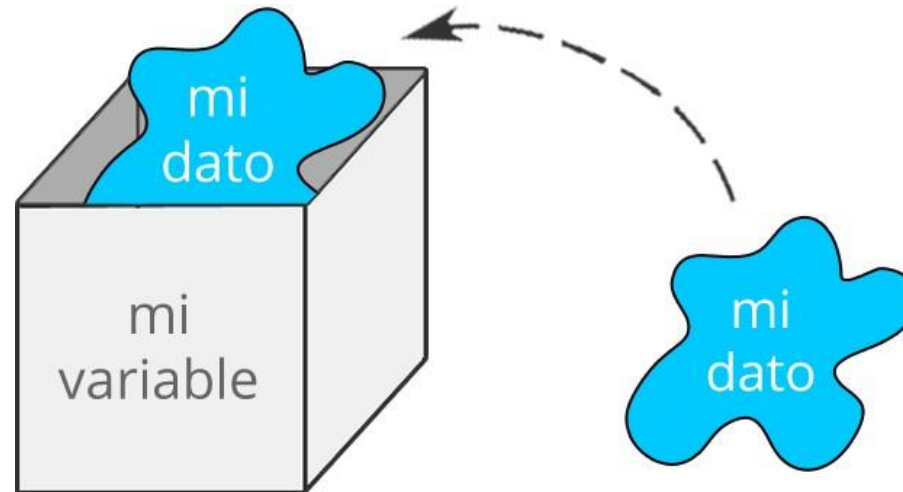


IITA 2022



Variables

- **TIPOS DE DATOS** → Es la clase del dato. Puede ser numérico o no numérico. Su clasificación y las operaciones sobre ellos dependen del lenguaje de programación que se este utilizando
- **VARIABLE** → Porción de memoria donde se almacena un valor numérico de un tipo de dato, y se identifica por un nombre dado.



Ejemplo:

Edad = 25
Nombre = 'Nico'



IITA 2023



Existe la sentencia “`type()`”
para indicar el tipo de dato de
una variable

Python – Tipos de Datos

- Tipos de datos principales:

- Entero Corto: `int` (-2.147.483.648 a 2.147.483.647)
- Entero Largo: `long` (-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807)
- Real: `float` ($\pm 2,2250738585072020 \times 10^{-308}$ a $\pm 1,7976931348623157 \times 10^{308}$)
 - Los números reales se expresan con “puntos” y no con “comas”
- String: `str` (secuencia de char o caracteres)
- Logicos: `bool`: True - False
- Listas y Tuplas:
 - Para una lista se usa `[]`
 - Para un tupla se usa `()`



Conversores

- Int: Convierte la expresión encerrada entre paréntesis en un entero.
 - Str: Convierte la expresión encerrada entre paréntesis en un string.
 - Bool: Convierte la expresión encerrada entre paréntesis en un booleano.
 - Float: Convierte la expresión encerrada entre paréntesis en un decimal.
-
- Por ejemplo:
 - Edad = int('25')



Operadores

Operador	Descripción	Ejemplo
+	Suma	<pre>>>> 3 + 2 5</pre>
-	Resta	<pre>>>> 4 - 7 -3</pre>
-	Negación	<pre>>>> -7 -7</pre>
*	Multiplicación	<pre>>>> 2 * 6 12</pre>
**	Exponente	<pre>>>> 2 ** 6 64</pre>
/	División	<pre>>>> 3.5 / 2 1.75</pre>
//	División entera	<pre>>>> 3.5 // 2 1.0</pre>
%	Módulo	<pre>>>> 7 % 2 1</pre>

Operador	Descripción	Ejemplo
==	¿son iguales a y b?	<pre>>>> 5 == 3 False</pre>
!=	¿son distintos a y b?	<pre>>>> 5 != 3 True</pre>
<	¿es a menor que b?	<pre>>>> 5 < 3 False</pre>
>	¿es a mayor que b?	<pre>>>> 5 > 3 True</pre>
<=	¿es a menor o igual que b?	<pre>>>> 5 <= 5 True</pre>
>=	¿es a mayor o igual que b?	<pre>>>> 5 >= 3 True</pre>

Operador lógico “o” (Ú): Se usa la palabra reservada “**or**”.

Operador lógico “y” (^): Se usa la palabra reservada “**and**”



IITA 2023



Python - Reglas

- Python SI distingue entre minúsculas y mayúsculas
- Después de cada **ciclo** o **decisión** debe ir “:” y las instrucciones tienen que ir indentadas
- Los **comentarios** empiezan con “#” o “ ” ”
- No poner caracteres blancos en las siguientes situaciones:
 - Inmediatamente dentro de paréntesis, corchetes o llaves
 - Inmediatamente antes de una coma, un punto y coma o dos puntos
 - Más de un espacio alrededor de un operador de asignación
- Palabras reservadas: **and, as, assert, break, class, continue, def, del, elif, else, except, finally, for, from, global, if, import, in, is, lambda, onlocal, not, or, pass, raise, return, try, while, with, yield**



Python – Sentencias

- Algunas sentencias interesantes son:
 - ASIGNACIONES
 - Se usa el “=”
 - Ejemplos:
 - Nombre=‘IITA’
 - c=c+ 1
 - INGRESAR DATOS
 - Se usa “input()”
 - Ejemplo: nombre=input()
 - MOSTRAR DATOS
 - Se usa “print()”
 - Dentro de los paréntesis podemos mostrar un mensaje o el valor de una variable
 - También podemos usar “end=“””, para que la salida o respuesta sea a continuación



Vamos a practicar un poco...

- 1- Mensaje simple: Almacene un mensaje en una variable e imprímalo en pantalla. Después cambie el valor del mensaje e imprímalo nuevamente
- 2 - Almacene el nombre de una persona en una variable, imprima un mensaje para esa persona. Por ejemplo “Hola Fede, ¿te gustaría aprender a programar?”. En este caso ‘Fede’ es el valor que tiene la variable utilizada



Vamos a practicar un poco...

- ¿Estos datos (5-3-8) se están guardando en la misma variable o en variables distintas?
 - Variable=5
 - variable=3
 - VaRiAbLe=8
 - ¿Qué valor tiene a?
 - a=8
 - a=20
 - print(a)
 - Cuidado con esto!
 - 1variable=23.95
- Los nombres de vbles SÓLO pueden comenzar con letras o guiones bajos (_)



Ahora un poquito más difícil...

- 3 - Escribí un programa que solicite al usuario un número y le reste el 15%, (almacenando todo en una única variable). A continuación, mostrar el resultado final en pantalla.
- 4 - Escribí un programa que solicite al usuario ingresar tres números para luego mostrarle el promedio de los tres.
- 5 - Escriba un programa que le pida al usuario que ingrese nombre y edad. Luego muestre un mensaje donde le informe el año en que va a cumplir 100
- 6 - Escriba un programa que solicite un peso (en kg) y muestre si equivalente en libras.
 - (1 kg = 2,2 lb)
- 7 - Escriba un programa que convierta la temperatura de grados Celsius a Fahrenheit

(1 Celsius = 33,8 F s/Google)

IITA 2023

