



Agencia de
Aprendizaje
a lo largo
de la vida

Codo a Codo inicial

Clase 5

Les damos la bienvenida

Vamos a comenzar a grabar la clase

Clase 04

Análisis y resolución de problemas

- Programa Informático. Concepto.
- Estructura de un programa.
- Metodologías para la resolución de problemas.
- Instalación de PSeint.
- Desafíos.

Clase 05

Sintaxis y variables

- **Variables. Características, declaración, asignación, semántica, alcance.**
- **Relación entre variables y datos.**
- **Constantes.**
- **Tipado.**
- **Desafíos.**

Clase 06

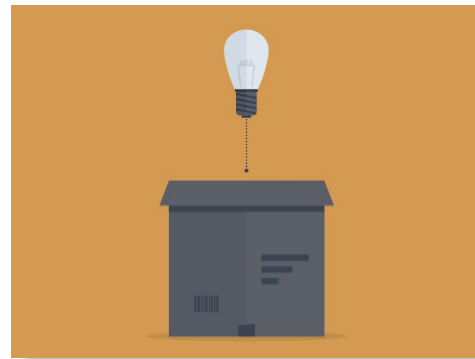
Operadores de asignación - Operadores aritméticos - input y output

- Definición de variables.
- Asignación de valores.
- Asignación de resultados.
- E/S de valores por pantalla.

Variables

Variables definición y concepto

- **Una variable** en programación es utilizada de forma habitual.
- Se trata de un elemento fundamental para gestionar la información que se decida incorporar en el código y básicamente **funciona para guardar datos**.
- **Una variable almacena un valor único.**

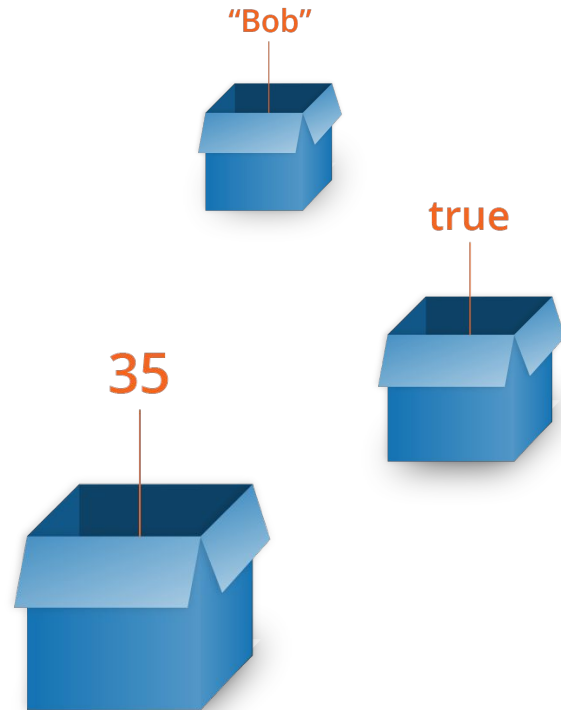


Relación entre variable y la memoria física.

Asignación de memoria.

Una variable es el **lugar en la memoria ram reservado para almacenar un dato.**

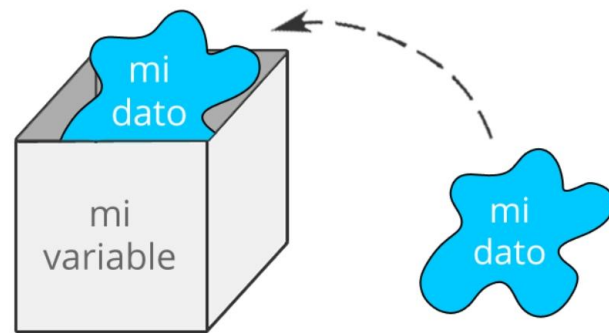
Cuando se ejecute un programa o una aplicación, la variable tendrá acceso al correspondiente bloque de memoria.



Relación entre variable y la memoria física.

Declaración de una variable.

En el algoritmo, expresado como pseudocódigo o código fuente, una variable, recibe **un nombre** y el **tipo de dato que se desea que contenga** (A este proceso se lo llama **DECLARAR LA VARIABLE**).



Declaración e inicialización de una variable

- **Una variable se declara** para indicarle al programa **a partir de qué lugar empieza a existir**, qué **nombre** tendrá y qué tipo de **datos almacenará**.
- Para declarar una variable usaremos una instrucción compuesta **del nombre del tipo de dato de la variable**, y el **nombre de la variable**.
- La variable se asigna o inicializa **con un operador de asignación (signo =) y un valor inicial**.

Ejemplo `miNumero = 5;`

Ejemplo de declaración de una variable en PseInt

```
1 Algoritmo inicializacion_de_variables
2
3   //Primero las declaro con nombre y tipo
4   Definir miPrimerVariable Como Real;
5   Definir otraVariable Como Real;
```

INSTRUCCION DE
APERTURA CON
LA PALABRA
RESERVADA
Definir

**NOMBRE DE LA
VARIABLE**
CUALQUIER
NOMBRE
CUIDANDO REGLA
Y CONVENCION

**DEFINICION DEL
TIPO DE
VARIABLE**

Caracter
Entero
Logico
Real

Nombre de las variables

Las reglas para el **nombre de una variable** viene definida según cada lenguaje de programación, la mayoría cumple estas reglas:

- No utilizar la letra ñ.
- Comenzar con una **letra minúscula**.
- Puede **contener** números y guiones. **NO iniciar** con número.
- No utilizar palabras reservadas.
- No se separan con espacio las palabras compuestas.

Nombre de las variables - Palabras compuestas

Las variables que necesariamente recurran a nombres compuestos por más de una palabra, deberán seguir algunas de las siguientes convenciones:

- **Regla del camel case**, cada nueva palabra inicia en mayúscula sin recurrir a separaciones entre ambas.

Ejemplo: miVariable

- **Regla snake case**, cada nueva palabra se la separa con guion bajo.

Ejemplo: mi_variable

- **Regla kebab case**, cada nueva palabra se la separa con guion medio.

Ejemplo: esta-es-una-variable-en-kebab

Inicialización de una variable

Es la **asignación de un valor inicial** a la variable.

Esta inicialización puede hacerse en la misma línea de instrucción donde se la declara o en una instrucción separada, dependiendo del lenguaje.

En PseInt se realiza de manera separada.

```
1 Algoritmo inicializacion_de_variables
2
3 //Primero las declaro con nombre y tipo
4 Definir miPrimerVariable Como Real;
5 Definir otraVariable Como Real;
6
7 //Inicializo ambas variables
8 miPrimerVariable = 1;
9 otraVariable = 2;
```

```
1 Algoritmo inicializacion_de_variables
2
3 //Primero las declaro con nombre y tipo
4 Definir miPrimerVariable Como Real;
5 Definir otraVariable Como Caracter;
6
7 //Inicializo ambas variables
8 miPrimerVariable = 1;
9 + otraVariable = 2;
```

en inst. 1: No coinciden los tipos.

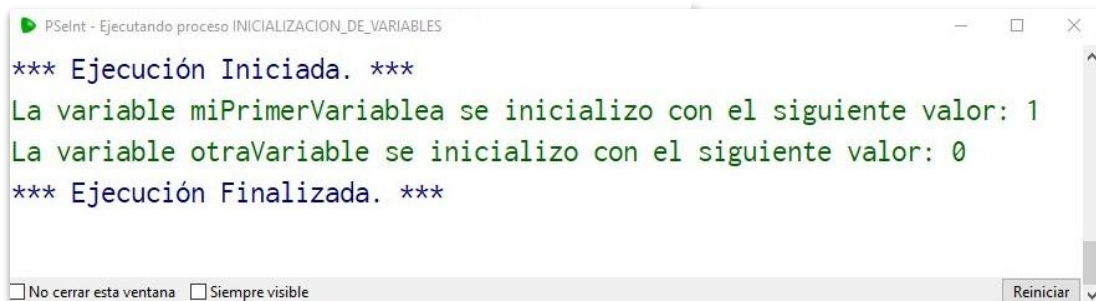
Inicialización de una variable

- Es importante tener en cuenta que dependiendo del lenguaje **las variables deben recibir un valor inicial antes de poder leer los datos que contienen.**
- En tal caso si un programador tratara de operar con variables que no han sido inicializadas o definidas, **los traductores normalmente reportarán un error y no compilarán el programa.**
- A este caso se lo llama como **variable sin inicializar o variable no definida.**



! PseInt en configuración flexible

```
1 Algoritmo inicializacion_de_variables
2
3     //Primero las declaro con nombre y tipo
4     Definir miPrimerVariable Como Real;
5     Definir otraVariable Como Real;
6
7     //Inicializo ambas variables
8     miPrimerVariable = 1;
9
10
11     //Pido al programa que me muestre el valor actual de la variable
12     Escribir "La variable miPrimerVariable se inicializo con el siguiente valor: ", miPrimerVariable;
13     Escribir "La variable otraVariable se inicializo con el siguiente valor: ", otraVariable;
14
15 FinAlgoritmo
```



PseInt - Ejecutando proceso INICIALIZACION_DE_VARIABLES

```
*** Ejecución Iniciada. ***
La variable miPrimerVariable se inicializo con el siguiente valor: 1
La variable otraVariable se inicializo con el siguiente valor: 0
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible Reiniciar

- **PseInt en flexible** compila, no arrojando error, reserva un espacio en memoria a la espera de inicializarla, por defecto inicializa la variable en 0.

Ámbito de una variable

Alcance de una variable.

Global o local, la diferencia entre ambas es hasta donde se pueda leer la información o valor de una variable.



De alcance global



De alcance local

¿Qué significa compilar?

- **Como lo vimos en la clase 2, compilar** es el proceso de **transformar** de **un lenguaje a otro**.
- Al programa que se encarga de compilar se le llama **compilador**. A veces, a esta tarea se le llama "ensamblar" o "construir", lo que suele implicar otros procesos adicionales, e.j. empaquetarlo en formato binario.
- Normalmente, un compilador **transforma un lenguaje de alto nivel** como **C o Java**, el cual es **legible por los humanos, en un lenguaje máquina que la CPU puede entender**.
- Algunos compiladores que traducen **de un lenguaje a otro del mismo nivel son llamados transpiladores**, por ejemplo al compilar de TypeScript a JavaScript.

Relación entre Variables y Datos

Datos en la informática

Para la computadora cualquier dato es un número binario. La diferencia se da a alto nivel, en cuanto a **cómo el programador interactúa** con estos datos.

Fuente: C. Cimino



A los fines de este curso podemos decir que:

El dato cambia, la variable no... Podrá cambiar su valor pero no su identidad. Una constante y su valor tampoco cambian.

Esquema de los tipos de datos



Tipos de datos que pueden contener las variables

Los tipos de datos que podemos encontrar frecuentemente, podemos decir que son:

- **Numéricos** (2, 89, 5.5, -3, 0)
- **Alfanuméricos** (“Hola”, “e”, ‘Jugador45’, ‘2,89’)
- **Booleanos** (True, False)

Datos numéricos

Todo tipo de dato **representado por un número, en general será numérico**. Ya sea que tenga valor positivo, negativo, entero o decimal.

En **PseInt** los tipos numéricos son:

- **Entero:** admite datos numéricos que sean enteros. Ejemplo: -26; 200.
- **Real** admite datos numéricos, que pueden ser enteros y/o decimales. Ejemplo: -2.5; 0.25.



Declaración de una variable numérica en PseInt

Sintaxis general

Definir <nombre> **Como** <tipo__de__dato>;

En PseInt:

Definir miEdad **Como** **Entero**;

Definir miEstatura **Como** **Real**;

Datos alfanuméricos

- El **caracter**, que puede ser **una letra, un símbolo o un número entre comillas**, es un dato **alfanumérico**.
- Cuando **agrupamos más de un carácter** se forma una **cadena de texto**, y se los puede almacenar **dentro de una variable** tipo **Caracter**.
- Ejemplos pueden ser: “Hola”, “HoL4”, “5+4 es nueve”, “15,26”.



Declaración de una variable alfanumérica en PseInt

Sintaxis general

Definir <identificador> **Como** <tipo__de__dato>;

En PseInt:

Definir miNombre **Como** **Caracter**;

Datos lógicos o booleanos

- Este tipo de dato puede representar sólo **dos valores: Verdadero o Falso**.
- **Son almacenados dentro de una variable de tipo Boolean**, en PseInt se la define como **Lógico**.
- Normalmente usadas para **representar las alternativas de veracidad a condiciones o preguntas lógicas**, lo veremos en el rombo de decisiones en el diagrama de flujo.

```
1 Algoritmo inicializacion_de_variables
2
3 //Primero las declaro con nombre y tipo
4 Definir miPrimerVariable Como Logico;
5
6
7 //Inicializo ambas variables
8 miPrimerVariable = Verdadero;
9
10
11 //Pido al programa que me muestre el valor actual de la variable
12 Escribir "La variable miPrimerVariable se inicializo con el siguiente valor: ", miPrimerVariable;
13
14 FinAlgoritmo
15
```

PSeInt - Ejecutando proceso INICIALIZACION_DE_VARIABLES

```
*** Ejecución Iniciada. ***
La variable miPrimerVariable se inicializo con el siguiente valor: VERDADERO
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible Reiniciar

Declaración de una variable lógica en PseInt

Sintaxis general

Definir <identificador> **Como** <tipo__de__dato>;

En PseInt:

Definir miVariable **Como** **Logico**;

Relación entre constantes y variables

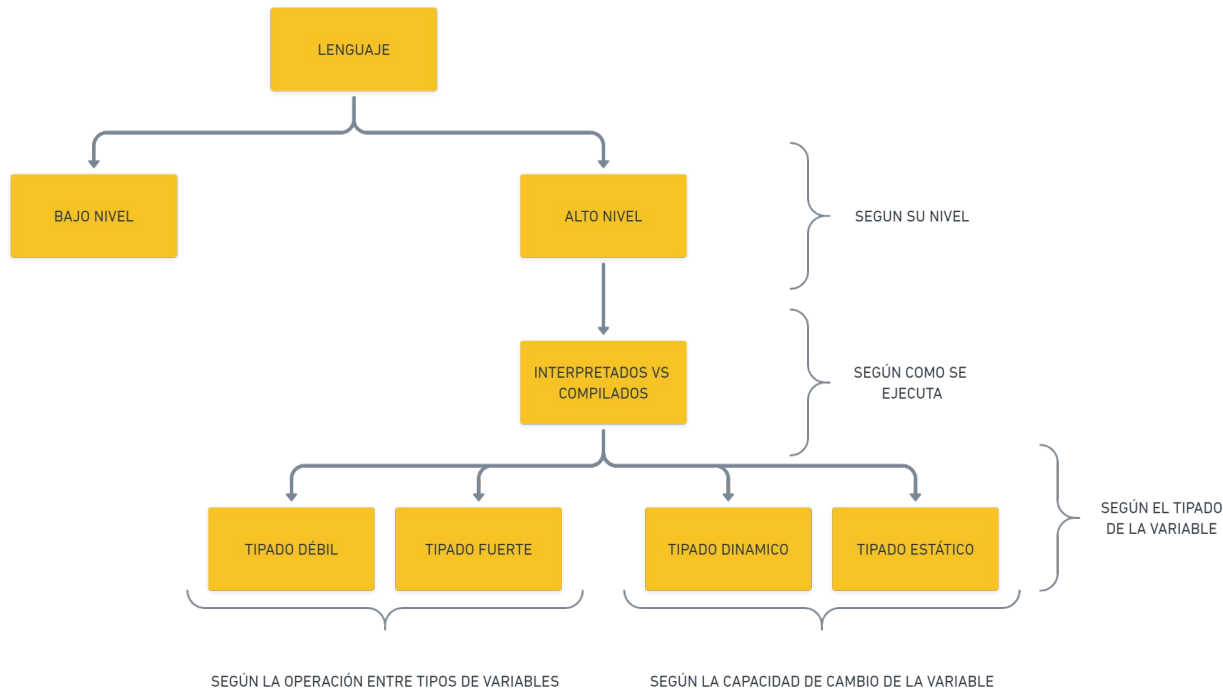
- Una variable puede ser una constante y finalmente sigue siendo un elemento de almacenamiento de datos.
- Representan una dirección de memoria en donde se almacena un dato **pero que no varía durante la ejecución del programa.**
- Por ejemplo el número pi, como el mismo no varía, se puede definir una constante MINUMEROPI y asignarle el valor 3.14.
- **En general los lenguajes ya traen consigo sus constantes, es decir no es necesario crearlas, sino que simplemente basta con llamarlas.**



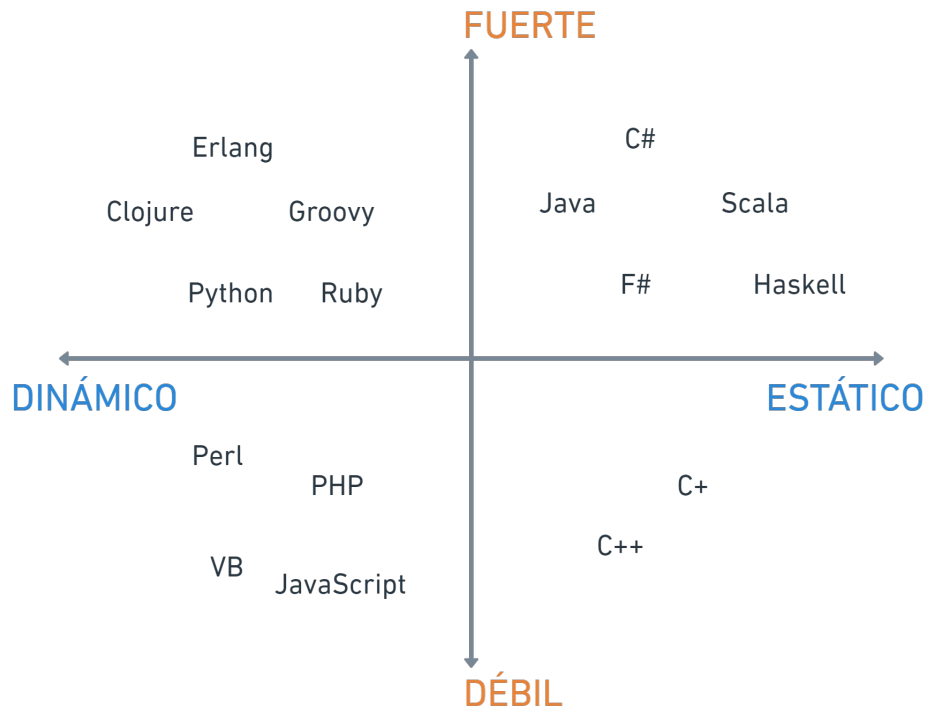
Lenguajes, ejecución y tipado

Cómo se relacionan

Lenguaje y variables - Clasificaciones



Clasificación de Lenguajes



Variables, según tipado en los lenguajes

- **Fuertemente tipados**, no podemos operar en distintos tipos de datos. Ej. "23"+4 da error.
- **Débilmente tipados**, el intérprete busca la mejor manera de dar un resultado a operaciones con distintos tipos. Ej "23"+4 da "234"
- **Tipado estático**, las variables se definen antes de usarlas y no cambiarán más.
- **Tipado dinámico**, el tipo depende del valor asignado.



Desafíos de Clase

- Con los siguientes desafíos afianzaremos los conocimientos sobre variables, asignación y operaciones varias.
- Ejercitaremos la estructuración lógica y pensamiento computacional.
- Tendremos en cuenta las buenas prácticas.

Desafíos

Desafío I de clase

1. Almacenar un nombre en una variable para que la computadora muestre un saludo por pantalla, utilizando ese nombre como parte del saludo, por ejemplo,

> Hola

> Carlos

Donde "Carlos" es el dato ingresado por el usuario.

Desafío II de clase

1. Almacenar un refrán en una variable y mostrarlo por pantalla.

Desafío para la próxima clase

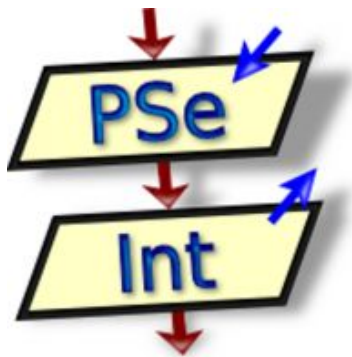
La clase pasada logramos pasar nuestro algoritmo a PSeInt

Para integrar los contenidos de esta clase se te pide:

- Identificá las variables que utilizás en tu algoritmo.
- Inicializalas al principio, comenzando a hacer uso de las buenas prácticas.
- Refactorizá tu código con estas nuevas variables.



Herramientas que utilizamos en clases



PSeInt es una herramienta para asistir a un estudiante en sus primeros pasos en programación

No te olvides de dar el presente

Recordá:

- **Revisar la Cartelera de Novedades.**
- **Hacer tus consultas en el Foro.**
- **Realizá los ejercicios obligatorios.**

Todo en el Aula Virtual.