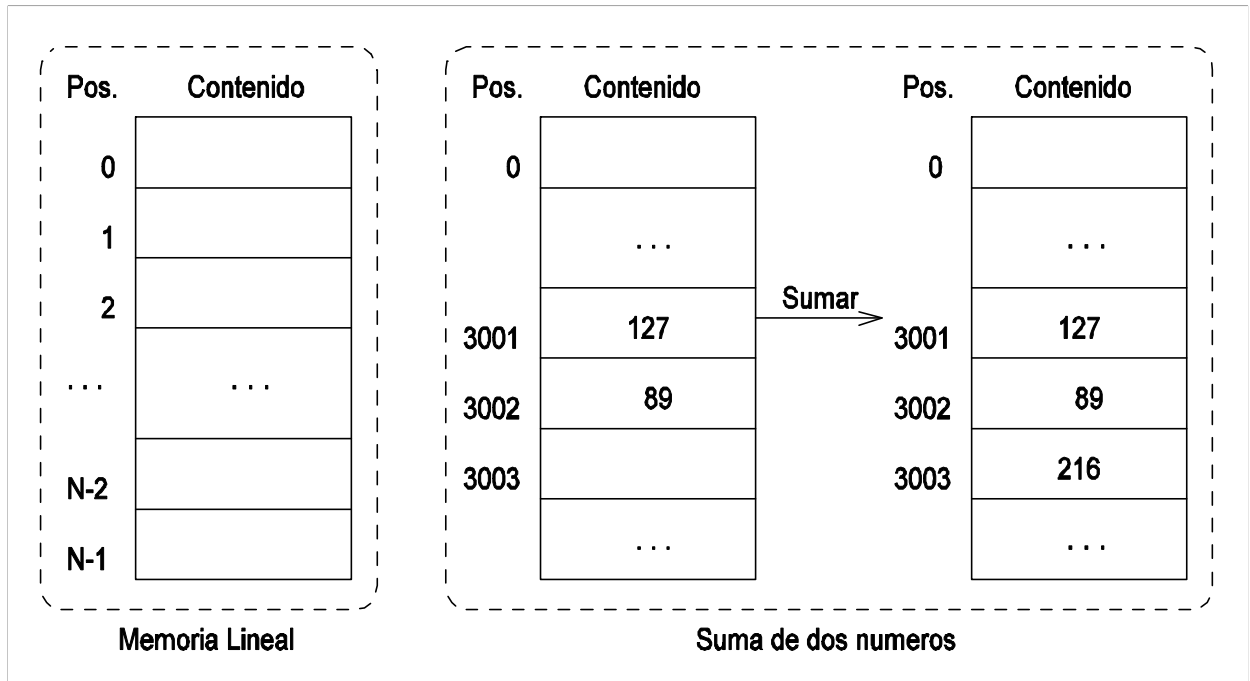


# Variables

Una variable no es más que un nombre simbólico que identifica una dirección de memoria:



“Suma el contenido de la posición 3001 y la 3002 y lo almacenas en la posición 3003”

vs.

$total = cantidad1 + cantidad2$

“Suma cantidad1 y cantidad2 y lo almacenas en total”

## Declaración de variables

Para usar una variable en un programa hay que declararla.

- El ordenador conoce así cómo codificar la información que se va a almacenar en la posición de memoria correspondiente.
- Al declarar una variable, se reserva el espacio de memoria necesario para almacenar un valor del tipo de la variable.
- El identificador asociado a la variable se puede utilizar para acceder al dato almacenado en memoria y para modificarlo.

## *Declaración de variables en Java*

```
<tipo> identificador;  
  
<tipo> lista de identificadores;
```

- Las variables se han de declarar antes de poder usarlas.
- Los identificadores de las variables son los nombres que utilizaremos para referirnos a ellas.
- Al declarar una variable, hay que definir su tipo: la variable sólo admitirá valores del tipo especificado.
- En una misma declaración se pueden declarar varias variables, siempre que sean del mismo tipo. En este caso, los identificadores de las variables se separan por comas.

### *Ejemplos*

```
// Declaración una variable entera x de tipo int  
int x;
```

```
// Declaración de una variable real r de tipo double  
double r;
```

```
// Declaración de una variable c de tipo char  
char c;
```

```
// Múltiples declaraciones en una sola línea  
int i, j, k;
```

## Identificadores en Java

- El primer símbolo del identificador será un carácter alfabético (a, ..., z, A, ..., Z, '\_', '\$') pero no un dígito. Después de ese primer carácter, podremos poner caracteres alfanuméricos (a, ..., z) y (0, 1, ..., 9), signos de dólar '\$' o guiones de subrayado '\_'.
- Los identificadores no pueden coincidir con las palabras reservadas, que ya tienen significado en Java:

abstract	continue	for	new	switch
boolean	default	goto	null	synchronized
break	do	if	package	this
byte	double	implements	private	threadsafe
byvalue	else	import	protected	throw[s]
case	extends	instanceof	public	transient
catch	false	int	return	true
char	final	interface	short	try
class	finally	long	static	void
const	float	native	super	while
cast	future	generic	inner	
operator	outer	rest	var	

- Las mayúsculas y las minúsculas se consideran diferentes.
- '\$' y '\_' se interpretan como una letra más.
- '\$' no se suele utilizar en la práctica (lo usa el compilador).

### *Ejemplos válidos*

a, pepe, r456, tu\_re\_da, AnTeNa, antena, usd\$

### *Ejemplos no válidos*

345abc, mi variable, Nombre.Largo, cañada, camión

Java es un lenguaje sensible a mayúsculas/minúsculas.
---

## Convenciones

- Los identificadores deben ser descriptivos: deben hacer referencia al significado de aquello a lo que se refieren.

```
int n1, n2;           // MAL
int anchura, altura; // BIEN
```

- Los identificadores asociados a las variables se suelen poner en minúsculas.

```
int CoNTaDoR;       // MAL
int contador;       // BIEN
```

- Cuando el identificador está formado por varias palabras, la primera palabra va en minúsculas y el resto de palabras se inician con una letra mayúscula.

```
int mayorvalor;     // MAL
int mayor_valor;   // ACEPTABLE
int mayorValor;     // MEJOR
```

## Inicialización de las variables

En una declaración, las variables se pueden inicializar:

```
int i = 0;

float pi = 3.1415927f;

double x = 1.0, y = 1.0;
```

**NOTA:** La inicialización puede consistir en una expresión compleja que se evalúa cuando se ejecuta el programa.

## *Definición de constantes en Java*

Una constante hace referencia a un valor que no puede modificarse

```
final <tipo> identificador = <valor>;
```

- Las constantes se definen igual que cuando se declara una variable y se inicializa su valor.
- Con la palabra reservada final se impide la modificación del valor almacenado

## **Convenciones**

- Los identificadores asociados a las constantes se suelen poner en mayúsculas.

```
final double PI = 3.141592;
```

- Si el identificador está formado por varias palabras, las distintas palabras se separan con un guión de subrayado

```
final char RETORNO_DE_CARRO = '\n';
```

```
final double ELECTRONIC_CHARGE = 1.6E-19;
```

---

Si intentásemos modificar el valor de una constante, el compilador nos daría un error:

```
RETORNO_DE_CARRO = '\r'; // Ilegal
```

produce el siguiente error

*“Cannot assign a value to final variable RETORNO\_DE\_CARRO”*