

# Objetos de un programa

## **Datos:**

Se denomina dato a toda aquella información característica de una entidad y que es susceptible de tratamiento en un programa informático.

## **Características:**

- Identificador
- Tipo
- Valor

# Objetos de un programa

**Identificador:** es el nombre que usa el programador para referenciar los datos y otros elementos del programa, permitiendo así su definición en una posición de la memoria del ordenador.

## **Normas:**

- Pueden estar constituidos por letras y dígitos y en algunos casos el guión bajo (\_)
- Deben comenzar por una letra.
- No deben contener espacios.
- El número máximo de caracteres y nombres reservados que se pueden emplear dependen del compilador utilizado.
- El nombre asignado debe tener relación con la información que contiene, pudiéndose emplear abreviaturas que sean significativas.

# Objetos de un programa

**Tipo:** rango de valores (dominio) que puede tomar el dato, determina el espacio de memoria reservado y operaciones definidas sobre ese dominio.

**Valor:** representa el elemento determinado perteneciente al rango de valores según su tipo y que estará contenido en el espacio de memoria reservado.

# Objetos de un programa

**Clasificación de los datos:** según el tamaño de la información en la memoria del ordenador.

Bit	Unidad mínima para la representación de la información, que sólo puede tomar los valores de 0 y 1
Byte u Octeto	Conjunto de ocho bits, que sirve de referencia en la compartición de la memoria del ordenador y se utiliza para representar un carácter
Palabra	Conjunto de bits que pueden ser manipulados por el ordenador en una sola operación. Pudiendo ser de 8, 16, 32 o 64 bits, dependiendo del procesador utilizado

# Objetos de un programa

**Clasificación de los datos:** según la representación de la información para su almacenamiento y tratamiento.

Datos Básicos	Numéricos	Entero		
		Real		
	Carácter			
	Lógico			
Datos Derivados	Punteros			
Datos Estructurados	Internos	Estáticos	Lineales	Tabla
		Dinámicos	Lineales	Lista
				Pila
			No Lineales	Cola
				Árbol
			Grafo	
	Externos	Fichero		
	Base de datos			
Compuesto	Estructura de datos o registro			

# Objetos de un programa

**Clasificación de los datos:** según la permanencia de su contenido o valor en memoria durante la ejecución del programa.

**Constantes**

**Variables**

# Objetos de un programa

## Datos básicos

**Numéricos:** sirven para contener magnitudes.

- a. **Numérico entero:** Se utiliza para representar números enteros (con y sin signo), se expresa mediante dígitos (0 a 9), pudiendo estar precedido por el signo de + o de -.

Dominio:  $\{-32768; \dots; 32767\}$

<u>Operaciones:</u>	+	suma
	-	resta
	*	multiplicación
	/	división entera
	mod	resto de la división

# Objetos de un programa

## Datos básicos

**Numéricos:** sirven para contener magnitudes.

### a. **Numérico entero:** (cont.)

Funciones: dependen del compilador utilizado.

Por ejemplo en C++: potencia:  $3^5$  `pow(3,5)`

Expresiones: se pueden combinar operaciones con funciones:

`3 * 2 + pow(4,3)`

cálculo de la hipotenusa en C++:

`hip=sqrt(pow(cat1,2)+pow(cat2,2))`



# Objetos de un programa

## Datos básicos

**Numéricos:** sirven para contener magnitudes.

**b. Numérico real:** Se utiliza para representar los números con parte decimal o los números muy grandes o muy pequeños que no pueden ser contenidos en un número entero.

Dominio:  $\mathcal{R}$

<u>Operaciones:</u>	+	suma
	-	resta
	*	multiplicación
	/	división (div)

# Objetos de un programa

## Datos básicos

**b. Numérico real:** Puede ser expresado de 2 formas:

- Punto decimal: Utiliza dígitos del 0 al 9 con su signo correspondiente y un punto (coma) para separar la parte entera de la fraccionaria.
- Notación científica o exponencial: utiliza el formato “mantisaEcaracterística”.

Mantisa= número real.

E= representación de la base decimal.

Característica: exponente correspondiente a un número entero con su signo.

# Objetos de un programa

## Datos básicos

- c. **Character:** Un solo byte, contiene un (1) carácter. Incluye el juego de caracteres disponibles en el ordenador (varía de un ordenador a otro), se corresponde con un número entero con o sin signo.

Ej. en C++: unsigned char 0 a 255

char -128 a 127

Los valores se escriben entre comillas simples.

Ej. 'a'

# Objetos de un programa

## Datos básicos

### c. **Character:** (cont.)

Operaciones: Si bien no existen operaciones internas entre caracteres, en C++ si defino una variable tipo *char* y luego la incremento, me muestra el carácter siguiente según el código que esté utilizando.

Por ej.: `char a= 'a'; //(Código ASCII 97)`

`a= a+1; // también a+=1;`

`cout<<a; //aquí me muestra el carácter 'b'.`

# Objetos de un programa

## Datos básicos

### c. **Character:** (cont.)

Funciones: En C++ existe una serie de funciones que pertenecen a las bibliotecas `string.h` y `ctype.h` que nos permite manipular las cadenas de caracteres.

En C++ no existe el tipo cadena, por lo cual se implementa mediante un tipo de dato estructurado (vector) que contiene datos de tipo `char`.

Ej. `char nombre[15];`

Operaciones:

Operadores relacionales:

- == igual
- != distinto (<>)
- < menor
- <= menor o igual
- > mayor
- >= mayor o igual

# Objetos de un programa

## Datos derivados

- a. **Punteros:** Se utiliza para contener la dirección de memoria de otra variable y debe ser definida con el mismo tipo de la variable que va a referenciar o a apuntar.

### Características:

- Definir las variables de tipo puntero con el mismo tipo de dato que las variables a las que va a apuntar.
- Asignar un contenido a la variable puntero: dirección de memoria (previamente definida) donde se encuentra ubicada la variable que se va a apuntar.
- Puede hacerse una referencia indirecta al valor de la variable a través de un puntero.
- Se puede reutilizar el puntero para apuntar a otra variable.

# Objetos de un programa

## Datos estructurados

### **Internos y externos:**

Internos: son los que residen en la memoria principal del ordenador. Por ej. vector (tabla) unidimensional de números enteros.

Externos: son los que residen en un soporte externo a la memoria principal (memoria auxiliar) Por ej. archivo en disco magnético.

### **Estáticos y dinámicos:**

Estáticos: su tamaño queda definido en la compilación y no se puede modificar durante la ejecución. Por ej. vector.

Dinámicos: su tamaño puede ser modificado durante la ejecución del programa. Por ej. lista encadenada.



# Objetos de un programa

## Datos estructurados

### Lineales y no lineales:

Lineales: son los que pueden ser enlazados sólo con un elemento anterior y uno posterior. Por ej. cola.

No lineales: pueden enlazarse con más de un elemento anterior y más de un elemento posterior. Por ej. árbol.

### Compuestos:

Son los formados por el programador utilizando tipos básicos y tipos derivados. Pueden ser internos (estructuras) o externos (archivo con diversos campos)

# Objetos de un programa

## Constantes y variables

**Constantes:** son aquellas cuya información es fija durante la ejecución del programa. Se pueden expresar de dos formas:

- a. De forma explícita mediante su valor.
- b. Utilizando un identificador para definir la constante en memoria asignándole un valor.

Ej. en C++:

```
int Pi=3,1416
```

```
# DEFINE Pi=3,1416
```

En el último caso el compilador asigna un espacio en memoria con el tamaño correspondiente al tipo de dato expresado por su valor.

# Objetos de un programa

## Constantes y variables

**Variables:** Son datos cuya información puede ser variable durante la ejecución del programa.

Se *definen* con un identificador y un tipo de dato.

El identificador permite hacer referencia a la variable en el programa.

El tipo de datos permite determinar el tamaño de la variable en la memoria y el conjunto de operaciones y funciones que se pueden utilizar sobre el mismo.

# Objetos de un programa

## Operadores

OPERADOR	SÍMBOLO	SIGNIFICADO
Paréntesis	()	Paréntesis
Aritméticos	**	Potencia
	*	Producto
	/	División
	div	División entera
	%, mod	Módulo o resto
	+	Signo positivo o suma
	-	Signo negativo o resta
Alfanuméricos	+	Concatenación
	-	Concatenación eliminando espacios

# Objetos de un programa

## Operadores

OPERADOR	SÍMBOLO	SIGNIFICADO
Relacionales	==, =	Igual a, asignación
	!=, <>	Distinto a
	<	Menor que
	>	Mayor que
	<=	Menor o igual que
	>=	Mayor o igual que
Lógicos	!, NOT, NO	Negación
	&&, AND, Y	Conjunción
	, OR, O	Disyunción

# Objetos de un programa

## Tablas de verdad

Para el operador AND

A	B	A AND B
V	V	V
V	F	F
F	V	F
F	F	F

Para el operador OR

A	B	A OR B
V	V	V
V	F	V
F	V	V
F	F	F

Para el operador NOT

A	NOT A
V	F
F	V

# Objetos de un programa

## Orden de prioridad de los operadores

1. Paréntesis (comenzando por los más externos)
2. Signo.
3. Negación.
4. Potencia.
5. Producto, división y módulo.
6. Suma y resta.
7. Concatenación.
8. Relacionales.
9. Conjunción (AND).
10. Disyunción (OR).

Los operadores con la misma prioridad se evalúan de izquierda a derecha.

# Objetos de un programa

**Expresiones:** Conjunto de datos (operandos) y operadores con unas reglas específicas de construcción.

Los operandos pueden ser también valores retornados por funciones. En la obtención del resultado se debe tener en cuenta el orden de prioridad de los operadores.

## **Clasificación:**

Numéricas: su resultado es numérico y utilizan operadores y operandos numéricos. Deben ser escritas en formato algorítmico para que puedan ser interpretadas por el ordenador.

Alfanuméricas: su resultado es una cadena de caracteres y utilizan operadores alfanuméricos.

Lógicas o booleanas: su resultado es Verdadero (V) o Falso (F) y utilizan operadores relacionales y lógicos.