

VARIABLES Y TIPOS DE DATOS

INTRODUCCIÓN A LA PROGRAMACIÓN

PROGRAMAS QUE TRABAJAN CON DATOS

- Los programas o aplicaciones trabajan con **datos**
 - Elementos de entrada o elementos de salida
- Ejemplo
 - Buscar un apartamento para pasar las vacaciones
 - Entrada: zona, fechas, rango de precios, características
 - Salida: lista de apartamentos, precios, condiciones, ...

DATOS

- **Piezas de información** que vamos a manipular
- En los pasos de un algoritmo, se pueden **crear** nuevos datos, **modificar** otros, **transformar** en otros, ...
- Ejemplo:
 - Al jugar al ahorcado vamos añadiendo 1 al número de errores si no acertamos la palabra.

TIPOS DE DATOS

- *Todos los datos no son iguales*
- Ejemplo:
 - El volumen de un motor se puede medir con un número
 - El asunto de un email es un texto
 - El resultado de comprobar si nos ha tocado la lotería es un valor VERDADERO o FALSO (booleano)
- Por ello, debemos elegir entre una serie de **tipos de datos**

TIPOS DE DATOS

- **Cada lenguaje de programación** puede tener sus propios tipos de datos
- Los programadores pueden **crear nuevos tipos de datos** (sobre todo en Programación Orientada a Objetos)
- Hay un conjunto de tipos de datos comunes a casi todos los lenguajes de programación: los **tipos primitivos**

TIPOS DE DATOS PRIMITIVOS

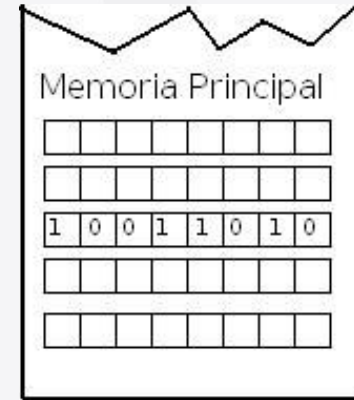
- Tipos **numéricos**: representan números: positivos y/o negativos; con o sin decimales; pequeños, medianos o grandes, ...
 - Se pueden hacer operaciones matemáticas con ellos.
 - Ejemplo: -1, 3.14, 12037410238471039427039174038
- Tipos **carácter**: representan texto: letras sueltas, palabras completas, textos más largos...
 - No se pueden hacer operaciones matemáticas con ellos.
 - Se suelen representar entre comillas.
 - Ejemplo: 'a', "Hola", "En un lugar de La Mancha..."

TIPOS DE DATOS PRIMITIVOS

- Tipos **lógicos**: pueden tomar dos valores, VERDADERO o FALSO
 - Suelen ser el resultado de una comparación de otros datos
 - Ejemplos:
 - ¿Es 1 igual a -1? FALSO.
 - ¿Es Pi menor que 4? VERDADERO.

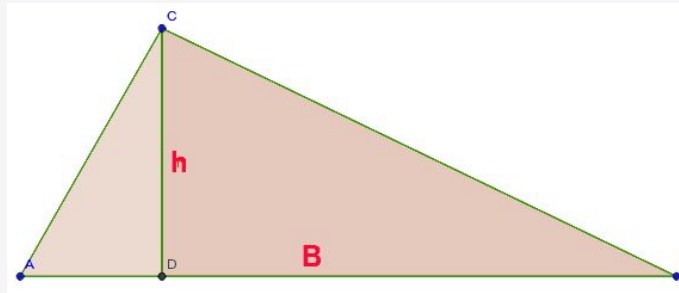
VARIABLES

- Los algoritmos de los programas trabajan con **datos**
- Los **valores concretos** que toman estos datos se almacenan en **variables**
- Una variable es una **región de memoria** de nuestro ordenador, donde vamos a guardar **temporalmente** un dato



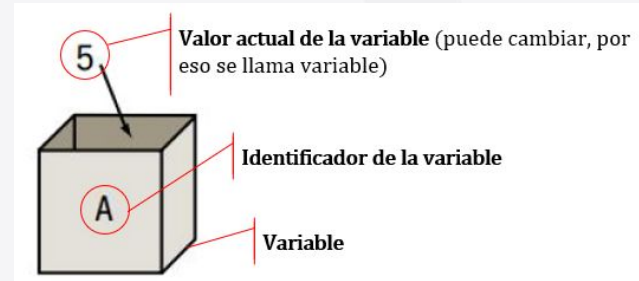
VARIABLES

- Ejemplo: Para el cálculo del área de un triángulo necesitamos tres variables
 - Una para guardar la *altura*
 - Otra para guardar la *base*
 - Otra para guardar el área resultante: $area = (base * altura) / 2$



VARIABLES

- Una variable queda definida por
 - Un **identificador**
 - Un **tipo de dato**
 - Un **valor actual**



~~VARIABLES~~ CONSTANTES

- Cuando queremos almacenar un valor en memoria que no se va a modificar a lo largo del programa, creamos una **constante**.
- Ejemplos
 - Pi: 3.1415
 - Número de meses del año: 12
 - Factor de conversión de litros a metros cúbicos: 0.001

ASIGNACIÓN DE UN VALOR A UNA VARIABLE

- Es la operación de dar un valor inicial, o de cambiar el valor actual de una variable
 - Ejemplos:
 - `altura = 27` (asignación a través de un valor literal)
 - `var2 = var1` (asignación a través de otra variable)
 - `area = base * altura` (asignación a través de una expresión)

OPERACIONES CON VARIABLES

- Los algoritmos requieren que realicemos operaciones con variables para conseguir su objetivo
 - Fórmulas matemáticas
 - Comparaciones
 - Operaciones lógicas
 - ...
- Estas operaciones se realizan usando **operadores**
- La mayoría son comunes a todos los lenguajes de programación

OPERADORES MATEMÁTICOS (ARITMÉTICOS)

- Suma (+): $a + 7$
- Resta (-): $n2 - n1$
- Multiplicación (*): $3 * 4$
- División (DIV, /): b / c
- Resto(MOD, %): $a \% 7$
- Potencia (^): b^2
- ...

OPERADORES DE COMPARACIÓN (RELACIONALES)

- Mayor que ($>$), mayor o igual que ($>=$)
- Menor que ($<$), menor o igual que ($<=$)
- Distinto ($!=$, $<>$)
- El resultado de la comparación es un valor booleano (*true*, *false*)
 - Ejemplos
 - $a >= 10$ (si a vale 11) devuelve *true*
 - $b != c$ (si b vale "Hola" y c vale "Hola") devuelve *false*

OPERADORES LÓGICOS

- Toman operandos de tipo lógico y producen un resultado de tipo lógico
- Conjunción (Y, AND, &&)
- Disyunción (O, OR, ||)
- Negación (NO, NOT, !)
- Ejemplos:
 - $(3 > 2) \text{ AND } (3 > 5) \Rightarrow \text{VERDADERO AND FALSO} \Rightarrow \text{FALSO}$
 - $(3 > 2) \text{ AND } (3 > 5) \Rightarrow \text{VERDADERO OR FALSO} \Rightarrow \text{VERDADERO}$

Puedes consultar [aquí](#) las tablas de verdad de estos operadores

RECAPITULAMOS

- Nuestros algoritmos (programas) trabajan con **datos**
- Los datos pueden ser de tipo **numérico**, de **texto** o **lógicos**
- Los datos son almacenados en **variables**, definidos por un **identificador**, un **tipo de dato** y un **valor actual**
- Las variables que no cambian en todo el programa se llaman **constantes**

RECAPITULAMOS

- Podemos crear expresiones o fórmulas con variables usando **operadores**
- Los operadores **aritméticos** sirven para realizar operaciones matemáticas
- Los operadores **relacionales** sirven para comparar datos
- Los operadores **lógicos** reciben operandos de tipo booleano, y producen resultados también de este tipo