



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

VARIABLES



¿Qué es una variable?

- Es un espacio en memoria que el programador reserva para guardar un dato que recibe y que necesitará utilizar más tarde
- La variable es una porción de memoria que ningún otro programa podrá utilizar
- Para el compilador cada variable es una dirección de memoria específica en donde se aloja el datos guardado



¿Qué significa crear una variable?

- Las variables se crean al momento de ejecutar nuestra aplicación.
- El compilador reserva un espacio libre en memoria y lo apunta a nuestra variable para que cada vez que la utilicemos vaya a dicha dirección
- La porción (extensión) de memoria que reserva dependerá del dato que deseamos guardar
- El valor almacenado puede cambiar en el transcurso de la ejecución del programa, pero siempre serán valores del tipo de dato al que pertenecen.



¿Qué es un tipo de dato?

- Las variables se definen por los tipos de datos que indicamos al crearlas.
- Un tipo de dato define:
 - *el tamaño que dicho tipo va a ocupar en la memoria*
 - *el rango de valores que puede almacenar dicho tipo*
 - *la forma en que se almacenan en memoria los diferentes valores*
 - *las operaciones que pueden realizarse con él*



¿Qué tipos de datos usa C?

○ Numéricos:

- *Enteros sin signo: para representar números enteros que no pueden ser negativos.*
- *Enteros con signo: para representar números enteros que pueden ser tanto positivos como negativos.*
- *Decimales: para representar cantidades con decimales.*

○ Lógicos

○ Caracteres



Enteros Sin Signo

- *unsigned char*
- *unsigned short int o unsigned short*
- *unsigned int o unsigned*
- *unsigned long int o unsigned long*
- *unsigned long long int o unsigned long long*



Enteros Con Signo

- *signed char*
- *short int* o simplemente *short*
- *int*
- *long int* o simplemente *long*
- *long long int* o simplemente *long long*



Decimales

- *float*
- *double*
- *long double*



Lógicos

- *Bool*

Habitualmente se utilizan los enteros para representar datos booleanos (que sólo tienen dos valores: VERDADERO y FALSO), utilizando el siguiente criterio:

- *Se considera VERDADERO para cualquier valor distinto de cero*
- *Se considera FALSO si es cero*



Caracteres

- *Char*
- *Almacenan 1 dígito correspondiente a:*
 - Caracteres alfabéticos: a b c ... z A B C ... Z
 - Caracteres numéricos: 0 1 2 ... 9
 - Caracteres no imprimibles: espacio, tabulador, cambio de línea...
 - Caracteres especiales: + - * / . , ; < > \$?



Ámbito de las variables

- El ámbito de una variable es la posición de código donde la variable está disponible (desde dónde se puede acceder a ella)
- Cuando una variable está disponible en una porción de código, diremos que es visible.
- Estudiaremos dos ámbitos:
 - Global
 - Local



Variables Globales

- Se declaran fuera del **main()**
- Son visibles en todo el código que sigue a su declaración
- Existen durante toda la ejecución del programa



Variables Locales

- Se declaran dentro del main o de una función (al principio del cuerpo)
- Sólo son visibles dentro del cuerpo de código al que pertenecen
- Se crean automáticamente cuando comienza la ejecución del bloque
- Se destruyen automáticamente cuando se termina la ejecución del bloque

Constantes

- El programador puede definir una variable constante la cual no podrá variar su valor una vez asignado

Para crear una variable float correspondiente a la cotización dólar (la cual no puede modificarse durante el uso de la aplicación) escribimos:

```
const float cotizacion = 5.13;
```



Constantes

- Se puede usar const antes o después del tipo de dato
- Generalmente se inicializa la variable al crearla dado que no podrá cambiarse de alguna otra forma
- La directiva del preprocesador `#define` es similar a la utilización de constantes sólo que NO actúa como una variable.
- Al utilizar `#define` no se reserva espacio en memoria para un dato sino que al momento de compilar se reemplaza el nombre definido por el valor indicado



Trabajando con variables

- Creación de variables
 - **int edad, mes, anio;**
 - **char sexo;**
 - **float sueldo;**
- Almacenamiento de valores
 - **edad = 22;**
 - **sexo = 'F';**
 - **sueldo = 4028.50;**

Trabajando con variables

- Almacenamiento de un valor introducido por teclado
 - `scanf ("%d",&edad);`
 - `scanf ("%c %f",&sexo,&sueldo);`
- Impresión del valor almacenado en una variable
 - `Printf ("La edad introducida es %d",edad);`
 - `Printf ("El sueldo promedio para un empleado de sexo %c es %f",sexo,sueldo);`

String

- Un string es un tipo de dato particular formado por una secuencia de caracteres agrupados que se utiliza para almacenar palabras.
- Posee características particulares que lo diferencian de las variables hasta ahora vistas:
 - Se crean indicando la cantidad de caracteres contenidos: `char nombre[20];`
 - Se le asigna un valor: `nombre="Juan"`
 - El nombre de la variable es la dirección de memoria donde fue alojado y por lo tanto se almacena desde el teclado como `scanf ("%s", nombre);`
 - Se imprime por teclado como `printf ("Su nombre es %s", nombre);`