

Expresiones

Expresiones

- ▶ Las expresiones o instrucciones son combinaciones de operandos, operadores y paréntesis, que al evaluarse dan como resultado un valor. Cada operando puede ser:
 - Variables
 - Constantes (valores fijos)
 - Funciones
 - Otra expresión (definida entre () como un subconjunto de la expresión completa)

Operadores

- ▶ Determinan las operaciones a realizar. Pueden ser:
 - Aritméticos
 - Relacionales
 - Lógicos
 - De manejo de bits
 - De asignación
 - De asignación compacta
 - De autoincremento y autodecremento
 - Condicional

Operadores Aritméticos

- ▶ Trabajan con valores numéricos y producen como resultado un valor numérico
- ▶ Operadores unarios (que se aplican a un único operando):
 - **-** : convierte en positivos los valores negativos, y en negativos los positivos.
- ▶ Operadores binarios (que se aplican a dos operandos):
 - **+** : suma dos valores numéricos
 - **-** : resta dos valores numéricos
 - ***** : multiplica dos valores numéricos
 - **/** : divide dos valores numéricos. Si ambos operandos son enteros, el resultado de la división será un valor entero (despreciando el resto); en cualquier otro caso, la división es con decimales
 - **%** : operador módulo; devuelve el resto resultante de dividir el operando de la izquierda por el de la derecha. Ambos operandos deben ser necesariamente valores de tipo entero. El resultado también es entero.

Operadores Relacionales

- ▶ Evalúan una comparación entre valores numéricos y devuelven un entero correspondiente al resultado de la comparación

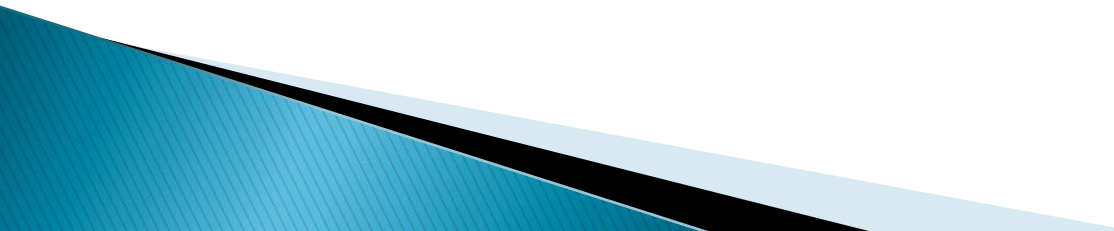
- **<** : compara si el valor de la izquierda es menor que el de la derecha
- **>** : compara si el valor de la izquierda es mayor que el de la derecha
- **<=** : compara si el valor de la izquierda es menor o igual que el de la derecha
- **>=** : compara si el de la izquierda es mayor que el de la derecha
- **==** : compara si ambos valores son iguales
- **!=** : compara si ambos valores son distintos uno del otro

Recordemos que el resultado será 1 en caso VERDADERO y 0 si es FALSO

Operadores Lógicos

- ▶ Se utiliza para evaluar la ocurrencia de condiciones, sin embargo para el compilador compara operando numéricos dando como resultado un valor entero (0 o 1, V o F)
 - **!** : operador unario; niega, contradice el valor del operando. Devuelve FALSO para cualquier valor distinto de 0
 - **&&**: operador binario; verifica que ambos operandos resulten VERDADEROS. Si el operando de la izquierda es 0 (FALSO), el operando de la derecha no se evalúa y el resultado de la operación es FALSO
 - **||**: operador binario; verifica que al menos uno de los dos operandos sea VERDADERO. Si el operando de la izquierda vale distinto de cero, el operando de la derecha no se evalúa y el resultado de la operación es VERDADERO

Operadores de Asignación

- ▶ El operador de asignación admite dos operandos; el de la izquierda debe ser una **variable**, mientras que el de la derecha puede ser cualquier expresión, del mismo tipo que la variable.
 - ▶ Se representa por el símbolo =
 - ▶ Permite asignar un valor a una variable.
- 

Operadores de Asignación Compacta

- ▶ Permiten simultáneamente realizar una operación con una variable, y asignar el resultado a esa misma variable.

`+=, -=, *=, /=, %=`

- ▶ No debe aparecer ningún espacio en blanco entre el operador y el símbolo =
- ▶ La forma **`var op= expresion`** es equivalente a **`var = var op (expresion)`**
- ▶ Se evalúan de derecha a izquierda

OPERADORES DE AUTOINCREMENTO Y AUTODECREMENTO

- ▶ Su único operando debe ser una variable.
- ▶ Los operadores son:
 - **++** : incrementa el valor de la variable en una unidad.
 - **--** : decrementa el valor de la variable en una unidad
- ▶ Si se utiliza el operador como prefijo, primero se modifica el valor de la variable (sumando o restando, dependiendo del operador) y se obtiene como valor de la expresión el valor ya modificado
- ▶ Si se utiliza como postfijo, primero se obtiene el valor de la variable (que es el resultado de la expresión) y posteriormente se modifica el valor de la variable

Operador Condicional

- ▶ Tiene tres argumentos numéricos
 - **expresion1 ? expresion2 : expresion3;**
- ▶ El resultado de la expresión será:
 - **expresion2**, si expresion1 es verdadera (distinta de 0), y en este caso no se evalúa el resto
 - **expresion3**, si expresion1 es falsa (igual a 0), y en este caso no se evalúa expresion2.
- ▶ Ejemplo:
2 + 3 == 5 ? “suma bien” : “error en la suma”;

Orden de prioridad de los operadores

Operadores	Asociatividad
() [] -> . (miembro)	izquierda a derecha
~ ! ++ -- sizeof (tipo) -(unario) *(indirección) & (dirección)	derecha a izquierda
* / %	izquierda a derecha
+ -	izquierda a derecha
<< >>	izquierda a derecha
< <= > >=	izquierda a derecha
== !=	izquierda a derecha
&	izquierda a derecha
^	izquierda a derecha
	izquierda a derecha
&&	izquierda a derecha
	izquierda a derecha
?:	derecha a izquierda
= += -= *= ...	derecha a izquierda
, (operador coma)	izquierda a derecha