

Informática I
Operadores

Claudio J. Paz

2 de mayo de 2018

Operadores de asignación

Operadores de asignación

Operador	Ejemplo	Equivalente	Explicación
=	$x = 3$		Asigna el valor 3 a x

Operadores de asignación

Operador	Ejemplo	Equivalente	Explicación
=	$x = 3$		Asigna el valor 3 a x
+=	$y += 1$	$y = y + 1$	Incrementa en 1 el valor de y

Operadores de asignación

Operador	Ejemplo	Equivalente	Explicación
=	<code>x = 3</code>		Asigna el valor 3 a <code>x</code>
+=	<code>y += 1</code>	<code>y = y + 1</code>	Incrementa en 1 el valor de <code>y</code>
--	<code>z -= 3</code>	<code>z = z - 3</code>	Decrementa en 3 el valor de <code>z</code>

Operadores de asignación

Operador	Ejemplo	Equivalente	Explicación
=	<code>x = 3</code>		Asigna el valor 3 a <code>x</code>
+=	<code>y += 1</code>	<code>y = y + 1</code>	Incrementa en 1 el valor de <code>y</code>
--	<code>z -= 3</code>	<code>z = z - 3</code>	Decrementa en 3 el valor de <code>z</code>
*=	<code>v *= 2</code>	<code>v = v * 2</code>	Duplica el valor de <code>v</code>

Operadores de asignación

Operador	Ejemplo	Equivalente	Explicación
=	<code>x = 3</code>		Asigna el valor 3 a <code>x</code>
+=	<code>y += 1</code>	<code>y = y + 1</code>	Incrementa en 1 el valor de <code>y</code>
--	<code>z -= 3</code>	<code>z = z - 3</code>	Decrementa en 3 el valor de <code>z</code>
*=	<code>v *= 2</code>	<code>v = v * 2</code>	Duplica el valor de <code>v</code>
/=	<code>w /= 2</code>	<code>w = w / 2</code>	Divide en 2 el valor de <code>w</code>

Operadores de asignación

Operador	Ejemplo	Equivalente	Explicación
=	x = 3		Asigna el valor 3 a x
+=	y += 1	y = y + 1	Incrementa en 1 el valor de y
--	z -= 3	z = z - 3	Decrementa en 3 el valor de z
*=	v *= 2	v = v * 2	Duplica el valor de v
/=	w /= 2	w = w / 2	Divide en 2 el valor de w
%=	n %= 2	n = n % 2	Módulo a 2 de n

Operadores de unarios

Operadores unarios

Operador	Ejemplo	Equivalente	Explicación
++	x++	x += 1	Utiliza e Incrementa en uno el valor

Operadores unarios

Operador	Ejemplo	Equivalente	Explicación
++	x++	x += 1	Utiliza e Incrementa en uno el valor
++	++x	x += 1	Incrementa en uno y Utiliza el valor

Operadores unarios

Operador	Ejemplo	Equivalente	Explicación
++	x++	x += 1	Utiliza e Incrementa en uno el valor
++	++x	x += 1	Incrementa en uno y Utiliza el valor
--	y--	y -= 1	Utiliza y Decrementa en uno el valor

Operadores unarios

Operador	Ejemplo	Equivalente	Explicación
++	x++	x += 1	Utiliza e Incrementa en uno el valor
++	++x	x += 1	Incrementa en uno y Utiliza el valor
--	y--	y -= 1	Utiliza y Decrementa en uno el valor
--	--y	y -= 1	Decrementa en uno y Utiliza el valor

Operadores unarios

```
#include <stdio.h>

int main(void)
{
    int i = 0;

    printf("%d\n", i+1 );

    if ( i == 0 )
        printf("i es cero\n");

    return 0;
}
```

Operadores unarios

```
#include <stdio.h>

int main(void)
{
    int i = 0;

    printf("%d\n", i+1 );

    if ( i == 0 )
        printf("i es cero\n");

    return 0;
}
```

```
1
i es cero
```

Operadores unarios

```
#include <stdio.h>

int main(void)
{
    int i = 0;

    printf("%d\n", i++ );

    if ( i == 0 )
        printf("i es cero\n");

    return 0;
}
```


Operadores unarios

```
#include <stdio.h>

int main(void)
{
    int i = 0;

    printf("%d\n", i++ );

    if ( i == 0 )
        printf("i es cero\n");

    return 0;
}
```

0

Operadores lógicos

Operadores lógicos

Operadores lógicos

Operador	Descripción
----------	-------------

!	Negación
---	----------

Operadores lógicos

Operador	Descripción
----------	-------------

!	Negación
---	----------

&&	AND lógico, ambas condiciones deben cumplirse
----	---

Operadores lógicos

Operador	Descripción
!	Negación
&&	AND lógico, ambas condiciones deben cumplirse
	OR lógico, alguna de las condiciones deben cumplirse

```
#include <stdio.h>

int main (void)
{
    int a, b, c;

    printf( "Ingrese tres números diferentes\n" );
    printf( "Ingrese número a: " ); scanf( "%d", &a );
    printf( "Ingrese número b: " ); scanf( "%d", &b );
    printf( "Ingrese número c: " ); scanf( "%d", &c );

    if ( a > b )
        if ( a > c )
            printf( "El número a es el mayor\n" );
        else
            printf( "El número c es el mayor\n" );
    else
        if ( b > c )
            printf( "El número b es el mayor\n" );
        else
            printf( "El número c es el mayor\n" );

    return 0;
}
```

```
#include<stdio.h>

int main (void)
{
    int a, b, c;

    printf( "Ingrese tres números diferentes\n" );
    printf( "Ingrese número a: " ); scanf( "%d", &a );
    printf( "Ingrese número b: " ); scanf( "%d", &b );
    printf( "Ingrese número c: " ); scanf( "%d", &c );

    if ( a > b && a > c )
        printf( "El número a es el mayor\n" );
    else if ( b > a && b > c )
        printf( "El número b es el mayor\n" );
    else
        printf( "El número c es el mayor\n" );

    return 0;
}
```



```

#include<stdio.h>

int main (void)
{
    int a, b, c;

    printf( "Ingrese tres números diferentes\n" );
    printf( "Ingrese número a: " ); scanf( "%d", &a );
    printf( "Ingrese número b: " ); scanf( "%d", &b );
    printf( "Ingrese número c: " ); scanf( "%d", &c );

    if ( a > b && a > c )
        printf( "El número a es el mayor\n" );
    else if ( b > a && b > c )
        printf( "El número b es el mayor\n" );
    else
        printf( "El número c es el mayor\n" );

    return 0;
}

```

a	b	c	a > b	a > c	1er if	b > a	b > c	2do if	mayor
---	---	---	-------	-------	--------	-------	-------	--------	-------

```

#include<stdio.h>

int main (void)
{
    int a, b, c;

    printf( "Ingrese tres números diferentes\n" );
    printf( "Ingrese número a: " ); scanf( "%d", &a );
    printf( "Ingrese número b: " ); scanf( "%d", &b );
    printf( "Ingrese número c: " ); scanf( "%d", &c );

    if ( a > b && a > c )
        printf( "El número a es el mayor\n" );
    else if ( b > a && b > c )
        printf( "El número b es el mayor\n" );
    else
        printf( "El número c es el mayor\n" );

    return 0;
}

```

a	b	c	a > b	a > c	1er if	b > a	b > c	2do if	mayor
1	2	3	F	F	F	V	F	F	c

```

#include<stdio.h>

int main (void)
{
    int a, b, c;

    printf( "Ingrese tres números diferentes\n" );
    printf( "Ingrese número a: " ); scanf( "%d", &a );
    printf( "Ingrese número b: " ); scanf( "%d", &b );
    printf( "Ingrese número c: " ); scanf( "%d", &c );

    if ( a > b && a > c )
        printf( "El número a es el mayor\n" );
    else if ( b > a && b > c )
        printf( "El número b es el mayor\n" );
    else
        printf( "El número c es el mayor\n" );

    return 0;
}

```

a	b	c	a > b	a > c	1er if	b > a	b > c	2do if	mayor
1	2	3	F	F	F	V	F	F	c
1	3	2	F	F	F	V	V	V	b

```

#include<stdio.h>

int main (void)
{
    int a, b, c;

    printf( "Ingrese tres números diferentes\n" );
    printf( "Ingrese número a: " ); scanf( "%d", &a );
    printf( "Ingrese número b: " ); scanf( "%d", &b );
    printf( "Ingrese número c: " ); scanf( "%d", &c );

    if ( a > b && a > c )
        printf( "El número a es el mayor\n" );
    else if ( b > a && b > c )
        printf( "El número b es el mayor\n" );
    else
        printf( "El número c es el mayor\n" );

    return 0;
}

```

a	b	c	a > b	a > c	1er if	b > a	b > c	2do if	mayor
1	2	3	F	F	F	V	F	F	c
1	3	2	F	F	F	V	V	V	b
3	1	2	V	V	V	-	-	-	a

Operadores lógicos

```
if ( a > b && a > c )
    printf( "El número a es el mayor\n" );
else if ( b > a && b > c )
    printf( "El número b es el mayor\n" );
else
    printf( "El número c es el mayor\n" );
```

Operadores lógicos

```
if ( a > b && a > c )
    printf( "El número a es el mayor\n" );
else if ( b > a && b > c )
    printf( "El número b es el mayor\n" );
else
    printf( "El número c es el mayor\n" );
```

$a > b$	$a > c$	$\&\&$
F	F	

Operadores lógicos

```
if ( a > b && a > c )
    printf( "El número a es el mayor\n" );
else if ( b > a && b > c )
    printf( "El número b es el mayor\n" );
else
    printf( "El número c es el mayor\n" );
```

$a > b$	$a > c$	$\&\&$
F	F	F

Operadores lógicos

```
if ( a > b && a > c )
    printf( "El número a es el mayor\n" );
else if ( b > a && b > c )
    printf( "El número b es el mayor\n" );
else
    printf( "El número c es el mayor\n" );
```

$a > b$	$a > c$	$\&\&$
F	F	F
F	V	

Operadores lógicos

```
if ( a > b && a > c )
    printf( "El número a es el mayor\n" );
else if ( b > a && b > c )
    printf( "El número b es el mayor\n" );
else
    printf( "El número c es el mayor\n" );
```

$a > b$	$a > c$	$\&\&$
F	F	F
F	V	F

Operadores lógicos

```
if ( a > b && a > c )
    printf( "El número a es el mayor\n" );
else if ( b > a && b > c )
    printf( "El número b es el mayor\n" );
else
    printf( "El número c es el mayor\n" );
```

$a > b$	$a > c$	$\&\&$
F	F	F
F	V	F
V	F	

Operadores lógicos

```
if ( a > b && a > c )
    printf( "El número a es el mayor\n" );
else if ( b > a && b > c )
    printf( "El número b es el mayor\n" );
else
    printf( "El número c es el mayor\n" );
```

$a > b$	$a > c$	$\&\&$
F	F	F
F	V	F
V	F	F

Operadores lógicos

```
if ( a > b && a > c )
    printf( "El número a es el mayor\n" );
else if ( b > a && b > c )
    printf( "El número b es el mayor\n" );
else
    printf( "El número c es el mayor\n" );
```

$a > b$	$a > c$	$\&\&$
F	F	F
F	V	F
V	F	F
V	V	

Operadores lógicos

```
if ( a > b && a > c )
    printf( "El número a es el mayor\n" );
else if ( b > a && b > c )
    printf( "El número b es el mayor\n" );
else
    printf( "El número c es el mayor\n" );
```

$a > b$	$a > c$	$\&\&$
F	F	F
F	V	F
V	F	F
V	V	V

Operadores lógicos

Operadores lógicos

```
#include <stdio.h>

int main (void)
{

    int n1 = 2;
    int n2 = 3;

    if ( n1 && n2 )
        printf( "Mensaje\n" );

    return 0;
}
```

Operadores lógicos

```
#include <stdio.h>

int main (void)
{
    int n1 = 2;
    int n2 = 3;

    if ( n1 && n2 )
        printf( "Mensaje\n" );

    return 0;
}
```

Mensaje

Operadores lógicos

Operadores lógicos

expresion1 | expresion2 | expresion1 && expresion2

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0
distinto de 0	0	

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0
distinto de 0	0	0

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0
distinto de 0	0	0
distinto de 0	distinto de 0	

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0
distinto de 0	0	0
distinto de 0	distinto de 0	1

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0
distinto de 0	0	0
distinto de 0	distinto de 0	1

expresion1	expresion2	expresion1 expresion2
------------	------------	--------------------------

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0
distinto de 0	0	0
distinto de 0	distinto de 0	1

expresion1	expresion2	expresion1 expresion2
0	0	

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0
distinto de 0	0	0
distinto de 0	distinto de 0	1

expresion1	expresion2	expresion1 expresion2
0	0	0

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0
distinto de 0	0	0
distinto de 0	distinto de 0	1

expresion1	expresion2	expresion1 expresion2
0	0	0
0	distinto de 0	

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0
distinto de 0	0	0
distinto de 0	distinto de 0	1

expresion1	expresion2	expresion1 expresion2
0	0	0
0	distinto de 0	1

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0
distinto de 0	0	0
distinto de 0	distinto de 0	1

expresion1	expresion2	expresion1 expresion2
0	0	0
0	distinto de 0	1
distinto de 0	0	1

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0
distinto de 0	0	0
distinto de 0	distinto de 0	1

expresion1	expresion2	expresion1 expresion2
0	0	0
0	distinto de 0	1
distinto de 0	0	1

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0
distinto de 0	0	0
distinto de 0	distinto de 0	1

expresion1	expresion2	expresion1 expresion2
0	0	0
0	distinto de 0	1
distinto de 0	0	1
distinto de 0	distinto de 0	1

Operadores lógicos

expresion1	expresion2	expresion1 && expresion2
0	0	0
0	distinto de 0	0
distinto de 0	0	0
distinto de 0	distinto de 0	1

expresion1	expresion2	expresion1 expresion2
0	0	0
0	distinto de 0	1
distinto de 0	0	1
distinto de 0	distinto de 0	1

Precedencia de Operadores

Precedencia de operadores

Precedencia de operadores

Operadores

Asociatividad

Precedencia de operadores

Operadores						Asociatividad
++	--	+	-	!	(tipo)	Derecha a Izquierda

Precedencia de operadores

Operadores						Asociatividad
++	--	+	-	!	(tipo)	Derecha a Izquierda
*	/	%				Izquierda a Derecha

Precedencia de operadores

Operadores						Asociatividad
++	--	+	-	!	(tipo)	Derecha a Izquierda
*	/	%				Izquierda a Derecha
+	-					Izquierda a Derecha

Precedencia de operadores

Operadores						Asociatividad
++	--	+	-	!	(tipo)	Derecha a Izquierda
*	/	%				Izquierda a Derecha
+	-					Izquierda a Derecha
<	<=	>	>=			Izquierda a Derecha

Precedencia de operadores

Operadores	Asociatividad
++ -- + - ! (tipo)	Derecha a Izquierda
* / %	Izquierda a Derecha
+ -	Izquierda a Derecha
< <= > >=	Izquierda a Derecha
== !=	Izquierda a Derecha

Precedencia de operadores

Operadores	Asociatividad
++ -- + - ! (tipo)	Derecha a Izquierda
* / %	Izquierda a Derecha
+ -	Izquierda a Derecha
< <= > >=	Izquierda a Derecha
== !=	Izquierda a Derecha
&&	Izquierda a Derecha

Precedencia de operadores

Operadores	Asociatividad
++ -- + - ! (tipo)	Derecha a Izquierda
* / %	Izquierda a Derecha
+ -	Izquierda a Derecha
< <= > >=	Izquierda a Derecha
== !=	Izquierda a Derecha
&&	Izquierda a Derecha
= += -= *= /= %=	Derecha a Izquierda

Precedencia de operadores

Operadores	Asociatividad
++ -- + - ! (tipo)	Derecha a Izquierda
* / %	Izquierda a Derecha
+ -	Izquierda a Derecha
< <= > >=	Izquierda a Derecha
== !=	Izquierda a Derecha
&&	Izquierda a Derecha
= += -= *= /= %=	Derecha a Izquierda
,	Izquierda a Derecha

Precedencia de operadores

Operadores	Asociatividad
++ -- + - ! (tipo)	Derecha a Izquierda
* / %	Izquierda a Derecha
+ -	Izquierda a Derecha
< <= > >=	Izquierda a Derecha
== !=	Izquierda a Derecha
&&	Izquierda a Derecha
= += -= *= /= %=	Derecha a Izquierda
,	Izquierda a Derecha


```
#include <stdio.h>

int main (void)
{
    int a = 2;
    int b = 3;
    int c = 5;
    int d = 4;
    int res;

    res = a * b + c * d;
    printf( "%d\n", res );

    return 0;
}
```

```
#include <stdio.h>

int main (void)
{
    int a = 2;
    int b = 3;
    int c = 5;
    int d = 4;
    int res;

    res = a * b + c * d;
    printf( "%d\n", res );

    return 0;
}
```

26

```
#include <stdio.h>
/*asignacion.c*/

int main (void)
{
    int a=0, b=0, c=0, d=0, e=0;

    e = a = b = c = d = 3;

    printf ("a=%d, b=%d, c=%d, d=%d, e=%d\n", a, b, c, d, e);

    return 0;
}
```

```
#include <stdio.h>
/*asignacion.c*/

int main (void)
{
    int a=0, b=0, c=0, d=0, e=0;

    e = a = b = c = d = 3;

    printf ("a=%d, b=%d, c=%d, d=%d, e=%d\n", a, b, c, d, e);

    return 0;
}
```

```
a=3, b=3, c=3, d=3, e=3
```

```
#include <stdio.h>
/*asignacion-2.c*/

int main (void)
{
    int a=0, b=0, c=0, d=0, e=0;

    e = a = b = c = d == 3;

    printf ("a=%d, b=%d, c=%d, d=%d, e=%d\n", a, b, c, d, e);

    return 0;
}
```

```
#include<stdio.h>
/*asignacion-2.c*/

int main (void)
{
    int a=0, b=0, c=0, d=0, e=0;

    e = a = b = c = d == 3;

    printf ("a=%d, b=%d, c=%d, d=%d, e=%d\n", a, b, c, d, e);

    return 0;
}
```

```
a=0, b=0, c=0, d=0, e=0
```

```
#include <stdio.h>
/* uno.c */

int main (void)
{
    int i = 0;

    if ( i = 1 )
        printf ("Es uno!!\n");

    if ( i = 0 )
        printf ("Es cero!!\n");

    return 0;
}
```

```
#include <stdio.h>
/* uno.c */

int main (void)
{
    int i = 0;

    if ( i = 1 )
        printf ("Es uno!!\n");

    if ( i = 0 )
        printf ("Es cero!!\n");

    return 0;
}
```

Es uno!!


```
#include <stdio.h>
/* uno.c */

int main (void)
{
    int i = 0;

    if ( i == 1 )
        printf ("Es uno!!\n");

    if ( i == 0 )
        printf ("Es cero!!\n");

    return 0;
}
```

```
#include <stdio.h>
/* uno.c */

int main (void)
{
    int i = 0;

    if ( i == 1 )
        printf ("Es uno!!\n");

    if ( i == 0 )
        printf ("Es cero!!\n");

    return 0;
}
```

Es cero!!

Consultas

claudiojpaz@gmail.com

Horario de Consulta: Miércoles 17:00-19:00hs
Of.5 Ed.Salcedo