



# Fundamental Coding with C

Operators 4: Assignment



# Operators 4: Assignment

## Assignment Operators

- Assignment operators are used to assigning value to a variable.
- The left side operand of the assignment operator is a variable and right side operand of the assignment operator is a value.
- The value on the right side must be of the same data-type of the variable on the left side otherwise will raise an error.
- They are very used in the C language.



# Operators 4: Assignment

## Assignment Operators

It is used to assign a particular value to a variable.

- **=** (Assignment)- Used to assign a value from right side operand to left side operand.
- **+=** (Addition Assignment)- To store the sum of both the operands to the left side operand.
- **-=** (Subtraction Assignment) – To store the difference of both the operands to the left side operand.
- **\*=** (Multiplication Assignment) – To store the product of both the operands to the left side operand.
- **/=** (Division Assignment) – To store the division of both the operands to the left side operand.
- **%=** (Remainder Assignment) – To store the remainder of both the operands to the left side operand.



# Operators 4: Assignment

Assignment Operators table:

Operator	Example	Meaning
<code>+=</code>	<code>A += B</code>	<code>A = A+B</code>
<code>-=</code>	<code>A -= B</code>	<code>A = A-B</code>
<code>*=</code>	<code>A *= B</code>	<code>A = A*B</code>
<code>/=</code>	<code>A /= B</code>	<code>A = A/B</code>
<code>%=</code>	<code>A %= B</code>	<code>A = A%B</code>



# Operators 4: Assignment

Aritmetic operators in C example:

```
int result= 5;  
int a = 10;  
result += a;  
printf("Result = %d \n", result);
```

result = result + a



Result = 15

It's time to try

<https://repl.it/languages/c>





# Operators 4: Assignment

```
#include<stdio.h>
int main()
{
    int number = 10;
    int result= 0;
    result = number;
    printf("result = %d \n", result);
    result += number;
    printf("result = %d \n", result);
    result -= number;
    printf("result = %d \n", result);
    result *= number;
    printf("result = %d \n", result);
    result /= number;
    printf("result = %d \n", result);
    result %= number; /
    printf("result = %d \n", result);
    return 0;
}
```