



Operators

1. Arithmetic Operators

These are used to perform basic mathematical operations.

```
$a = 10;  
$b = 20;  
  
// Addition  
echo $a + $b; // 30  
  
// Subtraction  
echo $a - $b; // -10  
  
// Multiplication  
echo $a * $b; // 200  
  
// Division  
echo $b / $a; // 2  
  
// Modulus (remainder)  
echo $b % $a; // 0  
  
// Exponentiation  
echo $a ** 2; // 100 (10 squared)
```

2. Assignment Operators

These operators assign values to variables.

```

$a = 10;
$a += 20; // $a = $a + 20 (a becomes 30)
echo $a; // 30

$b = 50;
$b -= 5; // $b = $b - 5 (b becomes 45)
echo $b; // 45

$c = 2;
$c **= 3; // $c = $c ** 3 (c becomes 8)
echo $c; // 8

```

3. Comparison Operators

These operators are used to compare two values.

```

var_dump(100 == 100); // true
var_dump(100 == "100"); // true (value comparison)

var_dump(100 === 100); // true (value and type comparison)
var_dump(100 === "100"); // false (different types)

var_dump(100 > 50); // true
var_dump(100 <=> 100); // 0 (equal)
var_dump(100 <=> 50); // 1 (greater)
var_dump(100 <=> 200); // -1 (lesser)

```

4. Increment and Decrement Operators

These operators are used to increase or decrease the value of a variable.

```

$likes = 0;
$likes++; // Increment by 1
$likes++; // Increment by 1

```

```

$likes--;    // Decrement by 1
echo $likes; // 1

$a = 0;
echo $a--;  // 0 (post-decrement)
echo $a;    // -1

$b = 0;
echo --$b;  // -1 (pre-decrement)
echo $b;    // -1

```

5. Logical Operators

These operators are used to compare conditions.

```

var_dump(100 > 50 and 100 > 80);    // true (both are true)
var_dump(100 > 50 && 100 > 100);    // false (last condition is false)

var_dump(100 > 50 or 100 > 110);    // true (first condition is true)
var_dump(100 > 50 xor 100 > 80);    // false (both conditions are true)

```

6. String Operators

These operators are used to concatenate strings.

```

$a = "Elzero";
$b = "Web";
$c = "School";

// Concatenation
echo $a . " " . $b . " " . $c;    // "Elzero Web School"

// Concatenate with .=

```

```
$x = "Elzero ";
$x .= "Web"; // "Elzero Web"
$x .= " School"; // "Elzero Web School"
echo $x; // "Elzero Web School"
```

7. Array Operators

These operators are used to perform operations on arrays.

```
$arr1 = [1 => "A", 2 => "B"];
$arr2 = [3 => "C", 4 => "D"];
$arr3 = $arr1 + $arr2; // Union of arrays
print_r($arr3); // [1 => "A", 2 => "B", 3 => "C", 4 => "D"]

$arr4 = [1 => "10", 2 => "20"];
$arr5 = [2 => 20, 1 => 10];
var_dump($arr4 == $arr5); // true (same key-value pairs)
var_dump($arr4 === $arr5); // false (same keys, different order)
```

8. Error Control Operator

This operator suppresses errors that might occur during code execution.

```
@$a = 10; // Suppress any error (if variable $a is not found)

// File Handling
$f = @file("non_existing_file.txt") or die("File Not Found");
// If file not found, show error message
```

9. Operator Precedence

Operator precedence determines the order of operations.

```
echo 2 + 4 * 5; // 22 (multiplication is done first)
echo (2 + 4) * 5; // 30 (addition inside parentheses is done
first)

echo 10 || false; // 1 (true, because 10 is truthy)
var_dump(10 || false); // true
```