**1. What is an Operator and Relational Operators with Symbols**

**Operator**: An operator is a symbol that tells the compiler or interpreter to perform a specific mathematical, logical, or relational operation and produce a result. Operators work on operands (data values) to perform the desired computation.

**Relational Operators**: Relational operators are used to compare two values. They return either true or false based on the comparison result.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| == | Equal to | $a == $b (true if $a is equal to $b) |
| != | Not equal to | $a != $b (true if $a is not equal to $b) |
| > | Greater than | $a > $b (true if $a is greater than $b) |
| < | Less than | $a < $b (true if $a is less than $b) |
| >= | Greater than or equal to | $a >= $b (true if $a is greater than or equal to $b) |
| <= | Less than or equal to | $a <= $b (true if $a is less than or equal to $b) |

**2. Difference Between Binary and Unary Operators**

* **Binary Operator**: Requires **two operands** to perform an operation (e.g., addition, subtraction). Example: $a + $b (adds two numbers).
* **Unary Operator**: Requires **only one operand** to perform an operation. Example: -$a (negates the value of $a).

**Example:**

* **Binary Operator**: $x + $y (requires two operands).
* **Unary Operator**: - $x (requires one operand).

**3. What is a Preceding Operator?**

The **preceding operator** concept refers to the order in which operators are evaluated in an expression. Some operators have a higher precedence than others, meaning they are executed before operators with lower precedence. For example, multiplication has a higher precedence than addition, so in the expression $a + $b \* $c, the multiplication happens first.

**Example:**

$x = 5 + 3 \* 2; // Here \* (multiplication) is executed before + (addition).

**4. What is BODMAS Operation?**

**BODMAS** stands for **Brackets, Orders (powers and roots), Division, Multiplication, Addition, and Subtraction**. It refers to the order in which mathematical operations are performed to get the correct result.

**Order of Operations**:

1. **B**: Brackets first.
2. **O**: Orders (i.e., powers and roots, such as squaring).
3. **DM**: Division and Multiplication (from left to right).
4. **AS**: Addition and Subtraction (from left to right).

**Example**:

$x = 3 + (2 \* 5); // Parentheses first, then multiplication, then addition.

**5. Difference Between =, ==, and === Operators**

* **= (Assignment Operator)**: This operator is used to **assign a value** to a variable.

**Example**:

$x = 5; // Assigns the value 5 to the variable $x.

* **== (Equality Operator)**: This operator is used to **compare two values** for equality, but it does not check for type. It only checks if the values are the same.

**Example**:

$x = 5;

$y = "5";

var\_dump($x == $y); // true (values are the same, types are not compared).

* **=== (Strict Equality Operator)**: This operator checks for **both value and type**. The comparison will only return true if both the value and the type of the operands are the same.

**Example**:

$x = 5;

$y = "5";

var\_dump($x === $y); // false (values are the same, but types are different).

In summary:

* =: Assigns a value.
* ==: Compares value only (ignores type).
* ===: Compares both value and type.