**Arithmetic Operators:**

| **Operator** | **Example** | **Explanation** |
| --- | --- | --- |
| Addition | **3 + 5** | Adds two numbers together. |
| Subtraction | **7 - 2** | Subtracts one number from another. |
| Multiplication | **4 \* 6** | Multiplies two numbers. |
| Division | **10 / 2** | Divides one number by another. |
| Floor Division | **10 // 3** | Performs integer division. |
| Modulus | **10 % 3** | Returns the remainder of division. |
| Exponentiation | **2 \*\* 3** | Raises one number to the power of another. |

**Assignment Operators:**

| **Operator** | **Example** | **Explanation** |
| --- | --- | --- |
| Assignment | **x = 5** | Assigns the value 5 to variable x. |
| Addition Assignment | **x += 3** | Adds 3 to the current value of x. |
| Subtraction Assignment | **x -= 2** | Subtracts 2 from the current value of x. |
| Multiplication Assignment | **x \*= 4** | Multiplies the current value of x by 4. |
| Division Assignment | **x /= 2** | Divides the current value of x by 2. |
| Floor Division Assignment | **x //= 2** | Performs integer division and assigns the result to x. |
| Modulus Assignment | **x %= 3** | Computes the modulus of x and assigns the result to x. |
| Exponentiation Assignment | **x \*\*= 2** | Raises x to the power of 2 and assigns the result to x. |
| Bitwise AND Assignment | **x &= 3** | Computes the bitwise AND of x and 3, assigns the result to x. |
| Bitwise OR Assignment | `x | = 3` |
| Bitwise XOR Assignment | **x ^= 3** | Computes the bitwise XOR of x and 3, assigns the result to x. |
| Bitwise Left Shift Assignment | **x <<= 2** | Shifts the bits of x to the left by 2 positions and assigns the result to x. |
| Bitwise Right Shift Assignment | **x >>= 2** | Shifts the bits of x to the right by 2 positions and assigns the result to x. |

**Comparison Operators:**

| **Operator** | **Example** | **Explanation** |
| --- | --- | --- |
| Equal to | **3 == 3** | Checks if two values are equal. |
| Not Equal to | **4 != 5** | Checks if two values are not equal. |
| Greater than | **7 > 5** | Checks if one value is greater than another. |
| Less than | **3 < 6** | Checks if one value is less than another. |
| Greater than or Equal to | **6 >= 4** | Checks if one value is greater than or equal to another. |
| Less than or Equal to | **3 <= 3** | Checks if one value is less than or equal to another. |

**Logical Operators:**

| **Operator** | **Example** | **Explanation** |
| --- | --- | --- |
| AND | **True and False** | Returns True if both conditions are true. |
| OR | **True or False** | Returns True if at least one condition is true. |
| NOT | **not True** | Returns True if the condition is false, and vice versa. |

**Bitwise Operators:**

| **Operator** | **Example** | **Explanation** |
| --- | --- | --- |
| Bitwise AND | **5 & 3** | Performs a bitwise AND operation. |
| Bitwise OR | `5 | 3` |
| Bitwise XOR | **5 ^ 3** | Performs a bitwise XOR operation. |
| Bitwise NOT | **~5** | Performs a bitwise NOT operation. |
| Bitwise Left Shift | **5 << 2** | Shifts the bits of a number to the left. |
| Bitwise Right Shift | **5 >> 2** | Shifts the bits of a number to the right. |

**Identity Operators:**

| **Operator** | **Example** | **Explanation** |
| --- | --- | --- |
| is | **x is y** | Returns True if x is the same object as y. |
| is not | **x is not y** | Returns True if x is not the same object as y. |

**Membership Operators:**

| **Operator** | **Example** | **Explanation** |
| --- | --- | --- |
| in | **'a' in 'banana'** | Returns True if a specified value is present in a sequence. |
| not in | **'z' not in 'banana'** | Returns True if a specified value is not present in a sequence. |