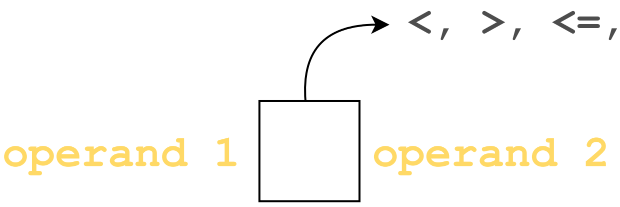
Comparison Operators

This lesson teaches the comparison operators in Rust.

We'll cover the following

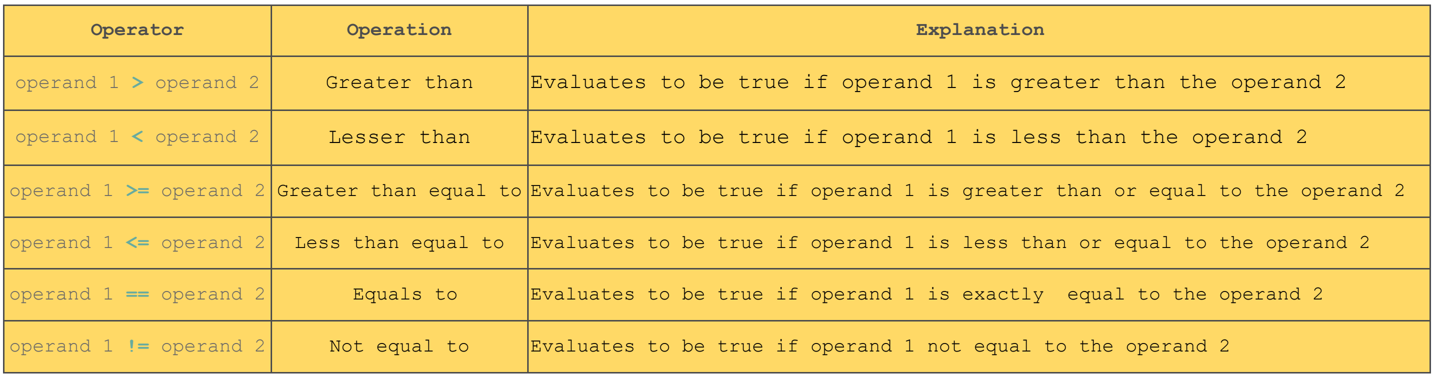
What Are Comparison Operators?

Comparison Operators are used for comparing the values of two operands.



Types

Below is the list of comparison operators in Rust.



Operator	Operation	Explanation
operand 1 > operand 2	Greater than	Evaluates to be true if operand 1 is greater than the operand 2
operand 1 < operand 2	Lesser than	Evaluates to be true if operand 1 is less than the operand 2
operand 1 >= operand 2	Greater than equal to	Evaluates to be true if operand 1 is greater than or equal to the operand 2
operand 1 <= operand 2	Less than equal to	Evaluates to be true if operand 1 is less than or equal to the operand 2
operand 1 == operand 2	Equals to	Evaluates to be true if operand 1 is exactly equal to the operand 2
operand 1 != operand 2	Not equal to	Evaluates to be true if operand 1 not equal to the operand 2

Comparison Operators

The following example shows the use of comparison operators in a program:

```
fn main() {
    let a = 2;
    let b = 3;
    let b = 5;
    println!("Operand 1:{}, Operand 2:{}", a , b);
    println!("a > b:{}", a > b);
    println!("a < b:{}", a < b);
    println!("a >= b:{}", a >= b);
    println!("a <= b:{}", a <= b);
    println!("a == b:{}", a == b);
    println!("a != b:{}", a != b);
}
```

Quiz

Test your understanding of comparison operators in Rust!

Quick Quiz on Comparison Operators!

Q

What is the output of the following code?

```
fn main() {
  let mut a = true;
  let mut b = true;
  a = a > b && b < a;
  b = !b;
  println!("a: {}", a);
  println!("b: {}", b);
}
```

Retake Quiz

In the next lesson, learn about the bitwise operators.