

## Conditional Expressions

A conditional expression is used to make a decision in a program. If the expression is true the code between the curly braces will execute.

### Conditional Operators

Operator	Description	Example
==	Is equal to	<code>if (result == 10) { }</code>
!=	Not equal to	<code>if (result != 10) { }</code>
>	Greater than	<code>if (result &gt; 5) { }</code>
>=	Greater than or equal to	<code>if (result &gt;= 5) { }</code>
<	Less than	<code>if (result &lt; val) { }</code>
<=	Less than or equal to	<code>if (result &lt;= val + 3) { }</code>
!	Not – True for "", 0, null	<code>if (!result) { }</code>

### One Way

Execute code if condition is true

```
if (result == 10)
{
}
```

### Two Way

If condition is true execute code, otherwise execute some other code.

```
if (result < 3)
{
}
else
{
}
```

### Nested

Check many conditions in a row, executing only one that is true with an optional else at the end

```
if (result != 5)
{
}
else if (result < 0)
{
}
else // optional
{
}
```

## Combining Conditional Expression

Conditional expressions can be combined to create more complex logic with the logical AND and OR operators.

### Logical AND

`&&` is used to denote logical AND, and means both expressions have to be true for the corresponding code to execute.

```
if (result > 5 && result !== 10)
{
    // execute if both conditions are true
}
```

### Logical OR

`||` is used to denote logical OR, and means that either expression can be true for the corresponding code to execute.

```
if (result <= 3 || result == 5)
{
    // execute if either condition is true
}
```