QUIZ

Complete this statement such that it evaluates to `true`

```
bool isItRaining = true;
bool isItSunny = false;

( isItRaining ) && (30 > 28);
```

You got it!

Complete this statement such that it evaluates to `false`.

```
true  &&  false
```

You got it!

What will be printed to the console when this code is executed?

```
int x = -100;

if (x == 0)
{
  Console.WriteLine("x is equal to zero.");
}
else if (x > 0)
{
  Console.WriteLine("x is greater than zero.");
}
else
{
  Console.WriteLine("x is less than zero.");
}
```

- `x is greater than zero.`
- `x is equal to zero.`
- Nothing will be printed.
- `x is less than zero.`

Yes! The `else` statement is executed because the other conditions don't evaluate to `true`.

What will happen when this code is executed?

```
int x = 50;

if (x == 0)
{
  Console.WriteLine("x is equal to zero.");
}
else if x > 0
{
  Console.WriteLine("x is greater than zero.");
}
else
{
  Console.WriteLine("x is less than zero.");
}
```

"x is equal to zero." is printed to the console.

There will be an error.

Correct! There must be parentheses around `x > 0`.

Complete this statement such that it evaluates to `true`.

```
int bottles = 3;

bottles [ == ] 3
```

You got it!

Use the ternary conditional operator to complete this statement:

```
num > 6 [ ? ] "big!" [ : ] "small!";
```

You got it!

What will be printed to the console when this code is executed?

```
bool hasAnger = true;
string expr;

if (hasAnger)
{
  expr = "<(`^´)>";
}
else
{
  expr = "¯\_(ツ)_/¯";
}

Console.WriteLine(expr);
```

`<(`^´)>`

👏 Correct! The `if` statement is executed because `hasAnger` is `true`.

What possible values can a `bool` variable take?

`"true"` and `"false"`

`true` and `false`

👏 Correct!

`on` and `off`

`right` and `wrong`

What's wrong with this code?

```
int num = 2;

switch (num)
{
  case 1:
    Console.WriteLine("One for the money");
  case 2:
    Console.WriteLine("Two for the show");
  default:
    Console.WriteLine("Let's go!");
}
```

There are no `break` statements.

Correct! Every case must have a "jump" statement, like `break`.