

Advanced Code Challenge: Always False

There are some situations that you normally want to avoid when programming using conditional statements. One example is a contradiction. This occurs when your condition will always be false no matter what value you pass into it. Let's create an example of a function that contains a contradiction. It will contain a few steps:

1. Define the function to accept a single parameter called num
2. Use a combination of <, > and and to create a contradiction in an if statement.
3. If the condition is true, return True, otherwise return False. The trick here is that because we've written a contradiction, the condition should never be true, so we should expect to always return False.

Create a function named `always_false()` that has one parameter named `num`.

Using an if statement, your variable `num`, and the operators `>`, and `<`, make it so your function will return False no matter what number is stored in `num`.

An if statement that is always false is called a contradiction. You will rarely want to do this while programming, but it is important to realize it is possible to do this.

Output: `Always_False_kaj`

Test Case 1 (using number '0')

```
False
```

Test Case 2 (using number '-1')

```
False
```

Test Case 3 (using number '1')

```
False
```