

## 4.4 - Deep Comparison - Instructions

The `==` operator compares objects by identity. But sometimes you'd prefer to compare the values of their actual properties.

Write a function `deepEqual` that takes two values and returns `true` only if they are the same value or are objects with the same properties, where the values of the properties are equal when compared with a recursive call to `deepEqual`.

To find out whether values should be compared directly (use the `===` operator for that) or have their properties compared, you can use the `typeof` operator. If it produces `"object"` for both values, you should do a deep comparison.

But you have to take one silly exception into account: because of a historical accident, `typeof null` also produces `"object"`.

The `Object.keys` function will be useful when you need to go over the properties of objects to compare them.

Your test for whether you are dealing with a real object will look something like `typeof x == "object" && x != null`. Be careful to compare properties only when *both* arguments are objects. In all other cases you can just immediately return the result of applying `===`.

Use `Object.keys` to go over the properties. You need to test whether both objects have the same set of property names and whether those properties have identical values. One way to do that is to ensure that both objects have the same number of properties (the lengths of the property lists are the same). And then, when looping over one of the object's properties to compare them, always first make sure the other actually has a property by that name. If they have the same number of properties and all properties in one also exist in the other, they have the same set of property names.

Returning the correct value from the function is best done by immediately returning `false` when a mismatch is found and returning `true` at the end of the function.