

This error code indicates a mismatch between the lifetimes appearing in the function signature (i.e., the parameter types and the return type) and the data-flow found in the function body.

Erroneous code example:

```
fn foo<'a>(x: &'a i32, y: &i32) -> &'a i32 { // error: explicit lifetime
                                           //      required in the type of
                                           //      `y`
    if x > y { x } else { y }
}
```

In the code above, the function is returning data borrowed from either `x` or `y`, but the `'a` annotation indicates that it is returning data only from `x`. To fix the error, the signature and the body must be made to match. Typically, this is done by updating the function signature. So, in this case, we change the type of `y` to `&'a i32`, like so:

```
fn foo<'a>(x: &'a i32, y: &'a i32) -> &'a i32 {
    if x > y { x } else { y }
}
```

Now the signature indicates that the function data borrowed from either `x` or `y`. Alternatively, you could change the body to not return data from `y`:

```
fn foo<'a>(x: &'a i32, y: &i32) -> &'a i32 {
    x
}
```