`'_` lifetime name or `&T` without an explicit lifetime name has been used on illegal place.

Erroneous code example:

```
fn underscore_lifetime<'_>(str1: &'_ str, str2: &'_ str) -> &'_ str {
                     //^^ `'_` is a reserved lifetime name
    if str1.len() > str2.len() {
        str1
    } else {
        str2
    }
}

fn and_without_explicit_lifetime<T>()
where
    T: Into<&u32>,
          //^ `&` without an explicit lifetime name
{
}
```

First, `'_` cannot be used as a lifetime identifier in some places because it is a reserved for the anonymous lifetime. Second, `&T` without an explicit lifetime name cannot also be used in some places. To fix them, use a lowercase letter such as `'a`, or a series of lowercase letters such as `'foo`. For more information about lifetime identifier, see the book. For more information on using the anonymous lifetime in Rust 2018, see the Rust 2018 blog post.

Corrected example:

```
fn underscore_lifetime<'a>(str1: &'a str, str2: &'a str) -> &'a str {
    if str1.len() > str2.len() {
        str1
    } else {
        str2
    }
}

fn and_without_explicit_lifetime<'foo, T>()
where
    T: Into<&'foo u32>,
{
}
```