A binary operation was attempted on a type which doesn't support it.

Erroneous code example:

```
let x = 12f32; // error: binary operation `<<` cannot be applied to
               //        type `f32`

x << 2;
```

To fix this error, please check that this type implements this binary operation. Example:

```
let x = 12u32; // the `u32` type does implement it:
               // https://doc.rust-lang.org/stable/std/ops/trait.Shl.html

x << 2; // ok!
```

It is also possible to overload most operators for your own type by implementing traits from `std::ops`.

String concatenation appends the string on the right to the string on the left and may require reallocation. This requires ownership of the string on the left. If something should be added to a string literal, move the literal to the heap by allocating it with `to_owned()` like in `"Your text".to_owned()`.