As warm-up for `2 + 2 ≠ 5` let's prove `0 ≠ 1`. To do this we need to introduce Peano's last axiom `zero_ne_succ n`, a proof that `0 ≠ succ n`. To learn about this result, click on it in the list of lemmas on the right.

Start with `intro h`.

Now change `1` to `succ 0` in `h`.

Now you can `apply zero_ne_succ at h`.

Level

Next

```
False
```

`rw[add_comm 1 0] at h` Retry

Active Goal

Assumptions:

h : `0 = 0 + 1`

Goal:

```
False
```

`rw[<-succ_eq_add_one 0] at h` Retry

Active Goal

Assumptions:

h : `0 = succ 0`

Goal:

```
False
```

`apply zero_ne_succ at h` Retry

Active Goal

Assumptions:

h : `False`

Goal:

```
False
```

`exact h` Retry

level completed! 🎉

zero_ne_succ

`(a : ℕ) : 0 ≠ MyNat.succ a`

`zero_ne_succ n` is the proof that `0 ≠ succ n`.

In Lean, `a ≠ b` is *defined to mean* `a = b → False`. Hence `zero_ne_succ n` is really a proof of `0 = succ n → False`. Here `False` is a generic false statement. This means that you can `apply zero_ne_succ at h` if `h` is a proof of `0 = succ n`.