


```
extern crate timely;
```

```
use timely::dataflow::operators::*;
```

```
fn main() {
```

```
    timely::example(|scope| {
```

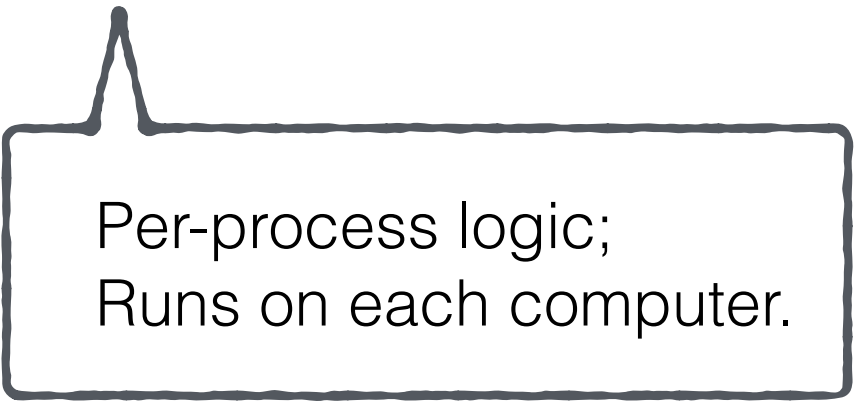
```
        (0..10).to_stream(scope)
```

```
            .exchange(|&x| x)
```

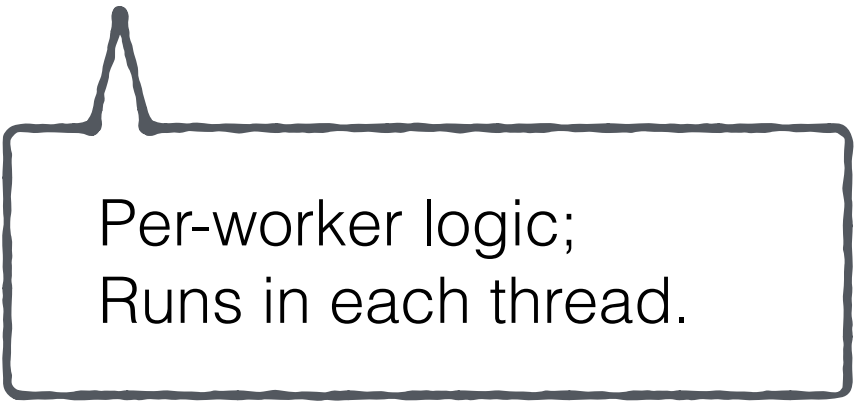
```
            .inspect(|x| println!("{}", x));
```

```
    });
```

```
}
```

A hand-drawn rectangular box with a thick, dark grey border. A small triangular pointer extends from the top-left corner of the box. Inside the box, the text "Per-process logic;" and "Runs on each computer." is written in a black, sans-serif font.

Per-process logic;
Runs on each computer.

A hand-drawn rectangular box with a dark gray border. A small triangle points upwards from the top-left corner of the box. Inside the box, the text "Per-worker logic;" and "Runs in each thread." is written in a black, sans-serif font.

Per-worker logic;
Runs in each thread.

A hand-drawn speech bubble with a dark gray, irregular border. The bubble has a small tail pointing upwards and to the left. Inside the bubble, the text "Your SPMD program. You write this once." is written in a black, sans-serif font, centered horizontally and vertically.

Your SPMD program.
You write this once.


```
extern crate timely;

use timely::dataflow::operators::*;

fn main() {

    timely::example(|scope| {

        (0..10).to_stream(scope)
            .exchange(|&x| x)
            .inspect(|x| println!("{}", x));

    });

}
```

```
extern crate timely;

use timely::dataflow::operators::*;

fn main() {

    timely::example(|scope| {

        (0..10).to_stream(scope)
            .exchange(|&x| x)
            .inspect(|x| println!("{}", x));

    });

}
```

Per-worker logic;
Runs in each thread.