```
extern crate timely;

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

fn main() {

    timely::example(|scope| {

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

    });

}
```

Rust iterator

```
extern crate timely;

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

fn main() {

    timely::example(|scope| {

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

    });

}
```

}

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

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

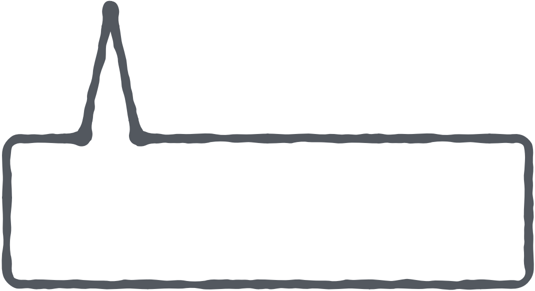
```
timely::example(|scope| {
});
```

```
extern crate timely;
```

```
fn main() {
```

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

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



Rust iterator

distributed stream