







Data movement requirements.  
Could have been **Pipeline**.



Tasteful name



Closure describing operator  
behavior when scheduled.

```
// convenient!  
stream.map(logic);
```

```
// equivalent to the above.  
stream.unary_stream(Pipeline, "Map", |input, output| {  
    while let Some((time, data)) = input.next() {  
        let session = output.session(&time);  
        for datum in data.drain(..) {  
            session.give(logic(datum));  
        }  
    }  
});
```



```
// convenient!  
stream.map(logic);
```

```
// equivalent to the above.  
stream.unary_stream(Pipeline, "Map", |input, output| {  
    while let Some((time, data)) = input.next() {  
        let session = output.session(&time);  
        for datum in data.drain(..) {  
            session.give(logic(datum));  
        }  
    }  
});
```



```
// convenient!  
stream.map(logic);
```

Data movement requirements.  
Could have been **Pipeline**.

Tasteful name

```
// equivalent to the above.  
stream.unary_stream(Pipeline, "Map", |input, output| {
```

```
    while let Some((time, data)) = input.next() {  
        let session = output.session(&time);  
        for datum in data.drain(..) {  
            session.give(logic(datum));  
        }  
    }  
}
```

```
});
```

Closure describing operator  
behavior when scheduled.