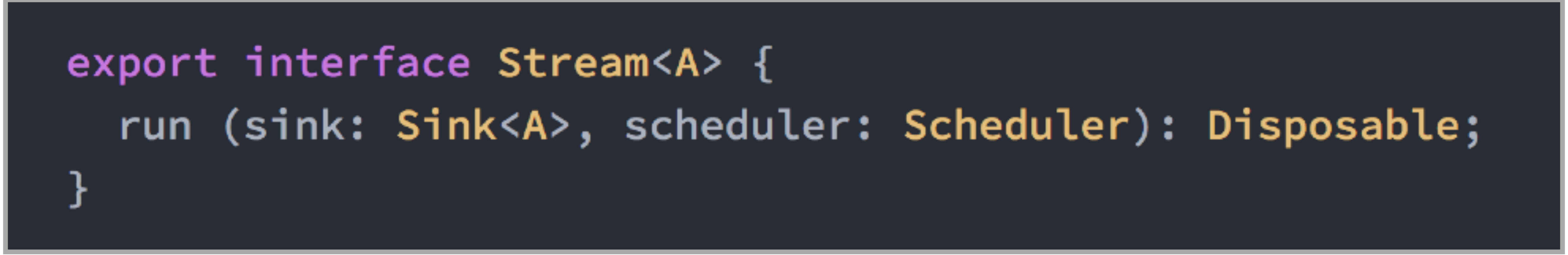
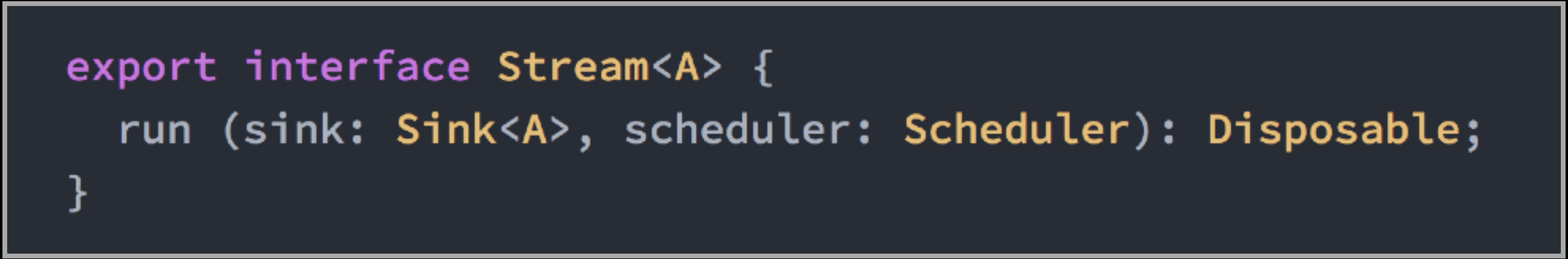


```
export interface Stream<A> {
  run (sink: Sink<A>, scheduler: Scheduler): Disposable;
}
```

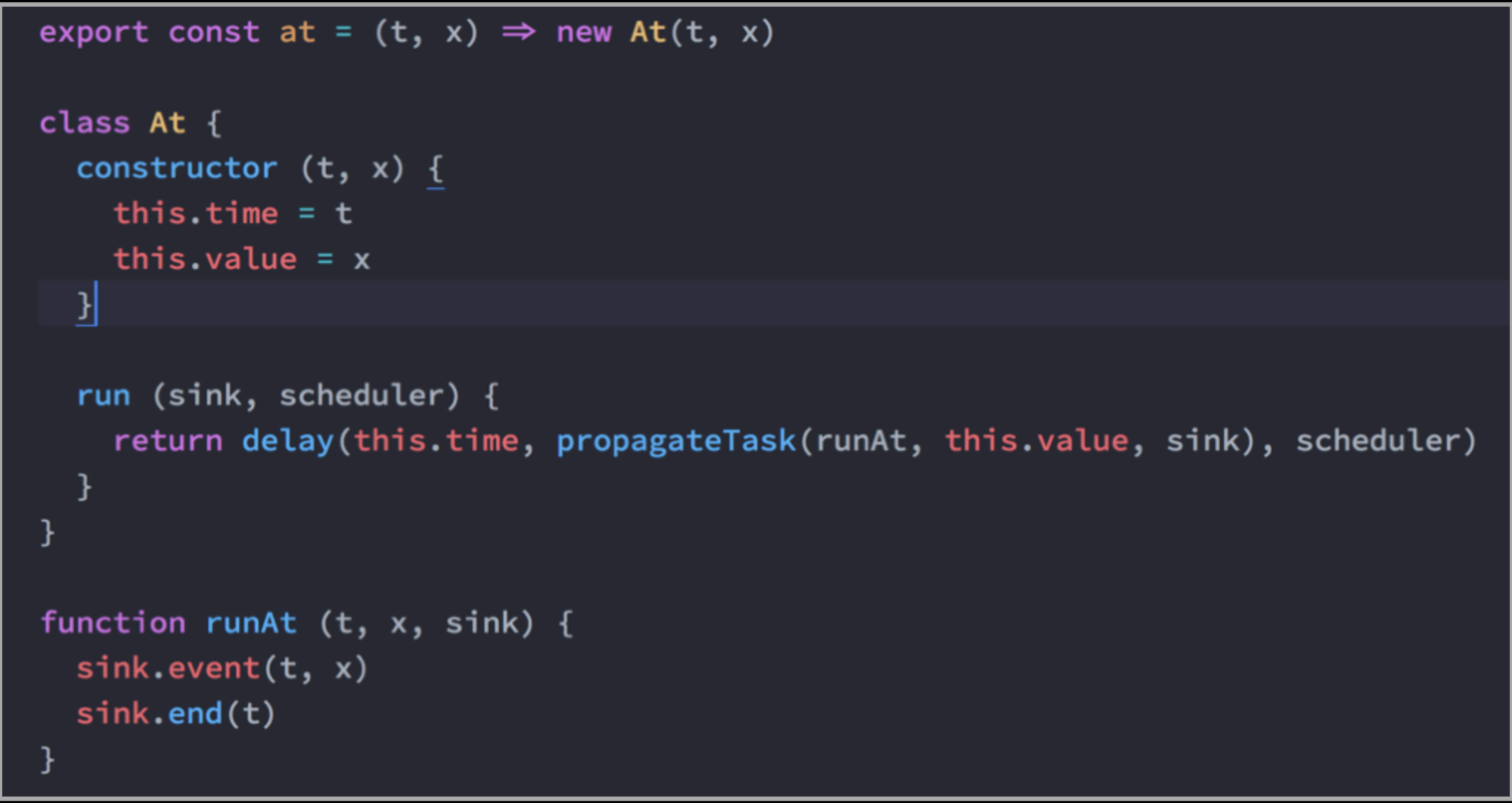
Async guarantee



```
export interface Stream<A> {
  run (sink: Sink<A>, scheduler: Scheduler): Disposable;
}
```

run() will never produce an event before it returns

Scheduler



```
export const at = (t, x) => new At(t, x)

class At {
  constructor (t, x) {
    this.time = t
    this.value = x
  }

  run (sink, scheduler) {
    return delay(this.time, propagateTask(runAt, this.value, sink), scheduler)
  }
}

function runAt (t, x, sink) {
  sink.event(t, x)
  sink.end(t)
}
```