Thinking Reactive with RxJS
Excercise

Starting Point

Here is our starting point:

```
const Observable = Rx.Observable;

const startButton = document.querySelector('#start');
const stopButton = document.querySelector('#stop');
const resetButton = document.querySelector('#reset');

const start$ = Observable.fromEvent(startButton, 'click');
const interval$ = Observable.interval(1000);
const stop$ = Observable.fromEvent(stopButton, 'click');
const reset$ = Observable.fromEvent(resetButton, 'click');


const data = {count:0};
const inc = (acc)=> ({count: acc.count + 1});
const reset = (acc)=> data;

const intervalThatStops$ = interval$
    .takeUntil(stop$);

const incOrReset$ = Observable.merge(
    intervalThatStops$.mapTo(inc),
    reset$.mapTo(reset)
);

start$
    .switchMapTo(incOrReset$)
    .startWith(data)
    .scan((acc, curr)=> curr(acc))
    .subscribe((x)=> console.log(x));
```

Your Challenge – Faster Timers

- Add buttons: Half and Quarter
- When they are clicked timer should start at matching speed (Second, Half a second or Quarter of a second)
- Hints
    - Use Observable.merge to start the 3 starter streams
    - Use mapTo to map those clicks to a milli second value
    - Change from switchMapTo to switchMap so you get the time from the stream
    - We cannot configure the streams as its right now
    - Inline the streams:
        - incOrReset$
        - intervalThatStops$
        - interval$
    - now change 1000 to time
    - You should have a working solution.