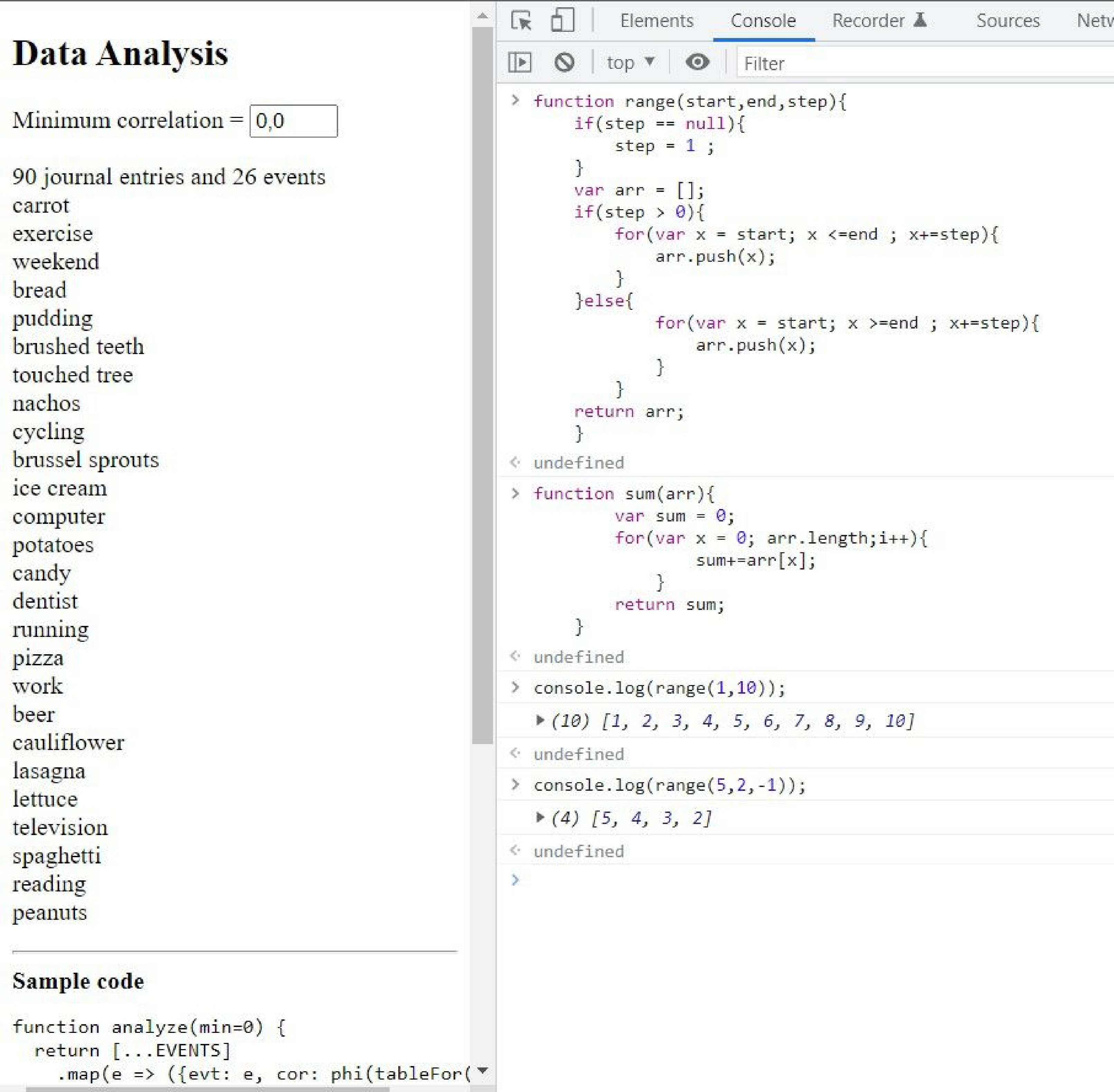


Data Analysis

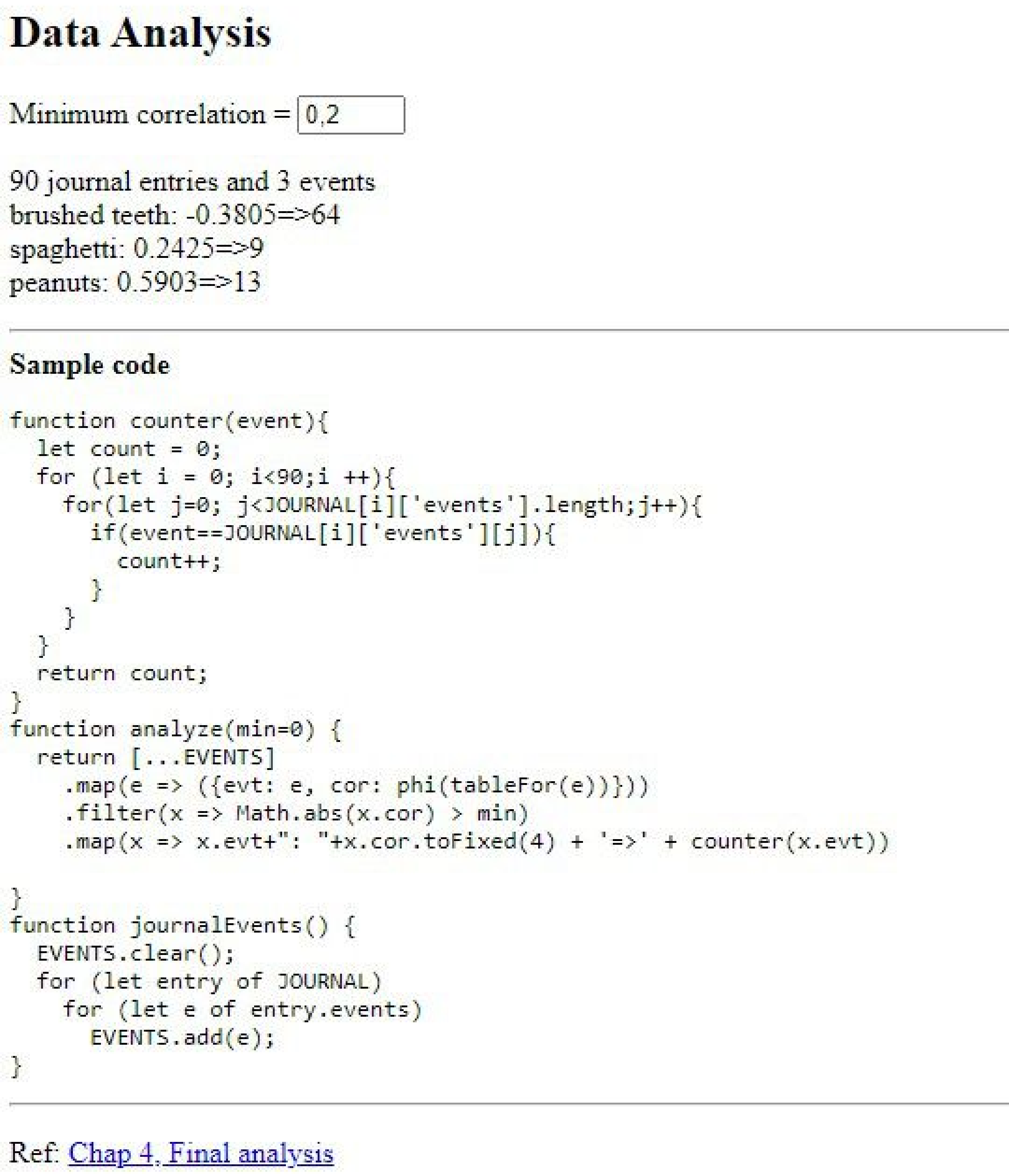
Minimum correlation = 0,0

90 journal entries and 26 events
carrot
exercise
weekend
bread
pudding
brushed teeth
touched tree
nachos
cycling
brussel sprouts
ice cream
computer
potatoes
candy
dentist
running
pizza
work
beer
cauliflower
lasagna
lettuce
television
spaghetti
reading
peanuts

Sample code

```
function analyze(min=0) {
  return [...EVENTS]
    .map(e => ({evt: e, cor: phi(tableFor(
```

```
> function range(start,end,step){
    if(step == null){
        step = 1 ;
    }
    var arr = [];
    if(step > 0){
        for(var x = start; x <=end ; x+=step){
            arr.push(x);
        }
    }else{
            for(var x = start; x >=end ; x+=step){
                arr.push(x);
            }
        }
    return arr;
    }
<· undefined
> function sum(arr){
        var sum = 0;
        for(var x = 0; arr.length;i++){
                sum+=arr[x];
            }
        return sum;
    }
<· undefined
> console.log(range(1,10));
  ▶ (10) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
<· undefined
> console.log(range(5,2,-1));
  ▶ (4) [5, 4, 3, 2]
<· undefined
>
```



Data Analysis

Minimum correlation = 0,2

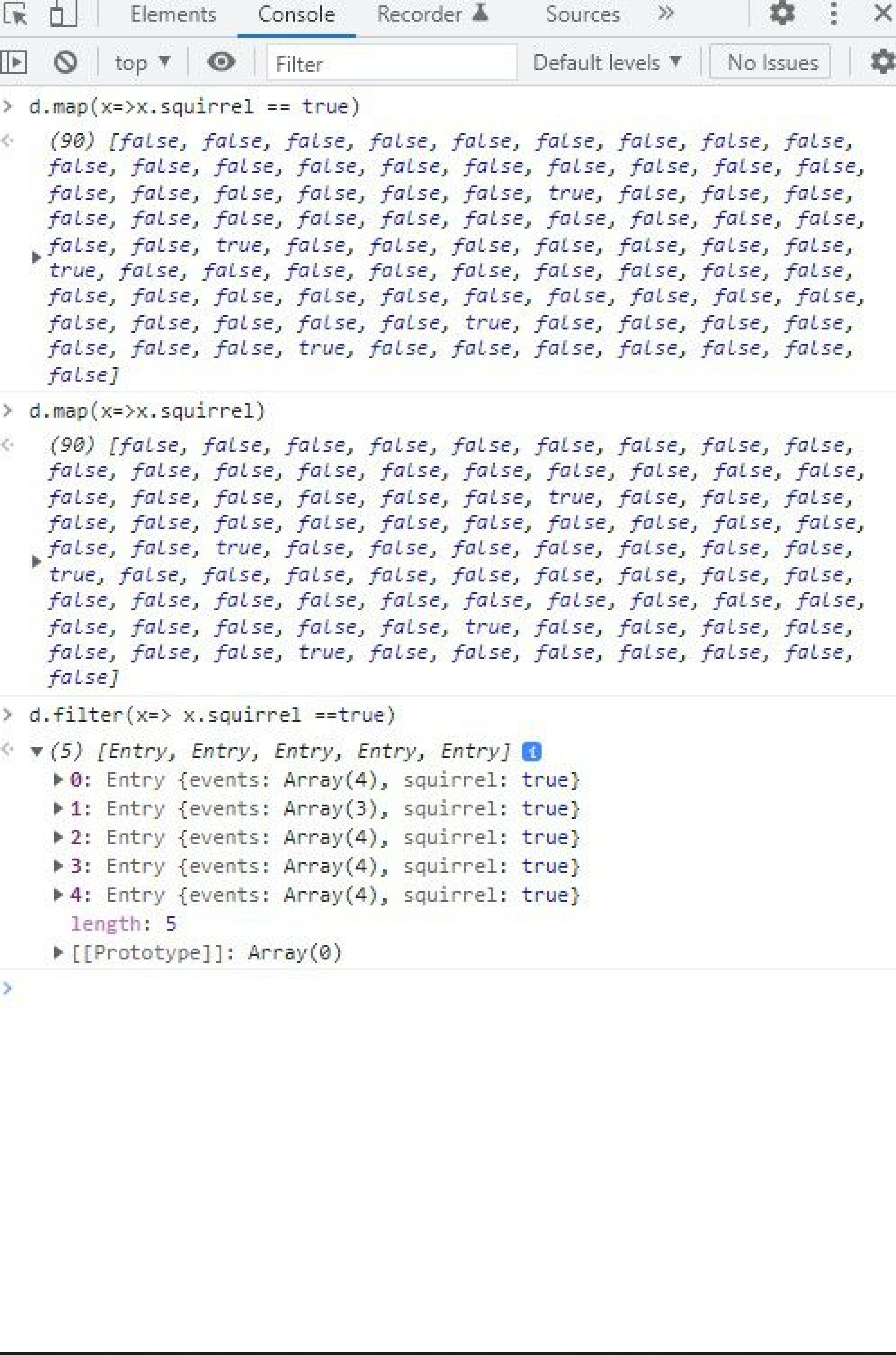
90 journal entries and 3 events
brushed teeth: -0.3805=>64
spaghetti: 0.2425=>9
peanuts: 0.5903=>13

Sample code

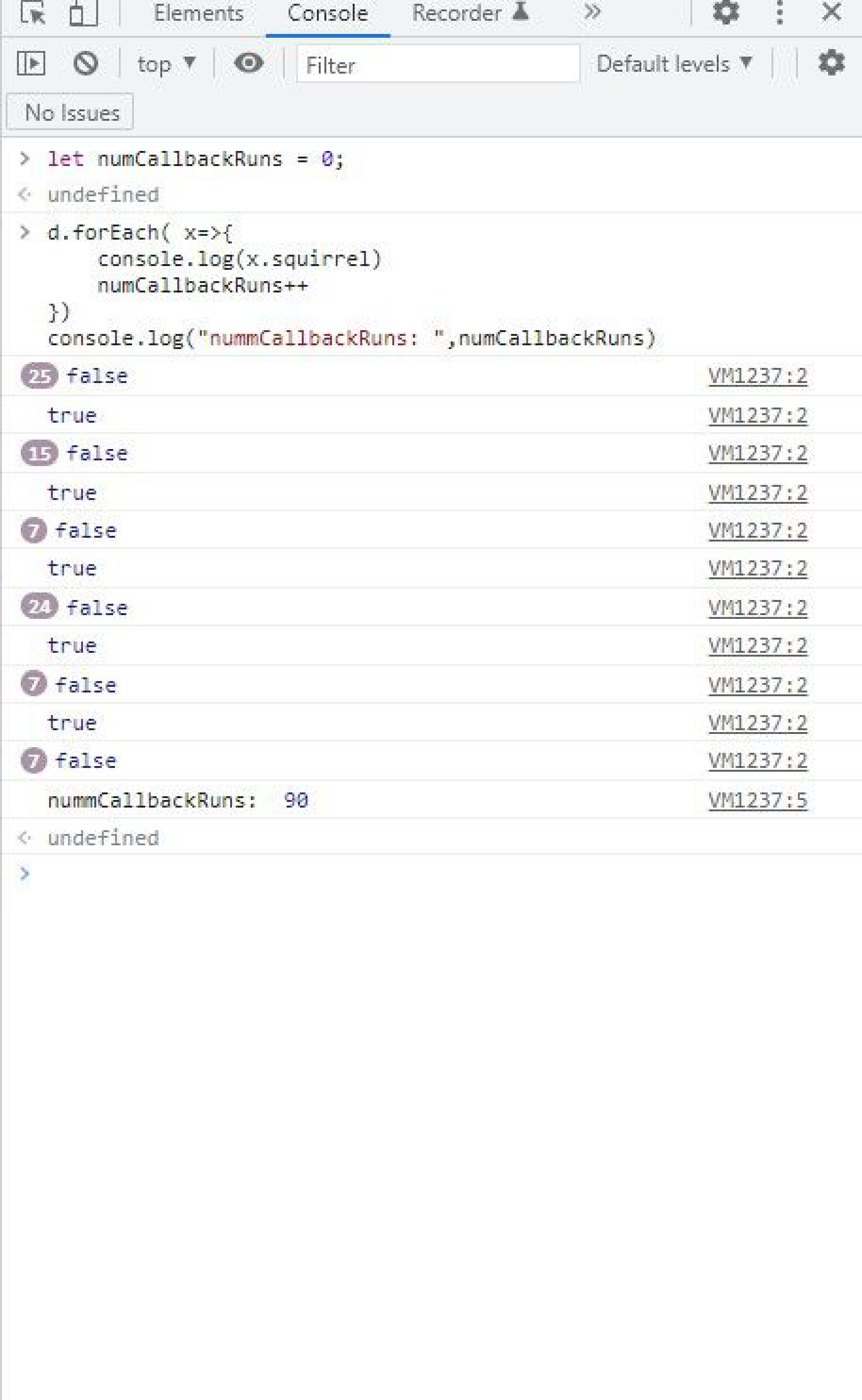
```
function counter(event){
  let count = 0;
  for (let i = 0; i<90;i ++){
    for(let j=0; j<JOURNAL[i]['events'].length;j++){
      if(event==JOURNAL[i]['events'][j]){
        count++;
      }
    }
  }
  return count;
}
function analyze(min=0) {
  return [...EVENTS]
    .map(e => ({evt: e, cor: phi(tableFor(e))}))
    .filter(x => Math.abs(x.cor) > min)
    .map(x => x.evt+": "+x.cor.toFixed(4) + '=>' + counter(x.evt))

}
function journalEvents() {
  EVENTS.clear();
  for (let entry of JOURNAL)
    for (let e of entry.events)
      EVENTS.add(e);
}
```

Ref: Chap 4, Final analysis



```
> d.map(x=>x.squirrel == true)
<   (90) [false, false, false, false, false, false, false, false, false,
    false, false, false, false, false, false, false, false, false, false,
    false, false, false, false, false, false, true, false, false, false,
    false, false, false, false, false, false, false, false, false, false,
  ▶ false, false, true, false, false, false, false, false, false, false,
    true, false, false, false, false, false, false, false, false, false,
    false, false, false, false, false, false, false, false, false, false,
    false, false, false, false, false, true, false, false, false, false,
    false, false, false, true, false, false, false, false, false, false,
    false]
> d.map(x=>x.squirrel)
<   (90) [false, false, false, false, false, false, false, false, false,
    false, false, false, false, false, false, false, false, false, false,
    false, false, false, false, false, false, true, false, false, false,
    false, false, false, false, false, false, false, false, false, false,
  ▶ false, false, true, false, false, false, false, false, false, false,
    true, false, false, false, false, false, false, false, false, false,
    false, false, false, false, false, false, false, false, false, false,
    false, false, false, false, false, true, false, false, false, false,
    false, false, false, true, false, false, false, false, false, false,
    false]
> d.filter(x=> x.squirrel ==true)
< ▼ (5) [Entry, Entry, Entry, Entry, Entry] i
    ▶ 0: Entry {events: Array(4), squirrel: true}
    ▶ 1: Entry {events: Array(3), squirrel: true}
    ▶ 2: Entry {events: Array(4), squirrel: true}
    ▶ 3: Entry {events: Array(4), squirrel: true}
    ▶ 4: Entry {events: Array(4), squirrel: true}
      length: 5
    ▶ [[Prototype]]: Array(0)
>
```



```
> let numCallbackRuns = 0;
< undefined
> d.forEach( x=>{
      console.log(x.squirrel)
      numCallbackRuns++
  })
  console.log("nummCallbackRuns: ",numCallbackRuns)
25 false                                          VM1237:2
  true                                            VM1237:2
15 false                                          VM1237:2
  true                                            VM1237:2
7 false                                           VM1237:2
  true                                            VM1237:2
24 false                                          VM1237:2
  true                                            VM1237:2
7 false                                           VM1237:2
  true                                            VM1237:2
7 false                                           VM1237:2
  nummCallbackRuns:  90                           VM1237:5
< undefined
>
```