The interesting thing from today's video was learn about event loop specially when to use setImmediate instead of setTimeout. I'm familiar with Node.js but I'm not sure that I used the method or not. The second problem was not easy to implement without seeing solution. I was trying to implement returning object literal but this way maybe understandable.

Homework 1.

```
Array.prototype.even = function(){
  return this.filter(el=>el % 2 == 0)
}

Array.prototype.odd = function(){
  return this.filter(el=>el % 2 != 0)
}

let a = [1,2,3,4,5,6,7,8]
console.log(a.even());
console.log(a.odd());
```

Homework 2.

```
function slow(callback){
  if (Math.random() > 0.5) {
    return callback("Error", null);
  }
  return callback(null, {id:12345})
}


function exec(fn){
  let obj = {};
  fn(function(error, data){
    obj.done = function(callback){
      if(error === null){
        callback(data);
      }
      return this;
    }
    obj.fail = function(callback){
      if(error !== null){
        callback(error);
      }
      return this;
    }
  })
  return obj;
}

exec(slow).done(function(data){ console.log(data); })
          .fail(function(err){ console.log("Error: " + err); });
```

Homework 3

1. **Explain why do we want sometimes to use setImmediate instead of using setTimeout?**
   setImmediate callbacks are called after I/O queue is empty. We use setImmediate if we want to queue the function behind whatever I/O event callbacks that are already in the event queue.

2. **Explain the difference between process.nextTick and setImmediate?**
   - process.nextTick run before any other I/O event is fired and it's going to be executed on the current iteration of the event loop, after current operation ends, which means always execute before setTimeout and setImmediate.
   setImmediate executes the callback after I/O eventhandler queue is empty.

3. **Name 10 global modules/methods available in Node environment**
   module
   global
   process
   buffer
   require
   setInterval
   setTimeout
   setImmediate
   clearInterval
   clearTimeout