

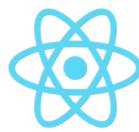
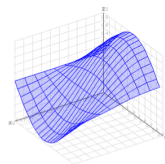
## JavaScript 101

# JavaScript 101

- 无处不在的JavaScript
- 函数式编程
- 常用JavaScript库
- Homework - bookmarks

## JavaScript is everywhere

# JAVASCRIPT IS EVERYWHERE



express

X

## 编译器

```
1 book at 12.49
1 "music CD" at 14.99
1 "chocolate bar" at 0.85
1 bottle of perfume at 18.99
1 packet of "headache pills" at 9.75
1 imported bottle of perfume at 27.99
1 box of imported chocolates at 11.25
```

```
expressions
: statements EOF
```

```

;

statements:
  statement
  | statements statement
;

statement: simple
  | with_quantity
  | with_imported
  | with_imported2
;

simple: COUNT words AT PRICE

with_quantity: COUNT words OF words AT PRICE

with_imported: COUNT IMPORTED words OF words AT PRICE

with_imported2: COUNT words OF IMPORTED words AT PRICE

words:
  STRING
  | WORD
;

```

<http://icodeit.org/2015/09/write-a-parser/>  
<http://icodeit.org/2015/10/mapfile-parser/>

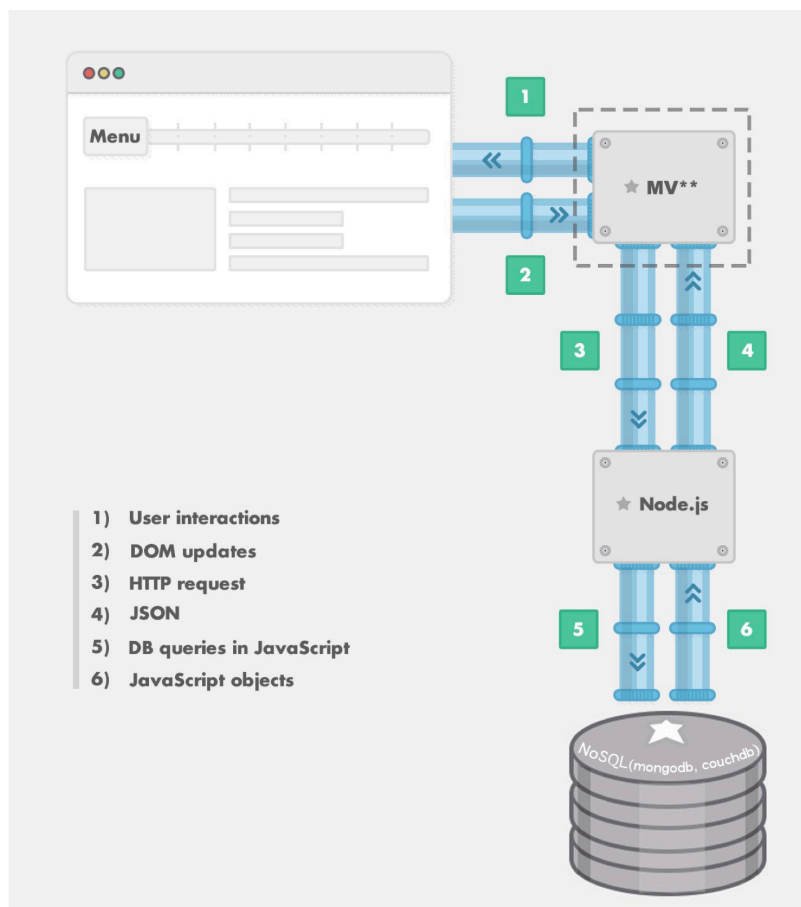
图形

d3.js绘制Charts



后台

node.js, express



x

## 函数式编程

### 过滤Filter

- 给定一个由数字组成的数组，请编写代码，返回这个数组中所有小于7的数字
- 给定一个由数字组成的数组，请编写代码，返回这个数组中所有小于10的数字

```
var numbers = [1, 2, 3, 5, 8, 13, 21, 34];  
var names = ['Juntao Qiu', 'Wenjun Lin', 'Xiaofeng Shi', 'Qianyun Hu', 'Ruili Ma', 'Yanke Tie'];
```

```
function lessThan7(array) {  
  var result = [];  
  for(var i = 0; i < array.length; i++) {  
    if(array[i] < 7) {  
      result.push(array[i]);  
    }  
  }  
  return result;  
}
```

```
function lessThan10(array) {
```

```

    var result = [];
    for(var i = 0; i < array.length; i++) {
        if(array[i] < 10) {
            result.push(array[i]);
        }
    }

    return result;
}

```

作为一个程序员，你很快就可以找出共性，并将代码实现为：

```

function lt(array, base) {
    var result = [];
    for(var i = 0; i < array.length; i++) {
        if(array[i] < base) {
            result.push(array[i]);
        }
    }

    return result;
}

lt(numbers, 7);
lt(numbers, 10);

```

- 给定一个由数字组成的数组，请编写代码，返回这个数组中所有大于**10**的数字

```

function greatThan10(array) {
    var result = [];
    for(var i = 0; i < array.length; i++) {
        if(array[i] > 10) {
            result.push(array[i]);
        }
    }

    return result;
}

function filter_number(array, comparator, base) {
    var result = [];
    for(var i = 0; i < array.length; i++) {
        if(comparator(array[i], base)) {
            result.push(array[i]);
        }
    }

    return result;
}

filter_number(numbers, '<', 7)
filter_number(numbers, '<', 10)
filter_number(numbers, '>', 10)

```

- 给定一个由字符串组成的数组，请编写代码，返回这个数组中，字符串长度等于**10**的字符串

```

filter_number(names, ...);

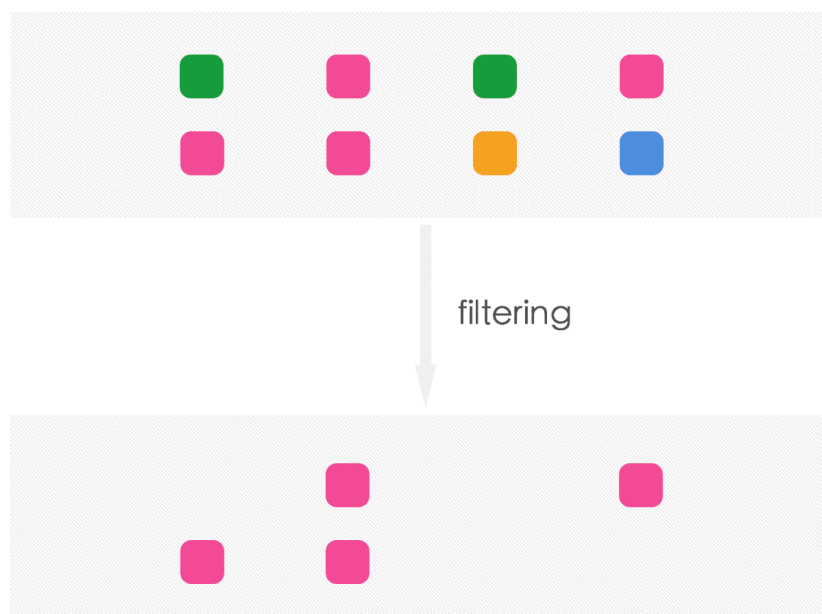
```

```
function equal10(array) {  
  var result = [];  
  for(var i = 0; i < array.length; i++) {  
    if(array[i].length === 10) {  
      result.push(array[i]);  
    }  
  }  
  return result;  
}
```

```
var result = array.filter(function(item) {  
  return item > 3;  
});
```

如果使用ES6的语法，更可以精简为:

```
array.filter(x => x > 3)
```



x

## 映射 Map

```
var names = ['Juntao Qiu', 'Wenjun Lin', 'Xiaofeng Shi', 'Qianyun Hu', 'Ruilu Ma', 'Yanke Tie'];
```

1. 修改所有的名字为小写字母，所有空格需要替换为连字符-

```
function convert(array) {
  var result = [];
  for(var i = 0; i < array.length; i++) {
    result.push(array[i].toLowerCase().replace(/\s+/g, '-'));
  }
  return result;
}
```

```
var numbers = [1, 2, 3, 5, 8, 13, 21, 34];
```

1. 将所有的数字扩大两倍

```
function convert(array) {
  var result = [];
  for(var i = 0; i < array.length; i++) {
    result.push(array[i]*2);
  }
  return result;
}
```

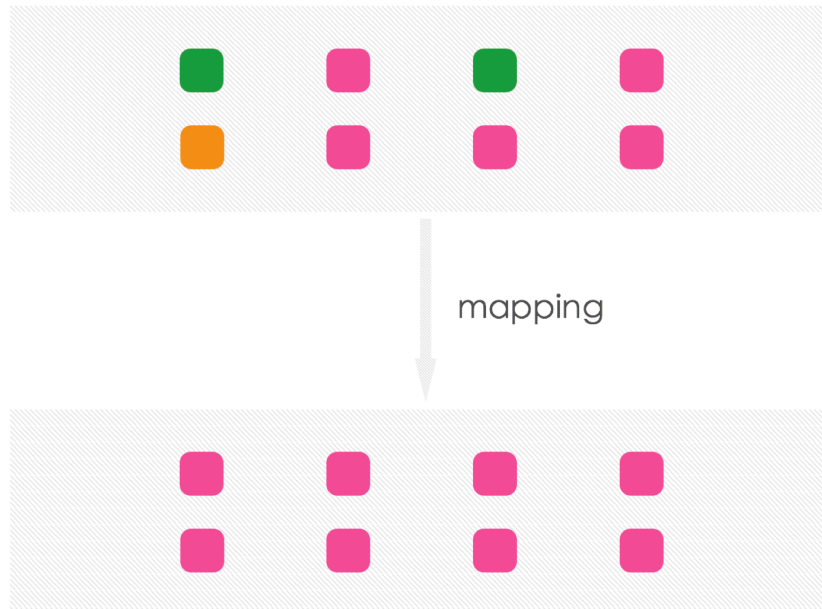
使用JavaScript原生的方式，可以简化为：

```
names.map(function(item) {
  return item.toLowerCase().replace(/\s+/g, '-');
});
```

```
numbers.map(function(item) {
  return item * 2;
});
```

使用ES6的话，可以简化为：

```
names.map(x => x.toLowerCase().replace(/\s+/g, '-'));
numbers.map(x => x * 2);
```



x

## 折叠/规约 **Reduce**

```
var numbers = [1, 2, 3, 5, 8, 13, 21, 34];
```

1. 计算一组数字的总和
2. 计算一组数字的总积

```
function sum(array) {  
  var sum = 0;  
  for(var i = 0; i < array.length; i++) {  
    sum += array[i];  
  }  
  return sum;  
}
```

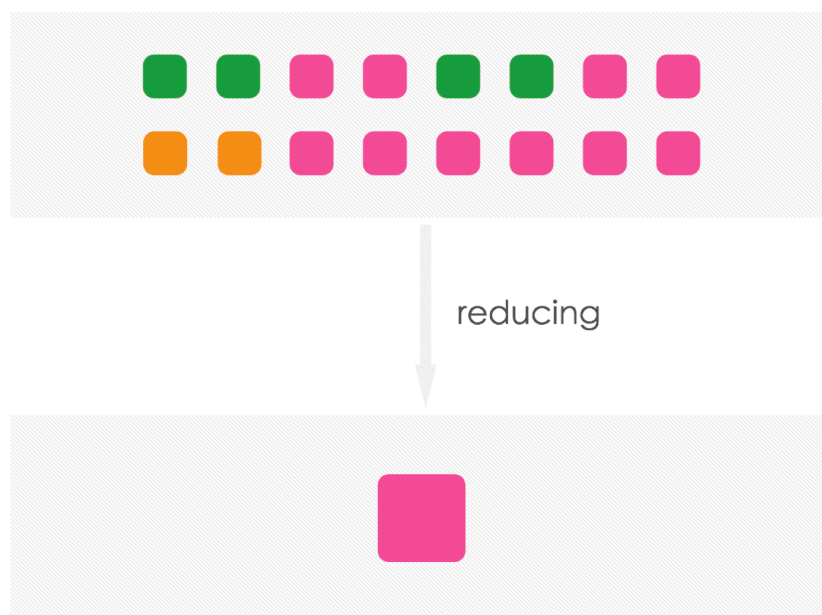
```
function product(array) {  
  var sum = 1;  
  for(var i = 0; i < array.length; i++) {  
    sum *= array[i];  
  }  
  return sum;  
}
```

```
numbers.reduce(function(p, c) {  
  p += c;  
  return p;  
}, 0);
```

```
numbers.reduce(function(p, c) {  
  p *= c;  
  return p;  
}, 1);
```

ES6可以简化为:

```
numbers.reduce((p, c) => p += c, 0);  
numbers.reduce((p, c) => p *= c, 1);
```



x

组织在一起

```
var names = ['a', 'b', 'juntao', 'c', 'wenjun', 'yoyo'];  
expected('JUNTAO, WENJUN, YOYO')
```



1. 给定一个名字组成的数组，期望结果是一个字符串，字符串的组成为每个名字由逗号分割。

```
names
  .filter(x => x.length > 1)
  .map(x => x.toUpperCase())
  .reduce((a, b) => a + ',' + b);
```

```
$.get('/resource.json').done(x => console.log(x)).fail(e => console.log(e))
```

## JavaScript库

### jQuery

- DOM操作
- 网络操作
- 事件绑定

```
for(var i = 0; i < 3; i++) {
  var item = $('<li></li>').text('item-'+i);
  $('#nav').append(item);
}
```

```
[1,2,3].each(x => {
  $('#nav').append($('<li></li>').text('item-'+x))
})
```

```
$('#button').on('click', function(e) {
  console.log('button is clicked');
});
```

```
$.get('/resource.json').done(x => console.log(x)).fail(e => console.log(e))
```

### underscore.js/lodash

```
_( [1,2,3,4,5,6,5,4,3,2,1] )
  .uniq()
  .map(x => x*2)
  .map(x => 'item-'+x);
```

<http://icodeit.org/2015/02/functional-programming-in-underscore-dot-js/>  
<http://icodeit.org/2015/02/collection-operations-in-underscore-dot-js/>  
<http://icodeit.org/2015/02/build-sample-application-by-using-underscore-and-jquery/>

## Homework

实现一个书签管理应用

1. 请求服务器上的bookmarks.json，并显式书签列表
2. 搜索功能，根据用户的输入，过滤已经渲染出来的书签（高亮选中的内容）
3. 符合mockup的样式

可能用到的技术：

1. jQuery（DOM操作，请求远程资源）
2. lodash
3. JavaScript中的正则表达式

