Single Page Apps

Rich clients, modern websites and AngularJS

First things first

A *lot* of learning

Hand-holding was forbidden

You will need to learn and research on your own

Stack Overflow is your friend

Web 101

Thick / Thin Client Cycle

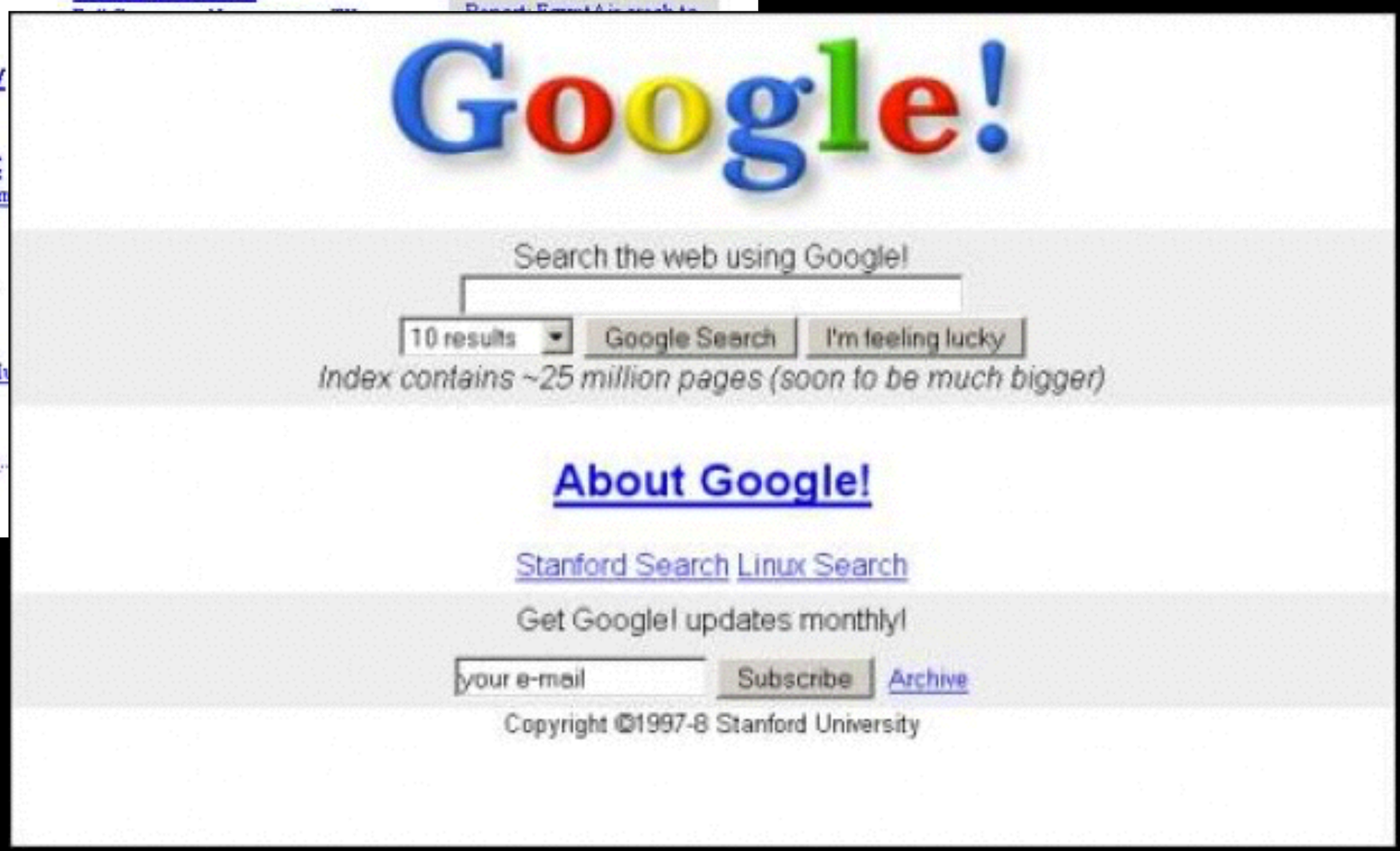

114 Files, 5943 Blocks
14433 Free blocks
.@
digital VT100

This product is licensed to:

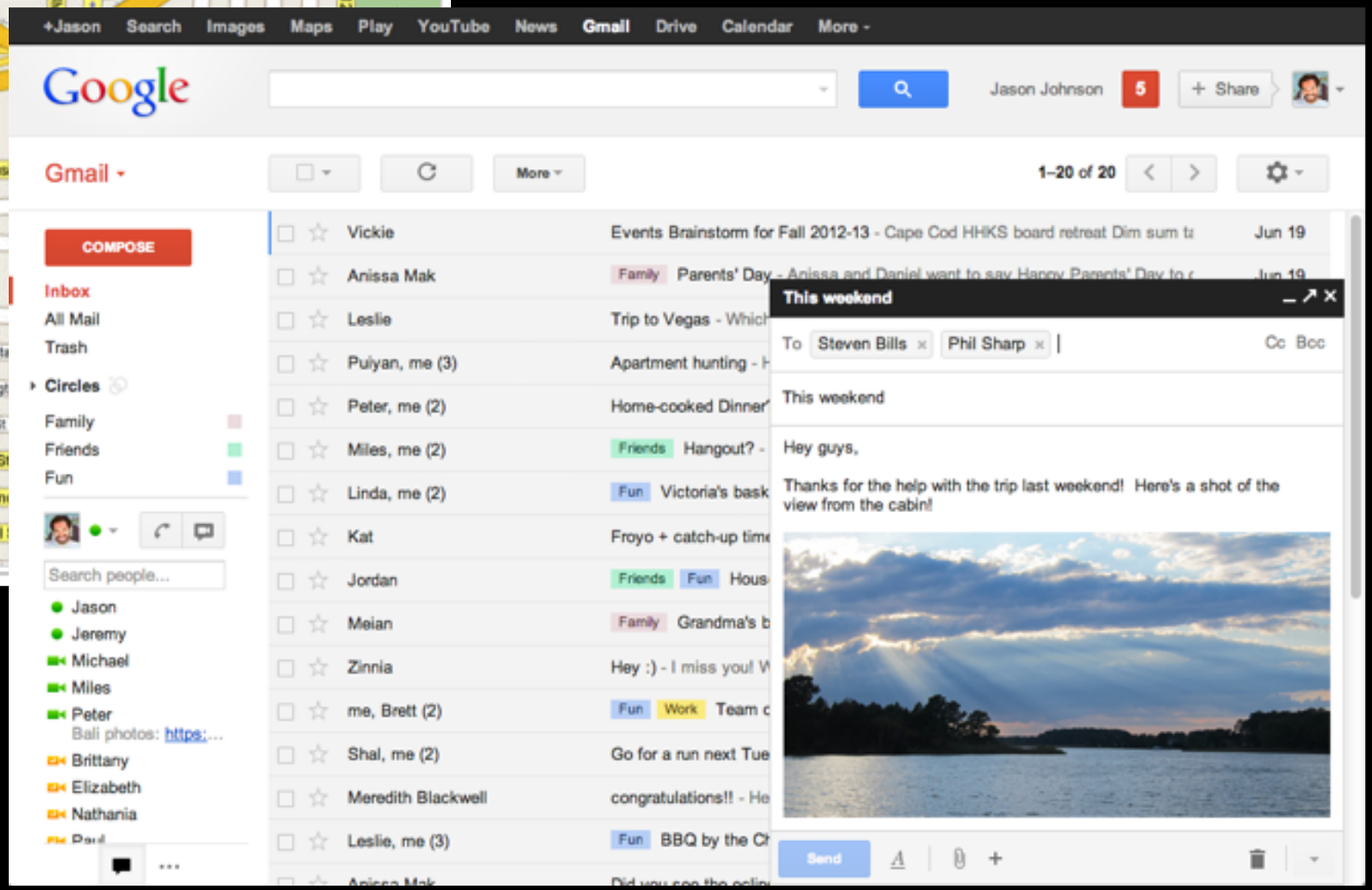
Microsoft Word

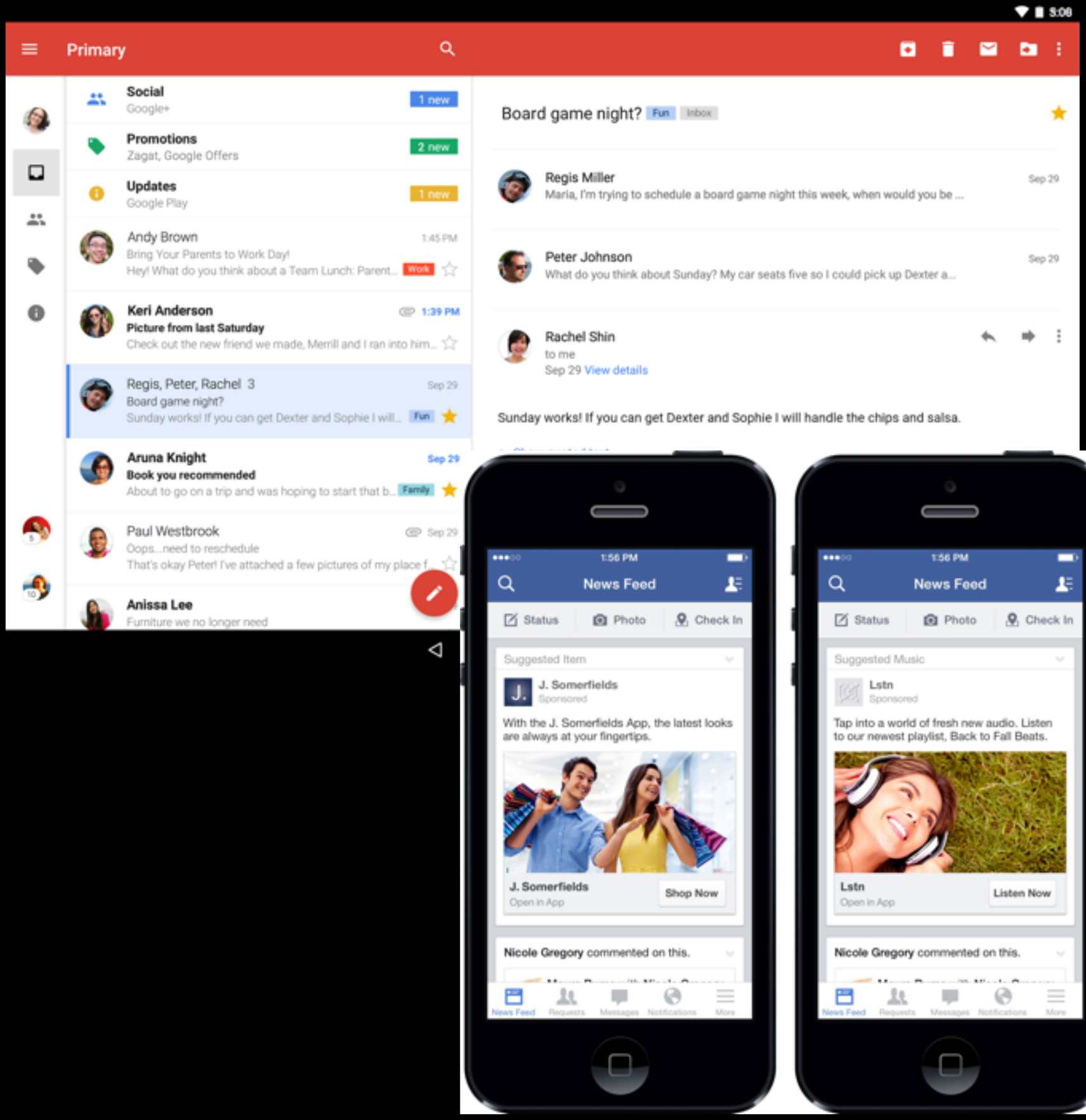
Version 6.0


What's New
Check Email
YAHOO!
My
Personalize
Help
Yahoo! Auctions
Pokemon, Beanies
Know when friends are online!
Click to download Yahoo! Messenger
Yahoo! Mail
free email for life
Search
advanced search
Yahoo! Shopping - Apparel, Computers, Videos, DVDs, CDs, Toys, Electronics and more
Shopping - Auctions - Yellow Pages - People Search - Maps - Travel - Classifieds - Personals - Games - Chat - Clubs
Mail - Calendar - Messenger - Companion - My Yahoo! - News - Sports - Weather - TV - Stock Quotes - more...
Arts & Humanities
Literature, Photography...
News & Media
In the News
Business & Economy
Companies, Finance, Jobs...
Computers & Internet
Internet, WWW, Software, Gam
Education
College and University, K-12...
Entertainment
Cool Links, Movies, Humor, Mu
Government
Elections, Military, Law, Taxes...
Health


Google!
Search the web using Google!
10 results
Google Search
I'm feeling lucky
Index contains ~25 million pages (soon to be much bigger)
About Google!
Stanford Search Linux Search
Get Google! updates monthly!
your e-mail
Subscribe
Archive
Copyright ©1997-8 Stanford University


NE Broadway St
NE Weidler St
NE Halsey St
NE Schuyler St
N Broadway St
N Winning Way
NE M L King Blvd
NE Grand Ave
NE 2nd Ave
NE 3rd Ave
NE 6th Ave
NE 7th Ave
NE 8th Ave
NE 9th Ave
NE 14th Ave
Rose Garden
Holladay Park Medical Center
NE Wasco St
Lloyd District
Lloyd Center
NE Multnomah
Holladay West Park
NW Front Ave
Broadway Bridge
Pacific Hwy W
Portland Union Station
NW Irving St
NW Hoyt St
NW Glisan St
NW Flanders St
NW Everett St
NW Naito Pkwy
NW 3rd Ave
NW Broadway
NW Couch St
Old Town - Chinatown
North Park Blocks
Portland
W Burnside St
Burnside Bridge
E Burnside
SW Oak St
Obryant Park
SW 6th Ave
SW 5th Ave
SW 4th Ave
SW 3rd Ave
SW 2nd Ave
SW Park Ave
SW Naito Pkwy
SW Alder St
Morrison Bridge
SW Morrison St
SW Taylor St
SW Salmon St
SW Main St
Tom McCall Waterfront Park
SE Washington
SE Alder St
SE Morrison St
SE Belmont
SE Yamhill
NE Pacific St
NE Oregon St
NE Lloyd Blvd
NE Irving
NE Glisan St
NE Martin Luther King Jr Blvd
99
99W
5
84
30
301
302A
300B
©2009 Google


+Jason
Search
Images
Maps
Play
YouTube
News
Gmail
Drive
Calendar
More
Google
Jason Johnson
5
+ Share
Gmail
More
1–20 of 20
COMPOSE
Inbox
All Mail
Trash
Circles
Family
Friends
Fun
Search people...
Jason
Jeremy
Michael
Miles
Peter
Bali photos: https:...
Brittany
Elizabeth
Nathania
Vickie
Events Brainstorm for Fall 2012-13 - Cape Cod HHKS board retreat Dim sum tz
Jun 19
Anissa Mak
Family
Parents' Day
Leslie
Trip to Vegas
Pulyan, me (3)
Apartment hunting
Peter, me (2)
Home-cooked Dinner
Miles, me (2)
Friends
Hangout?
Linda, me (2)
Fun
Victoria's bask
Kat
Froyo + catch-up time
Jordan
Friends
Fun
Meian
Family
Grandma's b
Zinnia
Hey :) - I miss you!
me, Brett (2)
Fun
Work
Team c
Shal, me (2)
Go for a run next Tue
Meredith Blackwell
congratulations!!
Leslie, me (3)
Fun
BBQ by the Ch
This weekend
To
Steven Bills
Phil Sharp
Cc
Bcc
This weekend
Hey guys,
Thanks for the help with the trip last weekend! Here's a shot of the view from the cabin!
Send


5:00
Primary
Social
Google+
1 new
Promotions
Zagat, Google Offers
2 new
Updates
Google Play
1 new
Andy Brown
1:45 PM
Bring Your Parents to Work Day!
Hey! What do you think about a Team Lunch: Parent...
Work
Keri Anderson
1:39 PM
Picture from last Saturday
Check out the new friend we made, Merrill and I ran into him...
Regis, Peter, Rachel 3
Sep 29
Board game night?
Sunday works! If you can get Dexter and Sophie I will...
Fun
Aruna Knight
Sep 29
Book you recommended
About to go on a trip and was hoping to start that b...
Family
Paul Westbrook
Sep 29
Oops...need to reschedule
That's okay Peter! I've attached a few pictures of my place f
Anissa Lee
Furniture we no longer need
Board game night?
Fun
Inbox
Regis Miller
Sep 29
Maria, I'm trying to schedule a board game night this week, when would you be ...
Peter Johnson
Sep 29
What do you think about Sunday? My car seats five so I could pick up Dexter a...
Rachel Shin
to me
Sep 29 View details
Sunday works! If you can get Dexter and Sophie I will handle the chips and salsa.
1:56 PM
News Feed
Status
Photo
Check In
Suggested Item
J. Somerfields
Sponsored
With the J. Somerfields App, the latest looks are always at your fingertips.
J. Somerfields
Open in App
Shop Now
Nicole Gregory commented on this.
News Feed
Requests
Messages
Notifications
More
1:56 PM
News Feed
Status
Photo
Check In
Suggested Music
Lstn
Sponsored
Tap into a world of fresh new audio. Listen to our newest playlist, Back to Fall Beats.
Lstn
Open in App
Listen Now
Nicole Gregory commented on this.
News Feed
Requests
Messages
Notifications
More
1:52 PM
News Feed
11 Likes 2 Comments
Like
Comment
Suggested Travel
HotelTonight
Sponsored
HotelTonight is now in over 100 destinations! Be spontaneous and book a weekend getaway.
Hotel Tonight
HotelTonight
Open in App
Book Now
News Feed
Requests
Messages
Notifications
More

Single **P**age **A**pps

(SPA)

Rendering is done on the client side

No refresh/reload

Which is cool but comes with a price

Data communication
is done via AJAX

A lot of hard problems are now client-side

Client side SPA history

Vanilla JavaScript

Suddenly, JS was a real language

jQuery

MVC start - Backbone

Awesome MVC

Angular Introduction

Binding *Magic*

Strict components

And stuff all MVC frameworks provide

Breakdown of demo

```
<div>{{ myModel.text }}</div>
```

```
<input type=text
ng-model=“myModel.text”>
```

Where’s myModel defined?

```
angular.module(‘exampleApp’)
.controller('ExampleCtrl',
function($scope) {
    $scope.myModel = {
        text: 'Initial text value'
    };
});
```

Model View Whatever

Todo List Web App

Data Model

```
angular.module('todoApp').controller('TodoCtrl',
function($scope) {
  $scope.todos = [
    {
      task: 'Buy milk',
      done: false
    },
    {
      task: 'Learn Angular',
      done: false
    },
    {
      task: 'Record Game of Thrones',
      done: false
    }
  ];
});
```

Displaying a list

```
<ul>
    <li ng-repeat="todo in todos”>
        {{ todo.task }}
    </li>
</ul>
```

Adding a todo

```
<input type="text" ng-model="newTodo.task">
<button ng-click="addTodo()">Add</button>

<ul>
    <li ng-repeat="todo in todos">
        {{ todo.task }}
    </li>
</ul>
```

Adding a todo

```
$scope.addTodo = function() {
    $scope.todos.push({
        task: $scope.newTodo.task,
        done: false
    });
    $scope.newTodo = {task: ''};
};
```

Marking as done

```
<ul>
    <li ng-repeat="todo in todos">
        <label>
            <input type="checkbox" ng-model="todo.done">
            {{ todo.task }}
        </label>
    </li>
</ul>
```

Hiding done todos

```
<ul>
    <li ng-repeat="todo in todos | filter:{done: false}">
        <label>
            <input type="checkbox" ng-model="todo.done">
            {{ todo.task }}
        </label>
    </li>
</ul>
```

Many more directives

- ng-class
- ng-if
- ng-change
- ng-src
- ng-href
- etc. etc.
- Google is your friend!

Services

Good design

==

break things to smaller pieces

Can’t put all logic in controllers

Services are plain objects that can be injected

```
angular.module('todoApp').factory('TodoRepository', function() {
    return {
        getAllTodos: function() {
            return [
                {task: 'Buy milk', done: false},
                {task: 'Learn Angular', done: false},
                {task: 'Record Game of Thrones', done: false}
            ];
        }
    };
});

angular.module(‘todoApp').controller('TodoCtrl',
function($scope, TodoRepository) {
    $scope.todos = TodoRepository.getAllTodos();
});
```

Async 101

Server data is always fetched asynchronously

Angular uses the **Promise**
design pattern for
handling async code

Promises objects are returned from async functions

A promise will, sometime later, either **resolve** or be **rejected**

```
var aPromise = doSomethingAsync();
aPromise.then(
    function(successValue) {
        console.log('Called on success');
    },
    function(rejectionReason) {
        console.log('Called on errors');
    }
);
```

```
angular.module('todoApp').factory('TodoRepository', function($q) {
    return {
        getAllTodos: function() {
            return $q.when([
                {task: 'Buy milk’, done: false},
                {task: 'Learn Angular’, done: false},
                {task: 'Record Game of Thrones’, done: false}
            ]);
        }
    };
});
```

```
angular.module(‘todoApp').controller('TodoCtrl',
function($scope, TodoRepository) {
  TodoRepository.getAllTodos().then(
      function(todos) {
          $scope.todos = todos;
      }
    );
});
```

Promises

- promise.then(successCallback, errorCallback);
- promise.then(successCallback);
- promise.then(null, errorCallback);
- promise.catch(errorCallback);
- promise.finally(callback);

Promises are chainable

```
somePromise.then(
    function(promiseResult) {
        return 'The value is ' + promiseResult;
    }
).then(
    function(currentResult) {
        // Here currentResult === 'The value is ...'
    }
);
```

And nestable

```
somePromise.then(
    function(promiseResult) {
        return anotherPromise;
    }
).then(
    function(promiseResult) {
        // This is only called after anothePromise resolves,
        // and promiseResult is the result of anotherPromise
    }
)
```

And allow error handling in the flow

```
somePromise.catch(
    function(rejection) {
        return 'good';
    }
).then(
    function(result) {
        // This is called and result === 'good'
        // If you want the rejection to keep
        // flowing you should have returned
        // $q.reject(rejection)
    }
);
```

Angular App Boilerplate

<script> with dependencies:
angular, jQuery,
angular-route, etc.

<html ng-app="myApp">

```
angular.module(
‘myApp’, [‘ngRoute’]);
```

```
angular.module(‘nameOfAppModule')
.controller(…);
```

<div ng-view>

Configure routes

```
angular.module('nameOfAppModule').config(function($routeProvider) {
    $routeProvider.when('/route', {
        templateUrl: '/path/to/template.html',
        controller: 'RouteCtrl'
    })
    .when('/route2', {
        templateUrl: '/path/to/template2.html',
        controller: 'Route2Ctrl'
    })
    .otherwise({redirectTo: '/route'});
});
```

Directives

In addition to angular's we can write our own

Directives allow DOM access

Directives let us create reusable UI components

DOM access

```
angular.module('todoApp').directive('myEnter', function() {
  return {
    link: function(scope, element, attrs) {
      var handler = element.bind('keyup', function(event) {
        if (event.keyCode === 13) {
          scope.$apply(function() {
            scope.$eval(attrs.myEnter);
          });
        }
      });

      scope.$on('$destroy', handler);
    }
  };
});
```

```
<input type=text ng-model="newTodo.task"
my-enter="formatInput()">
```

Reusable components

```
angular.module('todoApp').directive('todoItem', function() {
    return {
        templateUrl: 'path/to/template.html',
        scope: {
            todo: '='
        },
        controller: 'TodoItemCtrl'
    };
});
```

```
<div ng-repeat="todo in todos">
  <div todo-item="todo"></div>
</div>
```

Isolated scopes

How we write maintainable directives

```
angular.module('todoApp').directive(
    'myIsolatedDirective', function() {
    return {
        scope: {
            callback: '&'
        },
        link: function(scope) {
            // ... Something happens and:
            scope.callback({argument: 'the value'});
        }
    };
});
```

<div my-isolated-directive
callback="callback(argument)">

Routing with angular-route

```
angular.module('app').config(function($routeProvider) {
    $routeProvider.when('/state', {
        templateUrl: '/template.html',
        controller: 'MyCtrl'
    });
});
```

```
<div ng-view></div>
```

2 ways to change states

- State
- $location.path('/state');

```
angular.module('app').config(function($routeProvider) {
    $routeProvider.when('/project/:projectId', {
        templateUrl: '/template.html',
        controller: 'MyCtrl'
    });
});
```

$routeParams.projectId

http://example.com/#/project/123?new=true

$routeParams == {projectId: ‘123’, new: ‘true’}

Angular's closed garden

Handling the garden

- Don’t use setTimeout(), use $timeout or $interval
- Don’t use jQuery for AJAX, use $http
- When you use your own handlers (like click handlers that aren’t ng-click) you have to wrap it:

```
element.bind(‘click’, function() {
    $scope.$apply(function() {
        // handle click
    });
});
```