
Google™
10

Opening up Closure Library

Nathan Naze

May 19, 2010

View live notes and ask questions about this session on Google Wave:

http://bit.ly/9NEdA3

- **What is Closure Library?**
- Getting started with Closure Library
- The reference manual

What is Closure Library?

- Google's "Standard Library" for JavaScript
- With Closure Compiler, a typed, object-oriented means of collaboratively developing complex web applications
- The JavaScript library behind Google's web apps

Problems we're trying to solve

Large-Scale Web Development

- JavaScript + HTML/CSS/DOM
- Collaborative development
- Managing complexity
- Modularization and conventions
- Testing and stability

"Let's make the tools better"

Enter Closure Tools

- Closure Library
  - Started in 2005
  - all 20% contributions
  - 400+ engineers have contributed
  - The focus of this talk
  - 250k+ lines of JS (not including tests)
- Closure Compiler
  - JavaScript-to-JavaScript compiler
  - Type checker
- Closure Templates
  - Client- and server-side templating

Closure Library

- Namespaces allow for a broad library
- Types and type-checking
  - Object-oriented
  - Type enforcement by Closure Compiler
    - "Type hinting" - not required
- Shared, tested code
  - More users means more found bugs
  - More eyeballs mean more fixed bugs

Geared towards complex, collaborative UI development

Why use Closure Library?

- It's modular and broad
- Geared for complex applications
  - Type safety
- Very stable and tested
  - It's what runs many Google web apps
- Works well with Closure Compiler

Why not use Closure Library?

- Steeper learning curve
- It's lower level
- Intended for big team, collaborative development.
- Use the right tool for the job.

Namespacing

Namespaces 101

goog.provide('foo.bar') -> "I provide this namespace."

goog.require('foo.baz') -> "I require this namespace."

The result is a giant dependency tree.

This is unlike single-file libraries.

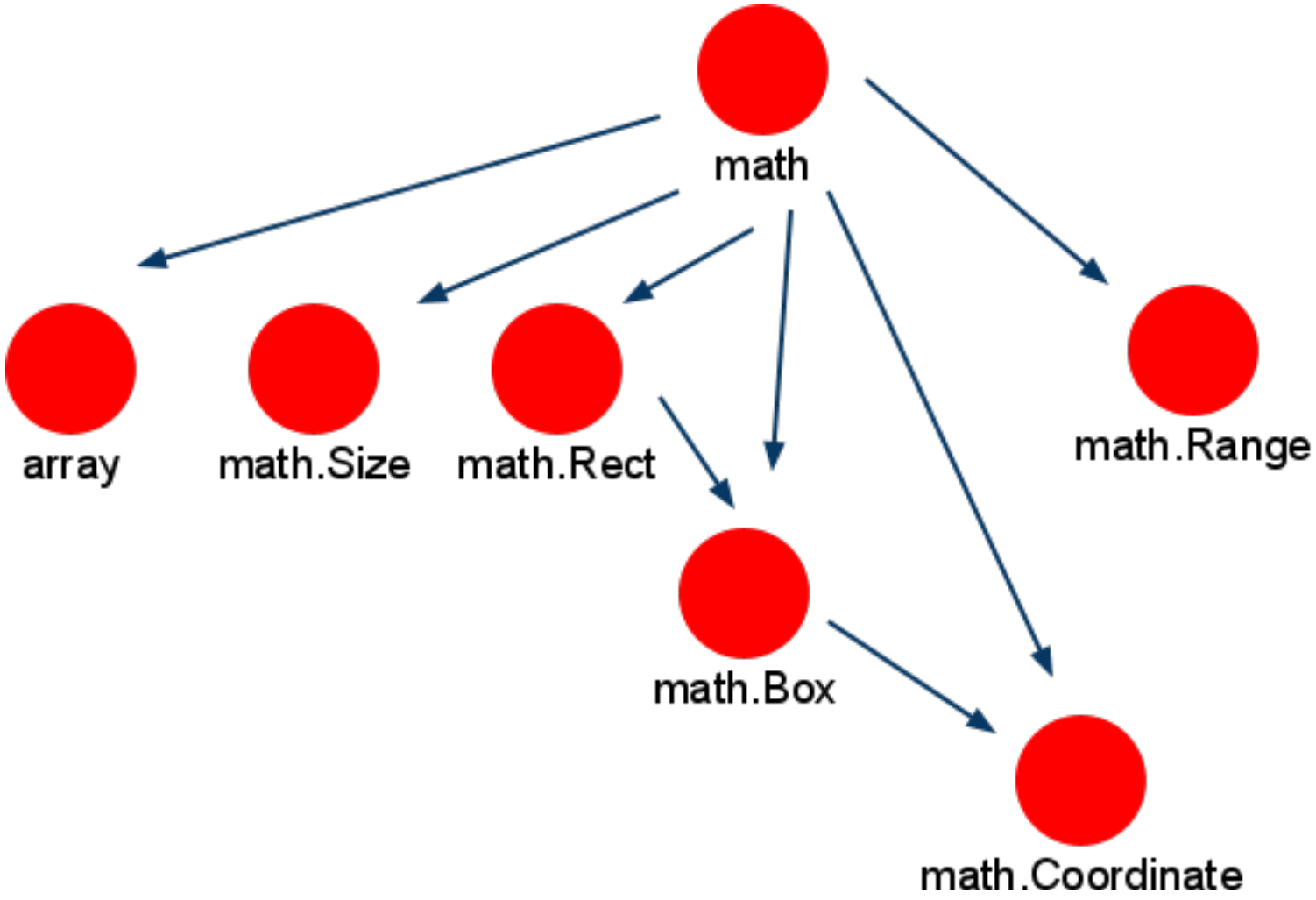
- Google AJAX loader isn't applicable here.
- Pull a the subtree of what you use.

Namespacing dependency example

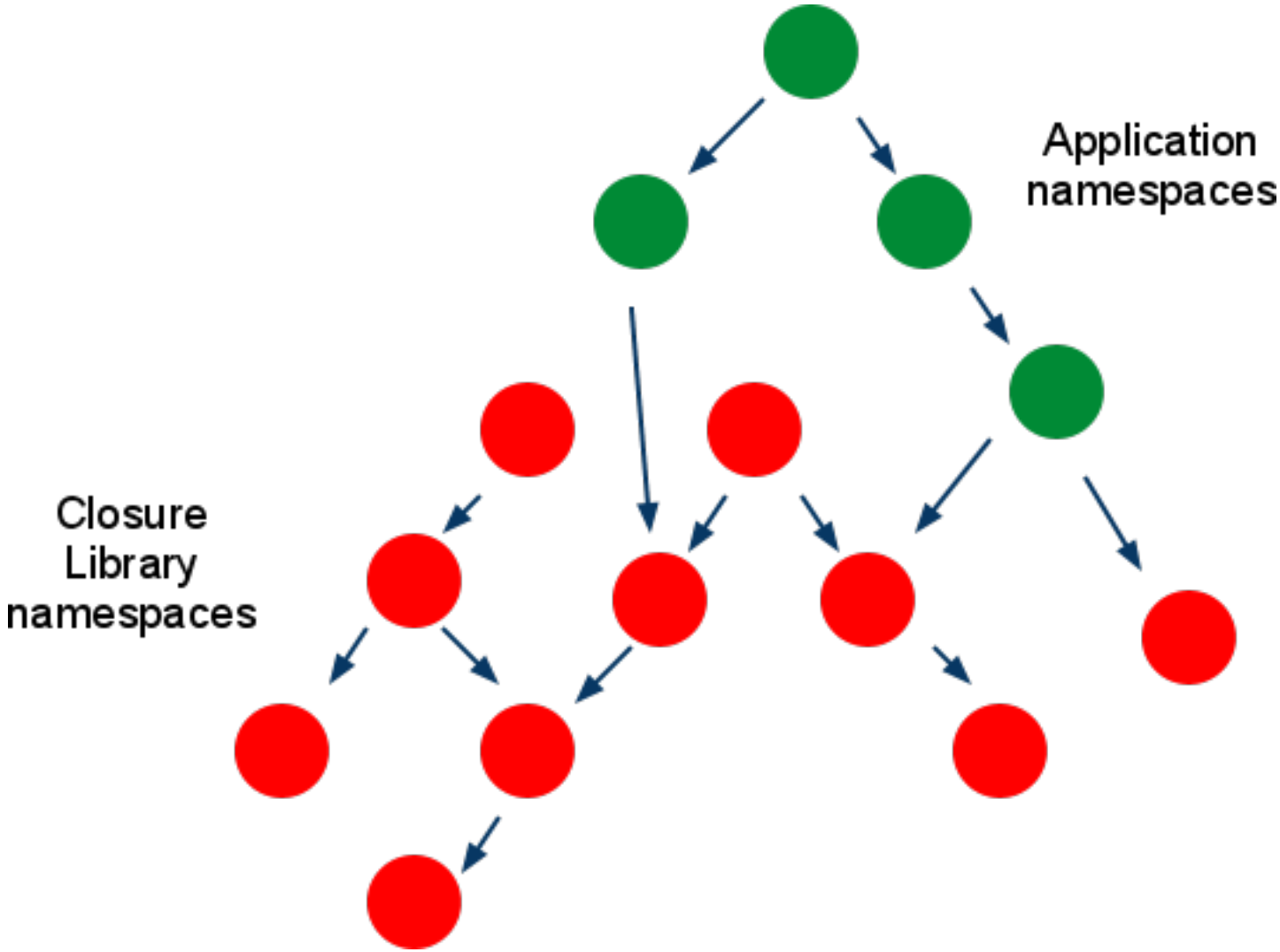
```
goog.provide('goog.math');

goog.require('goog.array'); goog.require('goog.math.
Box'); goog.require('goog.math.Coordinate'); goog.
require('goog.math.Range'); goog.require('goog.math.
Rect'); goog.require('goog.math.Size');
```

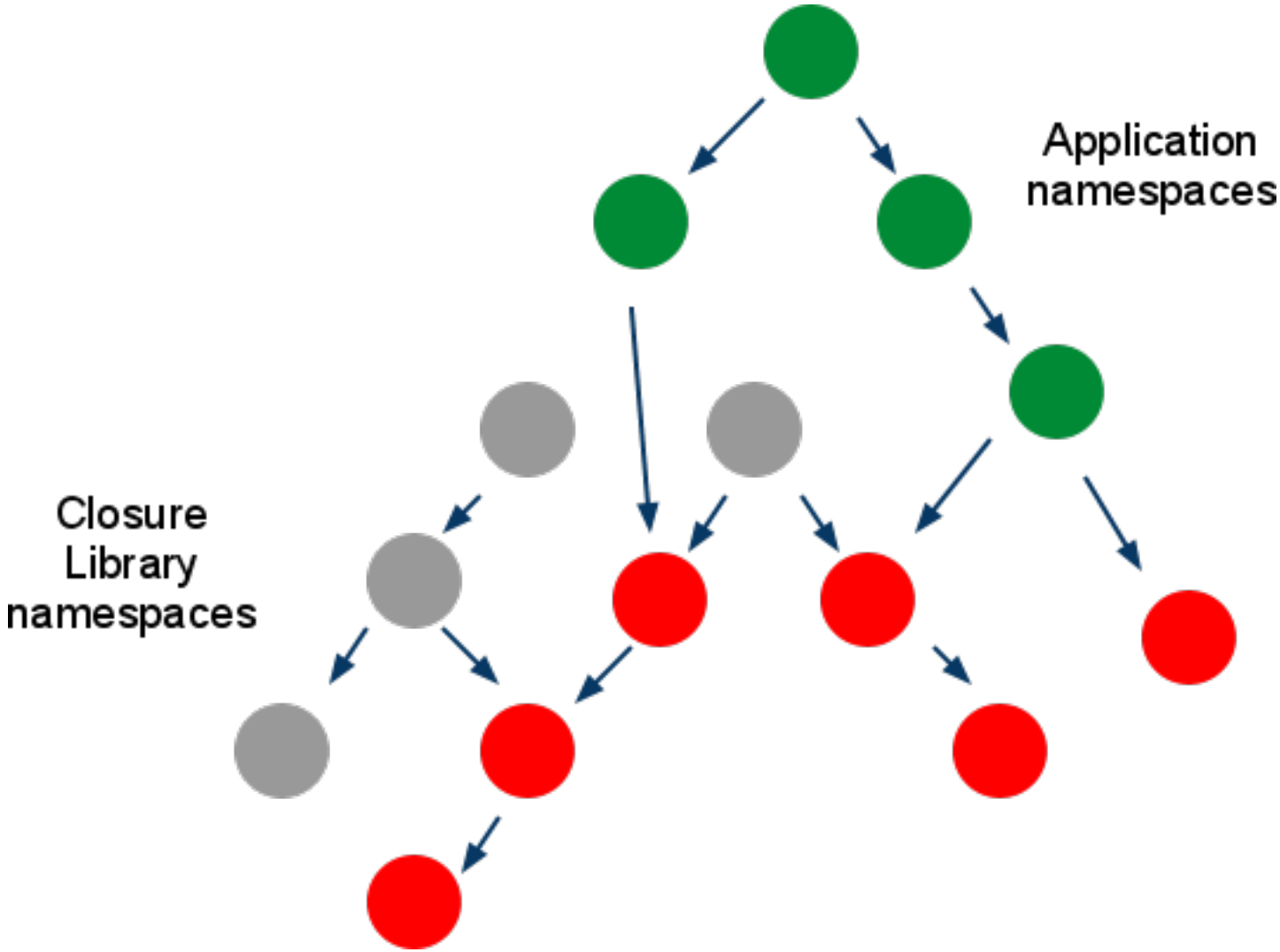
Namespacing dependency example



Subtrees of Closure Library



Subtrees of Closure Library



Type checking

"Stronger-typed" JavaScript

Why?

- Essential to collaboration
  - Self-documenting code
  - Encapsulation
    - Interfaces and implementations

How?

- JSDoc tags
- Type annotations enforced by Closure Compiler
  - "Type hinting" - not required

Typing example

```
/**
 * Returns the sum of the arguments.
 * @param {...number} var_args Numbers to add.
 * @return {number} The sum of the arguments.
 */
goog.math.sum = function(var_args) {
  return /** @type {number} */ (goog.array.reduce(arguments,
      function(sum, value) {
    return sum + value;
  }, 0));
};
```

Constructor example

```
/**
* DatePicker widget.
* @param {goog.date.Date|Date=} opt_date
* @param {Object=} opt_dateTimeSymbols
* @constructor
* @extends {goog.ui.Component}
*/
goog.ui.DatePicker = function(opt_date,
opt_dateTimeSymbols) {
goog.ui.Component.call(this);
// ...
};
goog.inherits(goog.ui.DatePicker, goog.ui.Component);
```

Closure Compiler

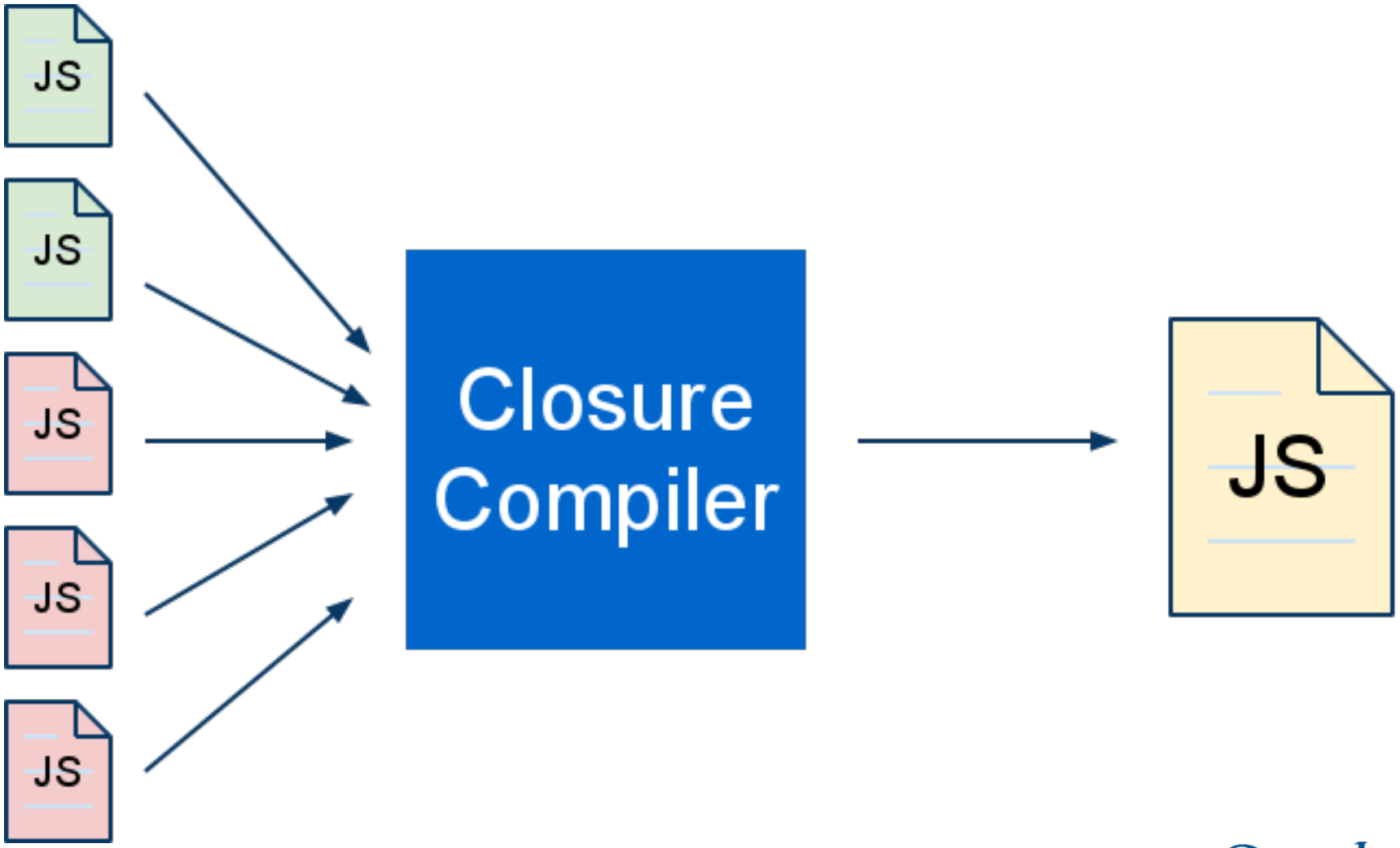
Closure Compiler

"The (Type) Enforcer"

- Efficiency - Lowers code size
  - Renaming
  - Removes dead code
  - Inlines functions
  - and more
- Code checking
  - Checks types
    - Function/method calls
  - Syntax errors
  - Understands user-defined types

Closure Compiler

A Visual Guide



- What is Closure Library?
- **Getting started with Closure Library**
- The reference manual

The Code

How to get it

- Homepage
  - http://code.google.com/p/closure-library
- SVN
  - http://code.google.com/closure/library
  - Periodic updates from Perforce canonical
    - We want to increase that rate
    - "Make Open Easy"
  - svn:externals if you're using SVN
  - Continuous integration - no versions

Server agnosticism

- Server agnostic
  - Pros
    - Inside Google, FEs are all over the map: C++, Python, Java, and more.
    - We don't want to tie to a server setup, language, or framework.
  - Cons
    - You are responsible for the glue.
    - Makes it harder to just start working with Closure.

The Code

How to get started

- base.js
  - Sets up the loading system
  - loads deps.js

- deps.js
  - registers where each namespace is stored
  - goog.addDependency(path, provides, requires)

User-defined namespaces use the same tools to make their own dependency files.

The Code

Example

```
<head>

<script src=".../path/to/goog/base.js">
</script>

<!-- possible additional deps files -->

<script src=".../path/to/mycode.js">
</script>

</head>
```

Example

Debug vs. Compiled

- Debug
  - The code you're editing
  - Important to run uncompiled
  - Never in production
- Compiled
  - Your code, run through Closure Compiler
  - One big file
  - Small over the wire
  - Dependency tools prepare your compiler input

More on dependency files

- Tools in flux, but
  - Scans files for goog.provide/goog.require
  - Writes out a dependency file
- Dependency tools
  - bin/calcdeps.py
    - legacy
  - bin/build/closurebuilder.py
    - new python implementation
  - building support into Closure Compiler
  - Better support for non-*NIX

- What is Closure Library?
- Getting started with Closure Library
- **The reference manual**

The Codebase

- Self-documented, broad
  - Generated Docs
- Event-driven programming
  - goog.events.EventTarget
  - Observer/listener design pattern
    - Not just DOM nodes
- Basic utilities (primitives, DOM, style, network) through UI widgets, animation, etc.

Naming conventions

- Where to find code
- In most cases, namespace matches file path.

- Examples:
- goog.dom -> goog/dom/dom.js
- goog.style -> goog/style/style.js
- goog.net.XhrIo -> goog/net/xhrio.js

Basic utilities

- goog.array
- goog.string
- goog.math.*
- goog.object
- goog.Uri

Data Structures

- In goog/structs

DOM and style

- DOM
  - goog.dom
  - goog.dom.classes
  - See all of goog/dom/...
- Style
  - goog.style
  - goog.window

More

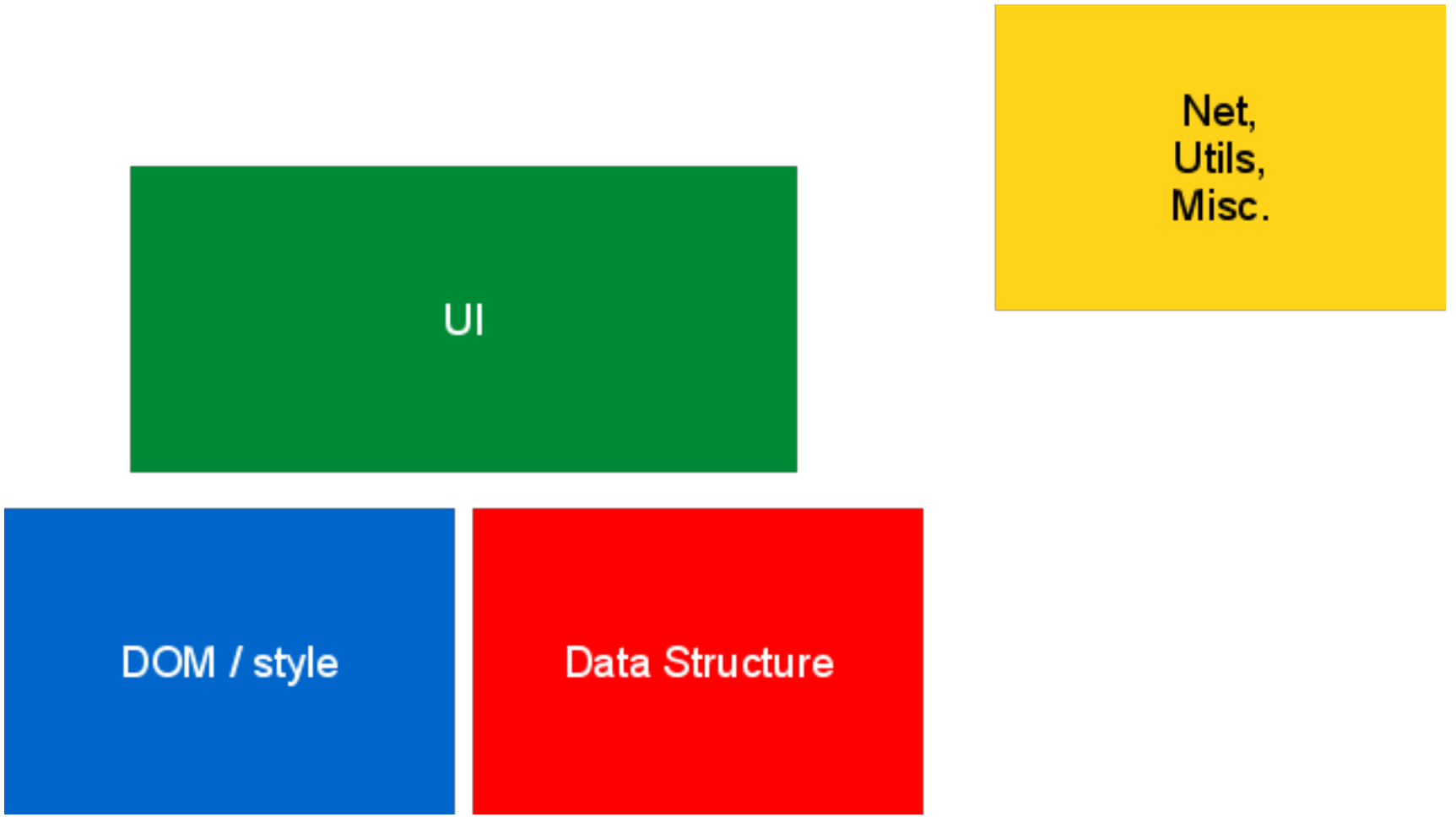
- Net
- goog.net.XhrIo and others
- goog.net.Jsonp
- goog.net.IframeIo
- goog.net.BrowserChannel
- ...
- Formats
- goog.json
- Misc
- Time: goog.Date, goog.Delay, goog.Timer
- Editor: goog.editor

UI

- Base concept:
- goog.ui.Component

- Subclasses:
- goog.ui.Control
- goog.ui.Container
- Too many to list here. See: goog/ui

Big Picture



Closure Library test suite

http://closure-library.googlecode.com/svn/trunk/all_tests.html

Unit Tests

Each class or namespace has an accompanying unit test

- goog.testing.*
  - Traditional asserts
  - Handling of async
  - Mocking tools

Unit Tests

- Server farm runs all tests on all browsers
- submit queue
- continuous build

The future

- Google and open source
  - "an open-source project" vs "published source code"
- Closure Library (and the other tools)
  - Not started with the thought of open source
  - Lots of internal build system
  - Non-Google code in repositiory

Announcements

- New Blog!
  - closuretools.blogspot.com
- New Twitter!
  - twitter.com/closuretools
- Same Google Group!
  - closure-{library,compiler,templates}-discuss
- Open Source Community
  - Now accepting patches!

Q&A

Google Wave:
Contact:
http://bit.ly/9NEdA3

code.google.com/closure/library

groups.google.com/group/closure-library-discuss

closuretools.blogspot.com

twitter.com/closuretools

Closure Compiler office hours:
Space B, 2:30pm - 5:00pm


Google™
10