Google's Closure Tools

Tom Payne / Camptocamp SA

24 March 2011

Closure Tools

- Library
- Compiler
- Linter
- Templates

→ http://code.google.com/closure/
→ http://code.google.com/p/google-styleguide/

Library

- Extensive
- Modular
- Cross-browser
- Tested
- Well documented
- A “standard library” for Javascript

→ http://code.google.com/closure/library/

Compiler

- Compiles Javascript to Javascript
- Output is a monolithic Javascript file
- Minimiser
- Optimiser
- Tightly integrated with library

→ http://code.google.com/closure/compiler/

Compilation levels

1. Whitespace only
2. Simple optimizations (80% / $1.25\times$, 85% / $1.2\times$ gzip'ed)
3. Advanced optimizations (25% / $4\times$, 50% / $2\times$ gzip'ed)

Optimization

- Faster
- Smaller
- Correct
- Compresses well

→ http://code.google.com/p/closure-compiler/source/browse/

Language extensions

- Uses @jsdoc tags in comments
- Strict, static type checking
- Constructors and interfaces
- Public, protected and private methods and attributes
- Constants, typedefs and enums
- Pre-processor

→ http://code.google.com/closure/compiler/docs/js-for-compiler.html
→ http://code.google.com/closure/compiler/docs/limitations.html

Name mangling

- Internally consistent
- Properties only, not strings
  - `obj.prop = 1; obj['prop'] += 1; // will fail`
- Need to explicitly specify exported symbols ("exports")
- Need to explicitly specify imported symbols ("externs")
- Can write interface files for external libraries

→ http://code.google.com/closure/compiler/docs/api-tutorial3.html

Dependencies

- In each source file (module):
  - Declare provides with `goog.provides`
  - Declare requirements with `goog.require`
- Throw everything at `depswriter.py`/`closurebuilder.py`
- Emits only what you need

Uncompiled code

- Load three scripts:
  1. `<script src="closure/goog/base.js">`
  2. `<script src="deps.js">`
  3. `<script>goog.require('my.module');</script>`
- `depswriter.py` generates `deps.js` (the map between modules and source files)
- `goog.require` loads source files as needed
- Great for debugging

→ http://code.google.com/closure/library/docs/depswriter.html

Compiled code

- `closurebuilder.py` builds monolithic JS files
- Load one script:
  - `<script src="compiled.js">`
- Pass `--namespace=my.module` to `closurebuilder.py` to seed
- Hard to debug
- FireBug extension

→ http://code.google.com/closure/library/docs/closurebuilder.html
→ http://code.google.com/closure/compiler/docs/inspector.html

Gotchas

- Name mangling
- Mismatch from imposing strict typing on a dynamic language
- Differences between compiled and uncompiled code
- No `$(document).ready()` by design

Practical experience

- Makes Javascript more like Java :-(/ :-)
- Refactoring required to use advanced optimizations
- Long compile times (JVM startup, but multithreaded)
- Interfacing with external packages can be tiresome
- Catches potential bugs
- Allows compiler to generate better code
- Improves my code

Demonstration


```
git clone https://github.com/twpayne/closure-toy.git
cd closure-toy
make
```