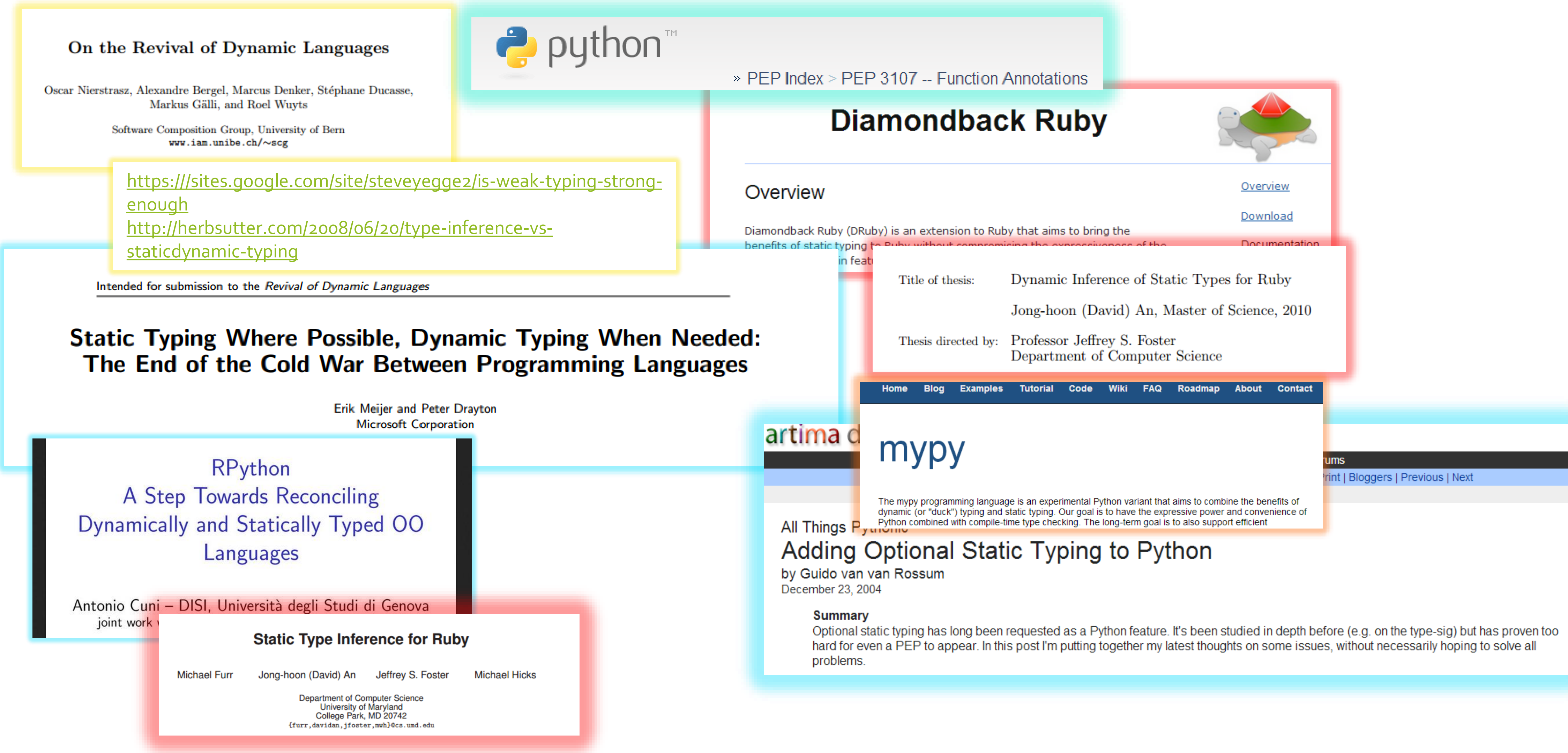
ANGULARJS IN TYPESCRIPT

INTRO

- Dynamic languages with optional static typing: Dart, Typescript,…
- A -somewhat- new idea…
- Where did this idea come from?
- http://www.slideshare.net/RichardWarburton/twins-oop-and-fp-31794436



On the Revival of Dynamic Languages

Oscar Nierstrasz, Alexandre Bergel, Marcus Denker, Stéphane Ducasse, Markus Gälli, and Roel Wuyts

Software Composition Group, University of Bern
www.iam.unibe.ch/∼scg

https://sites.google.com/site/steveyegge2/is-weak-typing-strong-enough
http://herbsutter.com/2008/06/20/type-inference-vs-staticdynamic-typing

python™

» PEP Index > PEP 3107 -- Function Annotations

Diamondback Ruby

Overview

Diamondback Ruby (DRuby) is an extension to Ruby that aims to bring the benefits of static typing

in feat

Overview
Download
Documentation

Title of thesis: Dynamic Inference of Static Types for Ruby

Jong-hoon (David) An, Master of Science, 2010

Thesis directed by: Professor Jeffrey S. Foster
Department of Computer Science

Intended for submission to the *Revival of Dynamic Languages*

Static Typing Where Possible, Dynamic Typing When Needed:
The End of the Cold War Between Programming Languages

Erik Meijer and Peter Drayton
Microsoft Corporation

RPython
A Step Towards Reconciling
Dynamically and Statically Typed OO
Languages

Antonio Cuni – DISI, Università degli Studi di Genova
joint work

Static Type Inference for Ruby

Michael Furr Jong-hoon (David) An Jeffrey S. Foster Michael Hicks

Department of Computer Science
University of Maryland
College Park, MD 20742
{furr,davidan,jfoster,mwh}@cs.umd.edu

mypy

The mypy programming language is an experimental Python variant that aims to combine the benefits of dynamic (or "duck") typing and static typing. Our goal is to have the expressive power and convenience of Python combined with compile-time type checking. The long-term goal is to also support efficient

artima d

All Things P

Adding Optional Static Typing to Python
by Guido van van Rossum
December 23, 2004

Summary
Optional static typing has long been requested as a Python feature. It's been studied in depth before (e.g. on the type-sig) but has proven too hard for even a PEP to appear. In this post I'm putting together my latest thoughts on some issues, without necessarily hoping to solve all problems.

ABOUT ME

- FORTRAN, Pascal, Lisp, C/C++, C#, JS
- JavaScript: Freedom, a bit like C
- Engineer, currently at Pearson as UX developer
- Been working with typescript since November 2012
- Projects I currently work on have ~30k lines of ts

CONTENT OF THIS TALK

- Background and overview
- Anders Hejlsberg
- What typescript gives you: ES6 stuff, typing, tooling, documentation of intend
- Where to get definition files => definitely typed
- Demo
- AngularJS related benefits
- Install none-MS tool chain, Node, npm, typescript, grunt, grunt plugin, sublime plugin

BACKGROUND AND OVERVIEW

- Transpiler->JS,
  - https://github.com/jashkenas/coffee-script/wiki/List-of-languages-that-compile-to-JS
- JS becomes a bit like byte code to Java or IL for C#, with the very important difference that it is human readable
- Optional typing, platform independent
- Very important: Typescript does not take away any of the Java script "coolness", in my opinion

ANDERS HEJLSBERG

- Language designer (his own words)
- Turbo Pascal, Borland
- Delphi
- C#
- “Can’t explain it in 5min -> not a good idea”, maybe
- “Typescript filters out the semantic subset of java script that makes sense”

WHAT YOU GET …

- Ecmascript6 stuff: module, class, extends, arrow function
  - https://github.com/lukehoban/es6features?utm_source=javascriptweekly&utm_medium=email#arrows
  - https://wiki.mozilla.org/ES6_plans
  - => handling of this pointer (lexically scoped)
  - ~ like coffee script, but not a new language
- Optional static typing, interface, implements
  - interfaces open ended, multiple files can contribute
  - annotations
- Static typing: not that there is anything wrong with that
  - inference, flowing the types via generics
  - inverse inference
  - lib.d.ts, the JS runtime lib and DOM
  - JS has type information but only at run time, when its too late IMHO
  - in typescript, not provably type safe
  - http://en.wikipedia.org/wiki/type_system

MORE

- Statement completion
  - Sublime Text and many others, when working with large libraries very helpful
- Documentation of intend
  - example: rest service format described via interface
  - write a test for what you actually get from the service against the interface
  - if you have to go back and talk with the backend people about their data, you have a piece of code as proof
- No TS “engine”/VM, no runtime library
  - impact only at design time
- Superset of JS
  - for people who know java script a shallow learning curve
- Debugging in Chrome, Firefox…
  - directly in TS files via source maps
- Upgrading to a newer version of a library?
  - upgrade the .d.ts file and fix the errors
- Handling of java script module systems AMD/CommonJS
  - via compiler flag

DEMO

- Simple examples
    - ES6 features: module, class, arrow
-
- Question
    - var t = "3" + 2;
    - var t1 = "3" - 2;
    - console.log(t); //????
    - console.log(t1); //????
- Complex examples
    - inference, reverse inference, refactoring
    - changing an interface from a staging to a production scenario, rest calls, js module systems
- Code re-use
    - name spaces/modules simplify code reuse
    - more utility functions used in many places, changing these becomes very simple
- Unit tests
    - saves unit tests.
- Convert d3.js GIST, Mitchell best candidate layout
    - first to TS, then to a NG directive

WHERE DOES TYPESCRIPT NOT HELP?

- AngularJS expressions in the html, typescript doesn’t know about that
- Accidental closures when refactoring
- Check types at run time (Dart does)
- Versions of libraries and corresponding .d.ts file can be a problem
- .d.ts files can be incorrect or incomplete or both

INSTALL: WHAT IS YOUR TOOL CHAIN?

- General
  - http://www.typescriptlang.org/
  - http://typescript.codeplex.com/documentation
- Nodejs package
  - install node, node-inspector if you want to debug the example code
  - npm install typescript -g
  - gives you the command line compiler-> tsc
- Grunt
  - http://gruntjs.com/ npm install -g grunt
  - https://npmjs.org/package/grunt-typescript
  - npm install grunt-typescript --save-dev
- Sublime text
  - browse packages,
    git clone https://github.com/Railk/T3S.git or git@github.com:Railk/T3S.git
- Voldemort
  - http://www.microsoft.com/en-us/download/details.aspx?id=34790 needs windows 7 sp1 and IE
- Typescript compile
  - compile typescript script tags and ts files directly in the browser: https://github.com/niutech/typescript-compile
- Where to get the definitions -> definitely typed
  - https://github.com/borisyankov/DefinitelyTyped
  - https://github.com/definitelytyped/tsd
  - http://definitelytyped.github.io/tsd/

QUESTIONS

- Thank you!

…some more links

- Demo using ts for angular 15min run down by Ali Basarat: http://www.youtube.com/watch?v=0-6vT7xgE4Y&hd=1
- Grunt plugins:
- https://www.npmjs.org/package/grunt-ts
- https://www.npmjs.org/package/grunt-typescript