Highway to the external world: Generate externs programmatically

Andy Li

WWX2016

Space Elevator by GlennClovis
http://glennclovis.deviantart.com/art/Space-Elevator-447553527

Andy Li

- from Hong Kong
- spoke at WWX 2015:
  “Continuous Integration for Haxe Projects”
- working full-time for the Haxe Foundation
  - jQuery extern
  - continuous integration
  - packaging (Linux, Homebrew, Chocolatey etc.)
  - misc: haxelib, website backends etc.

The External World by David OReilly
http://www.davidoreilly.com/films/the-external-world/

Importance of Externs

- interfacing native platform APIs
- interfacing native libraries
- interfacing legacy native code

« Conditional Compilation
Introduction
Types
Type System
Class Fields
Expressions
Conditional Compilation
Static Extension
String Interpolation
Array Comprehension
Iterators
Function Bindings
Metadata
Access Control
Inline constructors
Compiler Usage
Compiler Features
Macros
Standard Library
Haxelib
Target Details

Static Extension »

6.2

Externs

Externs can be used to describe target-specific interaction in a type-safe manner. They are defined like normal classes, except that

- the `class` keyword is preceded by the `extern` keyword,
- methods have no expressions and
- all argument and return types are explicit.

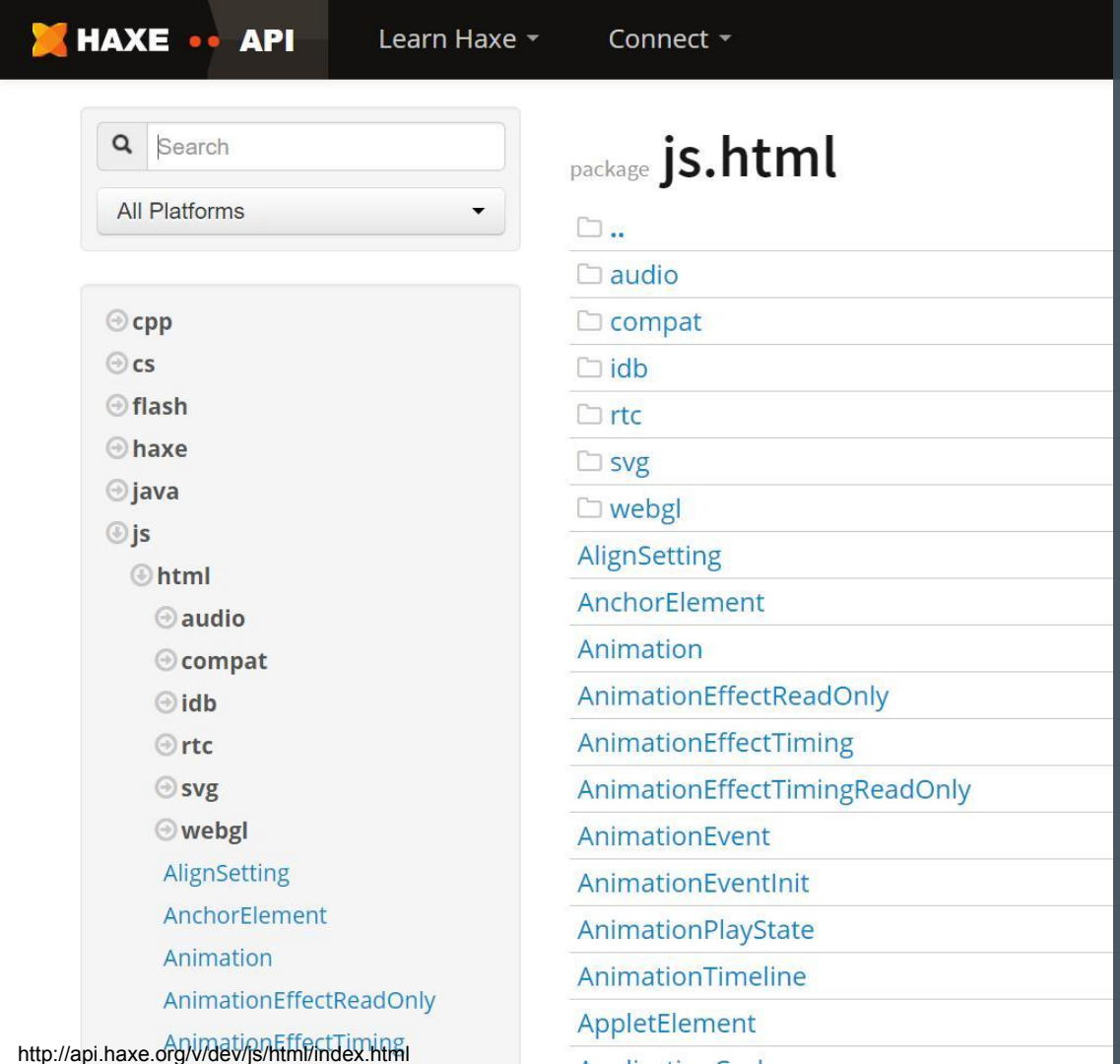
A common example from the Haxe Standard Library is the `Math` class, as an excerpt shows:

```
extern class Math
{
    static var PI(default,null) : Float;
    static function floor(v:Float):Int;
}
```

We see that externs can define both methods and variables (actually, `PI` is declared as a read-only property). Once this information is available to the compiler, it allows field access accordingly and also knows the types:

```
class Main {
  static public function main() {
    var pi = Math.floor(Math.PI);
    $type(pi); // Int
  }
}
```

http://haxe.org/manual/lf-externs.html



Not a New Concept

in our Haxe standard library:

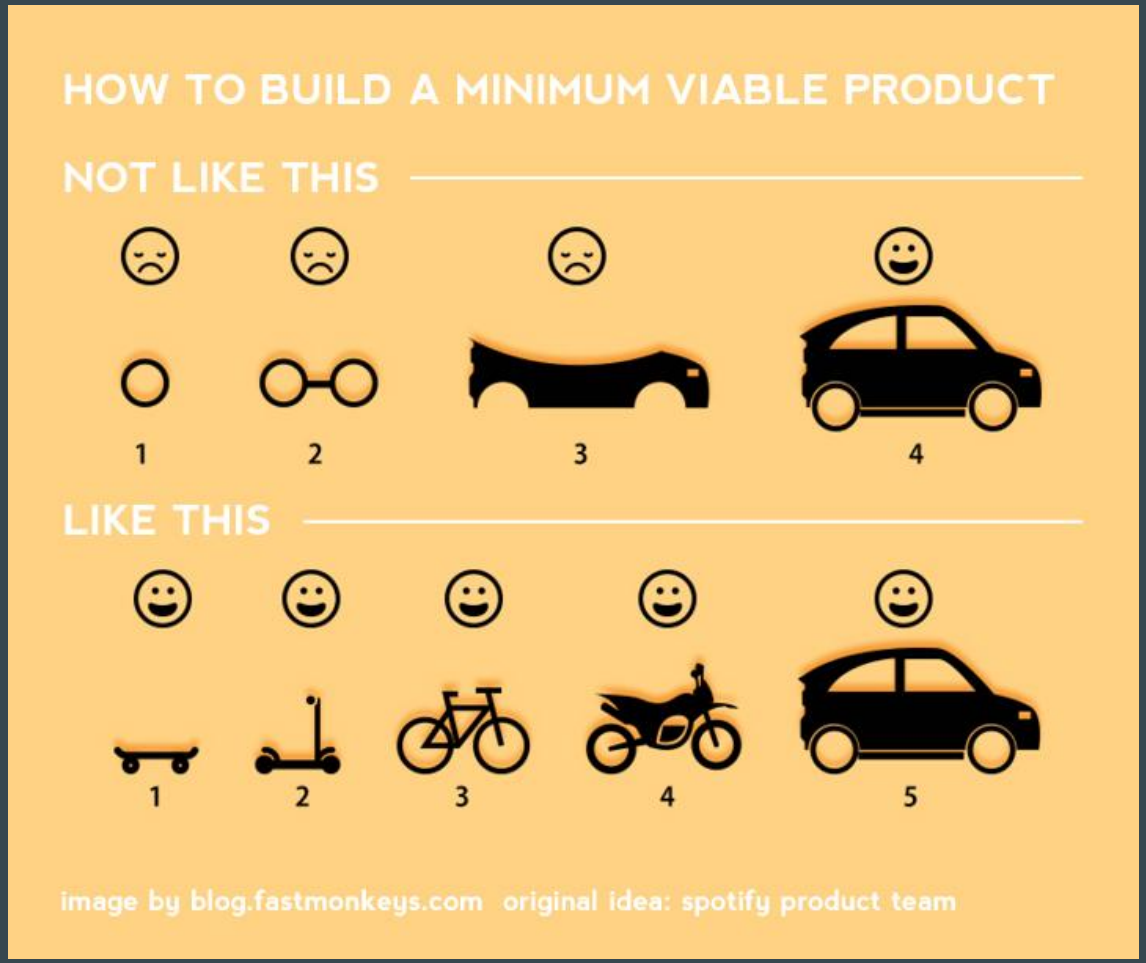
- the Flash API
  - playersglobals.swc
- the js.html package
  - Mozilla's IDL files
- the jQuery extern in js.jquery
  - jQuery's documentation (XML)

maybe in standard library one day:

- pyextern
  https://github.com/andyli/pyextern

Getting type information

- static-typed targets
  - parse source code
  - parse binary
- dynamic-typed targets
  - parse documentation
  - run-time reflection
  - others?



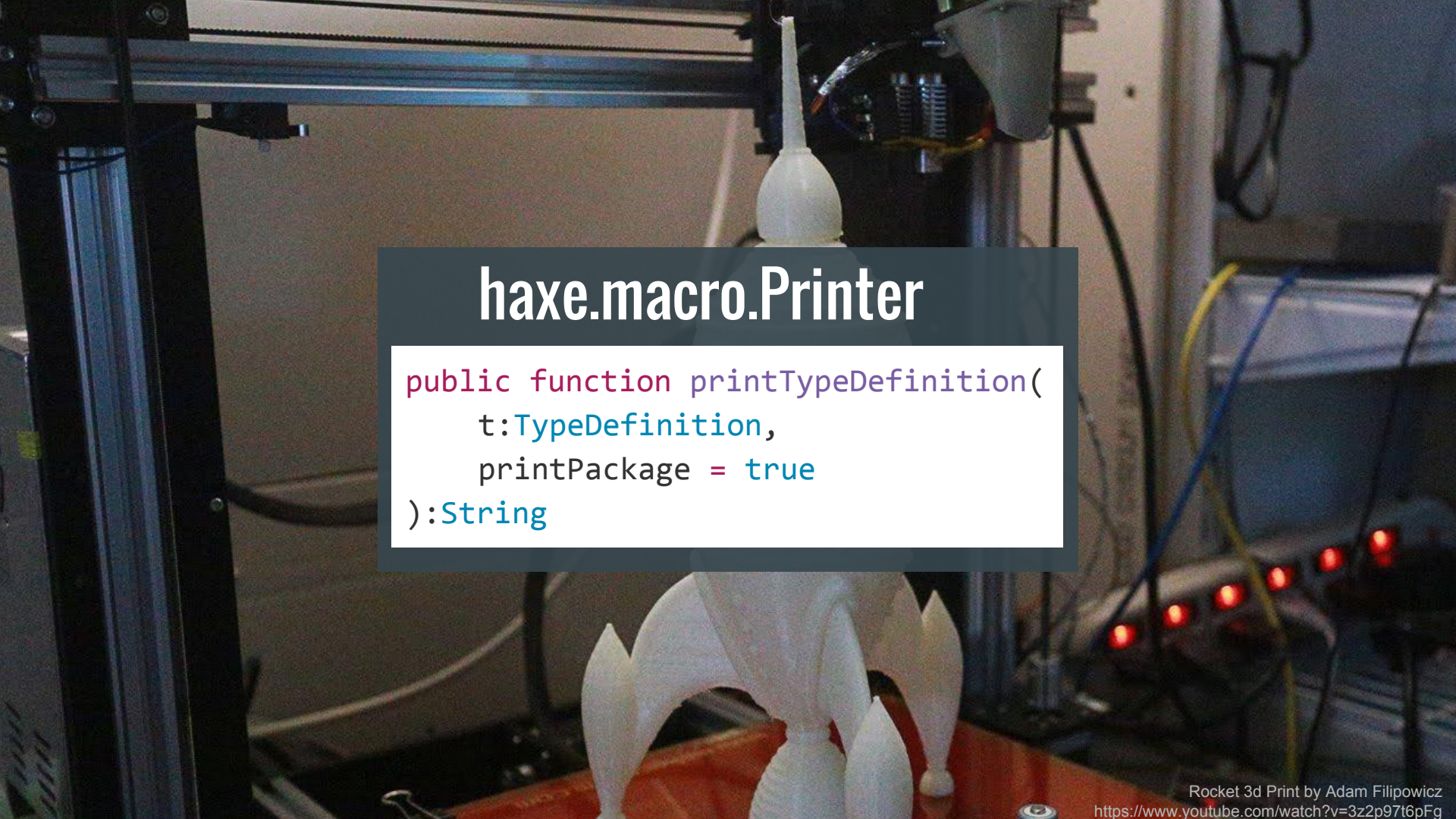
Structure first

- Dynamic for all!

afterwards...

- more accurate types
    - field types and function return types
    - function argument types
- inheritance
- doc strings
- a Haxey wrapper?

https://blog.fastmonkeys.com/2014/06/18/minimum-viable-product-your-ultimate-guide-to-mvp-great-examples/



haxe.macro.Printer

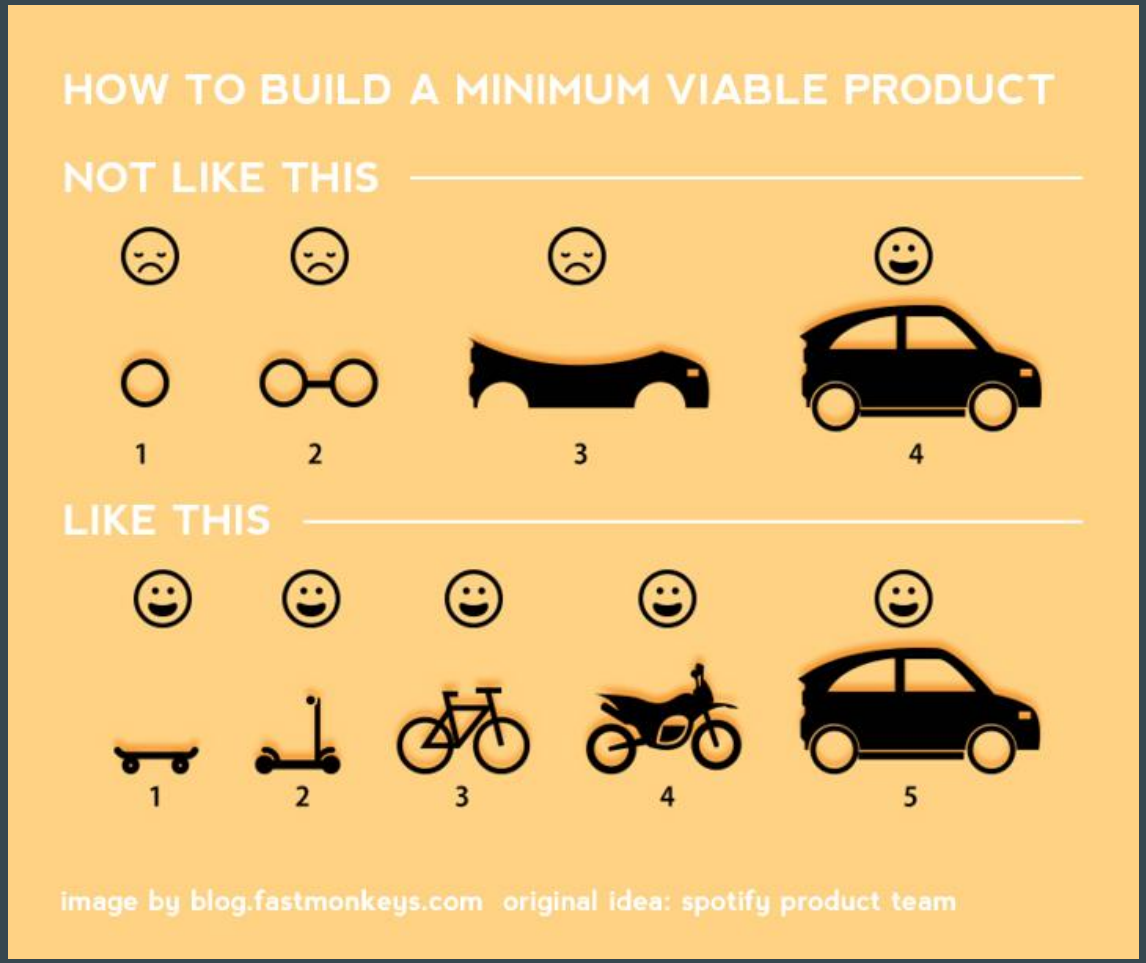
```
public function printTypeDefinition(
    t:TypeDefinition,
    printPackage = true
):String
```

Rocket 3d Print by Adam Filipowicz
https://www.youtube.com/watch?v=3z2p97t6pFg

Test Strategy

- run the Haxe compiler
  - --macro include("packageName")
  - --no-output
- just use it

Testing a lunar mini bike in a low gravity environment. Photo Credit: NASA
http://www.americaspace.com/?p=16953



Type Better

for pyextern

- inspect.getdoc()
- docutils.parsers.rst.Parser()

type mapping

- simple
  - “int”, “string”, “number”...
- complex
  - “string list”, “string or int”
- troublesome
  - no doc
  - incorrect doc

https://blog.fastmonkeys.com/2014/06/18/minimum-viable-product-your-ultimate-guide-to-mvp-great-examples/

typedef Unknown = Dynamic;

- completeness of the extern generator ⇔ number of Unknown
- Maybe embed the input doc string as a type-param? e.g. Unknown<"list of string">

Once upon a star over Southern India. By Scott Kelly
https://twitter.com/StationCDRKelly/status/666042034633883649/

ActionScript® 3.0 Reference for the Adobe® Flash® Platform

Home | Hide Packages and Classes List | Packages | Classes | What's New | Index | Appendixes

flash.display

BlendMode - AS3

Show Filters

Packages Quick Search

Top Level
adobe.utils
air.desktop
air.net
air.update
air.update.events
com.adobe.viewsource
fl.accessibility
fl.containers
fl.controls

Package flash.display

Interfaces

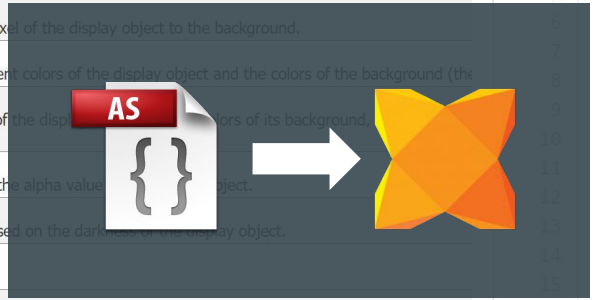
IBitmapDrawable
IGraphicsData
IGraphicsFill
IGraphicsPath
IGraphicsStroke

Classes

ActionScriptVersion
AVLoader
AVM1Movie
Bitmap
BitmapData
BitmapDataChannel
BitmapEncodingColorSpace
BlendMode
CapsStyle

Public Constants

Constant
ADD : String = "add" [static] Adds the values of the constituent colors of the display object to the colors of its background, applying
ALPHA : String = "alpha" [static] Applies the alpha value of each pixel of the display object to the background.
DARKEN : String = "darken" [static] Selects the darker of the constituent colors of the display object and the colors of the background (the
DIFFERENCE : String = "difference" [static] Compares the constituent colors of the display object to the colors of its background and subtracts the darker of the constituent colors from the lighter value.
ERASE : String = "erase" [static] Erases the background based on the alpha value of the display object.
HARDLIGHT : String = "hardlight" [static] Adjusts the color of each pixel based on the darkness of the display object.
INVERT : String = "invert" [static] Inverts the background.
LAYER : String = "layer" [static] Forces the creation of a transparency group for the display object.
LIGHTEN : String = "lighten" [static] Selects the lighter of the constituent colors of the display object and the colors of the background (the
MULTIPLY : String = "multiply" [static] Multiplies the values of the display object constituent colors by the constituent colors of the backgroun resulting in darker colors.
NORMAL : String = "normal"



1 contributor

20 lines (18 sloc) | 211 Bytes

```
1  package flash.display;
2
3  @:fakeEnum(String) extern enum BlendMode {
4          ADD;
5          ALPHA;
6          DARKEN;
7          DIFFERENCE;
8          ERASE;
9          HARDLIGHT;
10         INVERT;
11         LAYER;
12         LIGHTEN;
13         MULTIPLY;
14         NORMAL;
15         OVERLAY;
16         SCREEN;
17         SHADER;
18         SUBTRACT;
19 }
```

Type-Patching

when generating the Flash API externs...

- enum-like values → Haxe enum
- remove problematic or irrelevant fields

Part of the jQuery Extern Generator

```
/**
    Maps a type in api.xml to one or more Haxe ComplexType.
    tag is the xml node where the type is listed.
**/
function toComplexType(type:String, ?tag:Fast, ?inRest = false):Array<ComplexType> {
    var tagName = tag == null ? "" : tag.att.name;
    var entryName = tag == null ? "" : getEntryName(tag.x);
    if (type != null) type = type.trim();

    // ...

    return simple != null ? simple : switch ([entryName, tagName, type]) {
        case ["jQuery.each", "array", "Array"]:
            [macro:Array<Dynamic>];
        case ["jQuery.parseHTML", "jQuery.parseHTML", "Array"]:
            [macro:Array<$element>];
```

conclusion

- why generate externs
  - high API coverage
  - easy to update
  - similar effort to writing externs manually
- techniques
  - get info from doc / reflection
  - haxe.macro.Printer
  - structure first
  - use Dynamic/Unknown
  - type-patching (hard-coding) by pattern matching on the type and a context variable

It's still difficult

problems of the Haxe type system

- array access can only use Int keys
- no operator overloading
- abstract is poor at OOP inheritance
- structural types...
  - have no function overloading
  - have no array access
  - are not callable
- non-nullable type?
- all those C++ ref/poiner/const types?

A failed strut caused the SpaceX rocket explosion
http://www.theverge.com/2015/7/20/9004463/space-x-falcon-9-rocket-explosion-cause-explained

Future work

- complete pyextern
- start jsextern, phpextern, luaextern etc.
- a more powerful and flexible extern type system?
- call for contribution
  - manpower
  - money
  - marketing

GitHub Search GitHub Explore Features Enterprise Blog Sign up Sign in

Contributions Repositories Public activity Follow

Andy Li
andyli

Hong Kong
andy@onthewings.net
http://www.onthewings.net/
Joined on Jul 12, 2009

111 Followers 274 Starred 58 Following

Organizations

Popular repositories

jQueryExternForHaxe — Unleash the full power of jQuery in Haxe. 53
hxOpenFrameworks — Haxe binding to openFrameworks 36
hxLINQ — An implementation of LINQ for Haxe 28
hxColorToolkit — Haxe library for color conversion and color sc...

Repositories contributed to

HaxeFoundation/haxe — Haxe - The Cross-Platform Toolkit 943
caskroom/homebrew-cask — A CLI workflow for the administration of Mac ... 6,791
HaxeFoundation/hxcpp — ...e files for c++ backend for haxe 66
travis-ci/travis-build — .travis.yml => build.sh converter 107
HaxeFoundation/haxelib — The Haxe library manager. This repository co... 57

Keep in touch!
@andy_li

Public contributions

Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May
M W F

Summary of Pull Requests, issues opened, and commits. Learn more. Less More

Contributions in the last year | Longest streak | Current streak