



# Haxe, a statically-typed language that compiles to Python (and more)

Andy Li

**PyCon HK 2015**



To the extent possible under law, Andy Li has waived all copyright and related or neighboring rights to these presentation slides. This work is published from: Hong Kong.



Andy Li  
andyli

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

111 Followers  
274 Starred  
58 Following

Organizations



Contributions

Repositories

Public activity

Follow

Popular repositories

- jQueryExternForHaxe** 53 ★  
Unleash the full power of jQuery in Haxe.
- hxOpenFrameworks** 36 ★  
Haxe binding to openFrameworks
- hxudp** 29 ★  
UDP socket for Haxe/C++
- hxLINO** 28 ★  
An implementation of LINO in Haxe
- hxColorToolkit** 27 ★  
Haxe library for color conversions

Repositories contributed to

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



**Haxe Foundation**

Public contribution



Summary of Pull Requests, issues opened, and commits. [Learn more.](#)

Less More

Contributions in the last year  
**673 total**  
Jun 11, 2014 – Jun 11, 2015

Longest streak  
**11 days**  
April 7 – April 17

Current streak  
**0 days**  
Last contributed a day ago

Contribution activity

Period: 1 week

1 commit

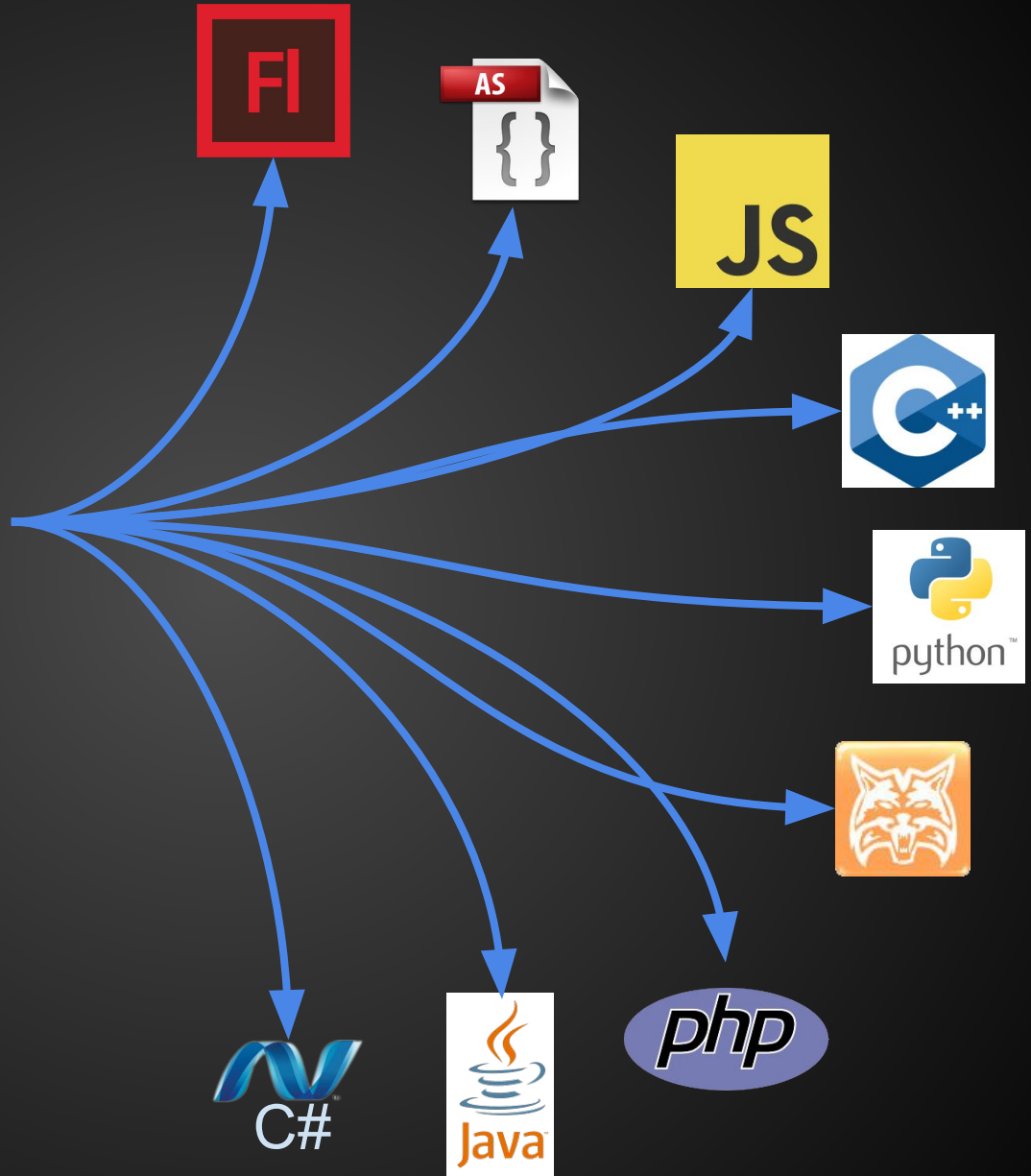
Pushed 1 commit to caskroom/homebrew-cask Jun 10

# Overview

- What is Haxe
- Haxe vs Python
- The Haxe Python target



Haxe



# Haxe features - JS/Java-like syntax

source code: [pycon/hk/HelloWorld.hx](https://pycon.hk>HelloWorld.hx)

```
package pycon.hk;

class HelloWorld {
  static function main() {
    var target = Sys.args()[0];
    var speaker = {
      first: "Andy",
      last: "Li"
    }
    trace('${speaker.first} ${speaker.last}: Hello, $target!');
  }
}
```

output: [HelloWorld.py](#)

PS: showing only the output of the pycon\_hk\_HelloWorld class

```
class pycon_hk_HelloWorld:
    __slots__ = ()

    @staticmethod
    def main():
        target = python_internal_ArrayImpl._get(Sys.args(), 0)
        speaker_first = "Andy"
        speaker_last = "Li"
        print(str(((((((("'" + ("null" if speaker_first is None else
speaker_first))) + " ") + ("null" if speaker_last is None else speaker_last))
+ ": Hello, ") + ("null" if target is None else target)) + "!"))))
```

# Haxe features - static typing

source code: [Typing.hx](#) [output: [Typing.py](#)]

```
class Typing {
  static function main():Void {
    var i = 123;           // same as var i:Int = 123;
    $type(i);             // Int
    // i = "123";         // error: String should be Int

    var floats = [1.0, 1.1, 1.2];
    $type(floats);        // Array<Float>
    $type(floats[0]);     // Float
    floats[0] = i;
    trace(floats);        // [ 123, 1.1, 1.2 ]
    // floats[0] = "string"; // error: String should be Float
  }
}
```

# Haxe features - OOP

source code: [OOP.hx](#) [output: [OOP.py](#)]

```
class Point {
    public var x:Float;
    public var y:Float;
    public function new(x:Float, y:Float):Void {
        this.x = x;
        this.y = y;
    }
    public function offset(dx:Float = 0, dy:Float = 0):Point {
        return new Point(x + dx, y + dy);
    }
}

class Opts {
    static function main():Void {
        var p = new Point(0, 0);
        var p2 = p.offset(1, 2);
        trace(p2.x); //1
    }
}
```

# Haxe features - functional programming

source code: [Functional.hx](#) [output: [Functional.py](#)]

```
using Lambda; // static extension
import haxe.ds.*;

class Functional {
  static function main() {
    // Array comprehension
    var evens:Array<Float> = [for (i in 0...15) if (i % 2 == 0) i];
    trace(evens); // [ 0, 2, 4, 6, 8, 10, 12, 14 ]

    // functional goodies from `using Lambda`
    var maxMultipleOf4 = evens
      .filter(function(i) return i % 4 == 0)
      .fold(function(i, a) return Math.max(i, a), evens[0]);
    trace(maxMultipleOf4); // 12

    // enum (GADT) and pattern matching
    function getAnyHigher(floats:Array<Float>, v:Float):Option<Float> {
      for (f in floats)
        if (f > v)
          return Some(f);
      return None;
    }
    switch (getAnyHigher(evens, 5)) {
      case Some(value):
        // string interpolation (not really FP, but still nice)
        trace('In evens, $value is higher than 5');
      case None:
        trace("No value in evens is higher than 5");
    }
  }
}
```



# What is Haxe?

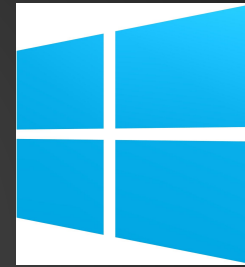
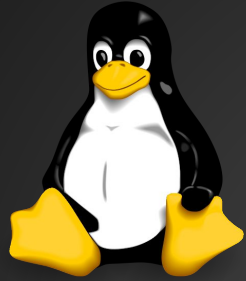
## True :D

- Haxe is free and open source.
- Haxe is a programming language that compiles to 9 different targets.
- Haxe provides a set of small yet enough data structures and APIs.
- Haxe allows accessing native APIs / libraries.
- If no target-specific things are used, Haxe code “should” automatically work on all targets.

## False...

- ~~Haxe is a young new language.~~ (appeared in 2005)
- ~~Haxe is a magical program that converts existing app to different platforms.~~
- ~~Haxe allows us to use APIs or libraries from arbitrary targets (e.g. use jQuery in the PHP target).~~
- ~~Haxe produced program cannot be faster / better than program written in the target language.~~

**Why do we want  
cross-platform?**



iOS





編號 : 36-10-0012038 SA

日期 :  
2010/10/13

名稱 : 全職銷售員兼平面設計及  
文員

行業 : 零售  
業

工作時間 : 上午9時至下午8時, 每週工作6天, 輪休

薪酬 : 每月\$6,000, 有獎賞, 佣金 及酌情  
性花紅

地區 : 元朗

資歷 : 中五程度; 3年經驗; 良好粵語; 良好普通話; 一般英語; 良好上海話; 懂讀寫中文; 懂讀寫英文; 懂Illustrator; 懂Corel Draw; 具有基本電腦操作知識; 具有一般互聯網知識; 懂MS Word; 懂MS PowerPoint; 懂MS Excel; 懂MS Outlook; 懂Freehand; 懂C/C++, Visual C++; 懂ASP; 懂Java Applet; 懂JavaScript; 懂TCP/IP; 懂Frontpage; 懂Flash; 懂Fireworks; 懂Dreamweaver; 懂MS Access; 懂MS SQL Server; 懂Photoshop; 懂AutoCad; 需熟用AI等列出的全部電腦程式。及對參茸行業有所經驗。有上進心及不怕辛苦。有需要時需加班。

職責 : 負責店舖銷售及理貨。需要獨自兼顧全公司之宣傳品設計及。獨自文件處理及輸入輸出。電腦簡單維修。

求職者可致電  與  先生聯絡。 



## Programmer / Web Developer

Create Alert

Follow

2 days ago from [jobs.gov.hk](http://jobs.gov.hk)

空缺：1 個

編號：32-15-0012751 CM 兼職 日期：28/05/2015

職位：程式員/網站開發員

公司/僱主名稱：APP MOCHA LIMITED

地區：沙田 行業：通訊業

職責：Commitment to finish the job (Past jobs in designing program of websites, PhoneGap, android java, ios objective c: Immediate available preferred)

資歷：專上教育：文憑/證書課程；良好粵語；良好普通話；一般英語；懂讀寫中文；懂讀寫英文；懂 JavaScript；懂HTML/XML；懂Flash；懂Dreamweaver；懂C/C++；Visual C++；Knowledge of HTML, CSS, Javascript; Knowledge of PHP/MySQL

待遇：每小時\$32.50，星期一至五：上午10時至下午6時，每週工作3天至5天，每天工作6-8小時



# I.T.公司最低工資\$32.5 請「萬能」程式員

Apply Now

★ Save Job

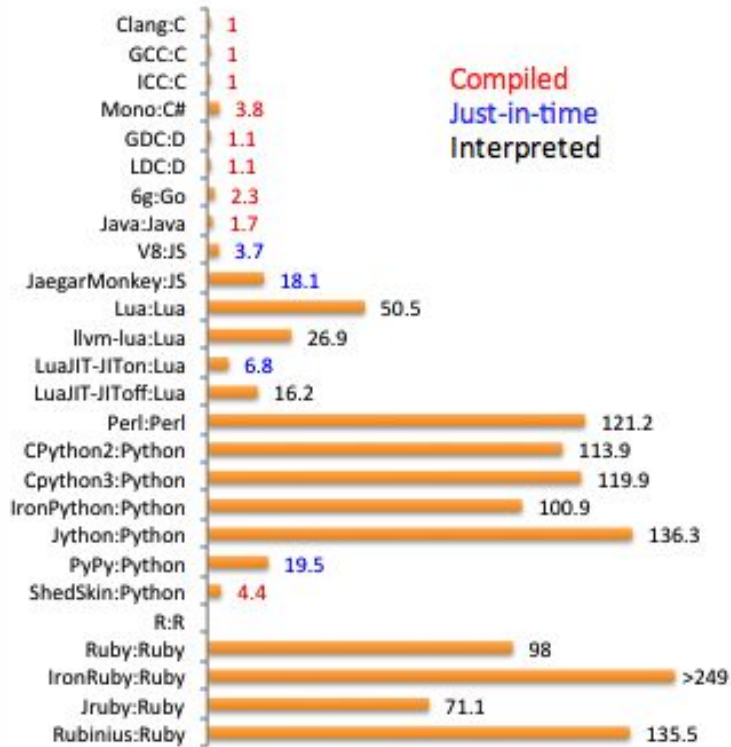
# code reuse



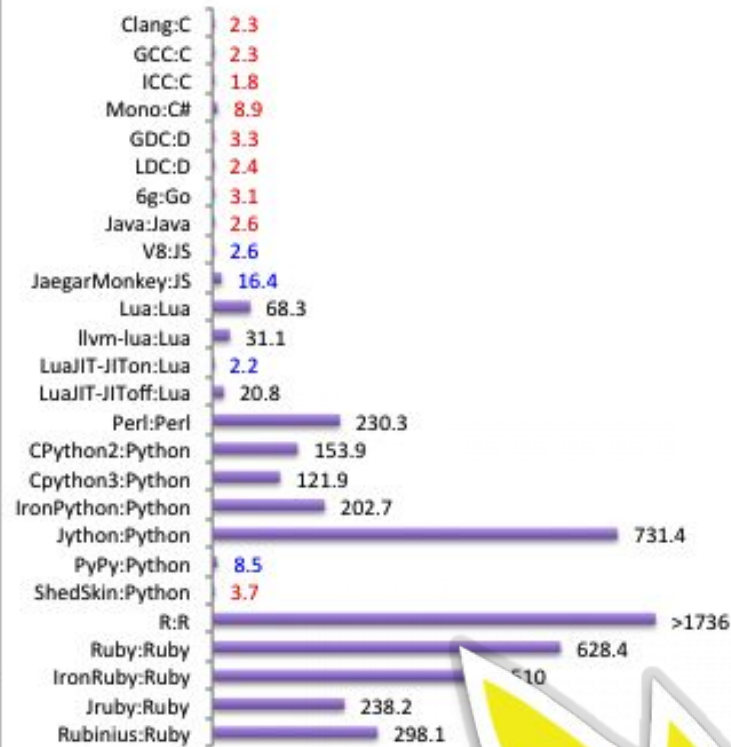
platforms / languages come and gone...



## Sudoku solving (CPU sec)



## Matrix multiplication (CPU sec)



test all languages!  
with your own app!



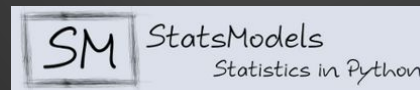
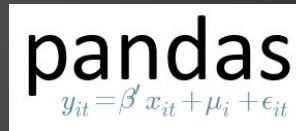


# Why would one compile **Haxe** to **Python**?

\$ diff haxe python

# the Python libs are great

- scientific stuffs
- data analysis
- machine learning
- web frameworks
- etc.



# static typing vs dynamic typing

- static typing catches errors earlier
- type annotation is one kind of doc
- static typing leads to better code generation -> better performance

# functional vs imperative

- **Haxe** is designed with **both** OOP & FP in mind. **Python** is mostly imperative / OOP.
- **Haxe** features not available in **Python**:
  - expression-oriented syntax + macros
  - GADT + pattern matching

# the **Haxe** Python target

status and how-to

# Current status

- the youngest one,  
added in Haxe 3.2.0 (released in 2015)
- main authors on Github:  
[@frabbit](#), [@Simn](#), and [@nadako](#)
- supports Python 3 only
- unit tests on the Haxe language specification are  
passing on all Windows/Mac/Linux for the Python target
  - there are [>6000 assertions](#)

# using Python lib in Haxe

- use the `untyped` keyword
  - tell the compiler to shut up:
    - undeclared variables are there,
    - types are correct although it looks like it's not...
- use [python.Syntax](#)
  - `python.Syntax.pythonCode("#whatever");`
- use [externs](#)
  - tell Haxe about the structure and types
  - Automatically generated externs:  
<https://github.com/andyli/pyextern>

# the `untyped` keyword

source code: [Untyped.hx](#)

```
class Untyped {  
    static function main():Void {  
        var l = untyped list("abc");  
        trace(l); // ['a', 'b', 'c']  
    }  
}
```

output: [Untyped.py](#)

```
# Generated by Haxe  
  
class Untyped:  
    __slots__ = ()  
  
    @staticmethod  
    def main():  
        l = list("abc")  
        print(str(l))  
  
Untyped.main()
```



# python.Syntax

source code: [PySyntax.hx](#)

```
import python.Syntax;

class PySyntax {
    static function main():Void {
        var string = Syntax.pythonCode('"abc" # type : str');
        trace(string);
    }
}
```

output: [PySyntax.py](#)

```
# Generated by Haxe

class PySyntax:
    __slots__ = ()

    @staticmethod
    def main():
        string = "abc" # type : str
        print(str(string))

PySyntax.main()
```

# externs

source code: [Extern.hx](#) [output: [Extern.py](#)]

```
import python.Tuple;

@:pythonImport("inspect") extern class Inspect {
    static public function getdoc(object:Dynamic):String;

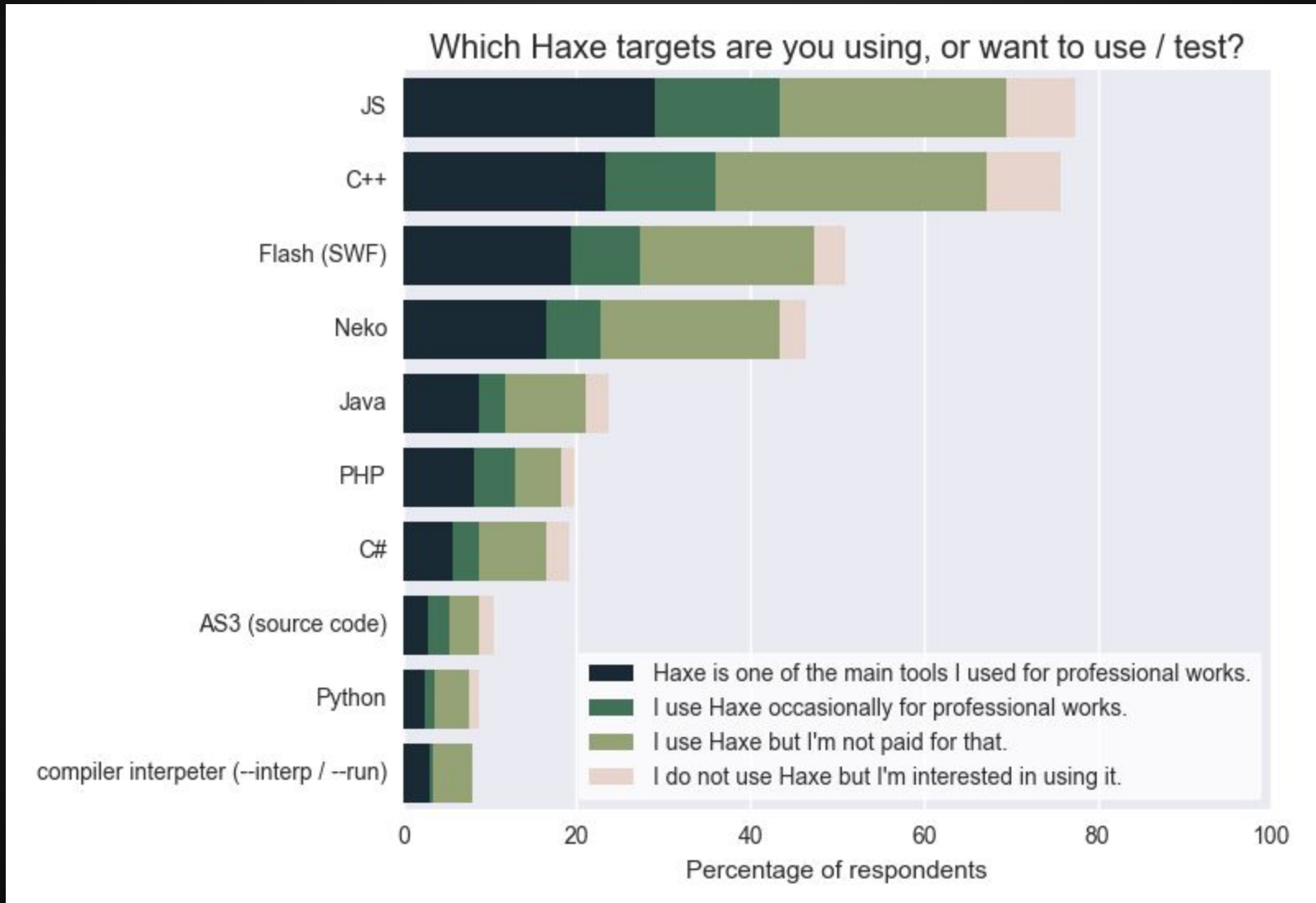
    static public function getmembers
        (object:Dynamic, ?predicate:have.Constraints.Function)
        :Array<Tuple2<String,Dynamic>>;

    static public function signature
        (obj:Dynamic, ?follow_wrapped:Bool = true)
        :Dynamic;
}

class Extern {
    static function main():Void {
        var getdocdoc = Inspect.getdoc(Inspect.getdoc);
        trace(getdocdoc); // "Get the documentation string for an object..."
    }
}
```

# Example: Data analysis of a Haxe usage survey

source: <https://github.com/andyli/haxe-usage-survey>



# Automatic generating externs

- <https://github.com/andyli/pyextern>
  - make use of `inspect` and `docutils`
  - `python3 Main.py numpy,scipy[,...] out`
  - cover all classes and functions
  - use `Dynamic` for most things and “guess” type from docstring

# Function call with named arguments

A typical function in a Python lib:

```
pandas.read_csv(filepath_or_buffer, sep=',', dialect=None, compression='infer', doublequote=True, escapechar=None, quotechar='"', quoting=0, skipinitialspace=False, lineterminator=None, header='infer', index_col=None, names=None, prefix=None, skiprows=None, skipfooter=None, skip_footer=0, na_values=None, true_values=None, false_values=None, delimiter=None, converters=None, dtype=None, usecols=None, engine=None, delim_whitespace=False, as_reccarray=False, na_filter=True, compact_ints=False, use_unsigned=False, low_memory=True, buffer_lines=None, warn_bad_lines=True, error_bad_lines=True, keep_default_na=True, thousands=None, comment=None, decimal='.', parse_dates=False, keep_date_col=False, dayfirst=False, date_parser=None, memory_map=False, float_precision=None, nrows=None, iterator=False, chunksize=None, verbose=False, encoding=None, squeeze=False, mangle_dupe_cols=True, tupleize_cols=False, infer_datetime_format=False, skip_blank_lines=True)
```

54 arguments!



# Function call with named arguments

source code: [Kw.hx](#) [output: [Kw.py](#)]

```
import python.KwArgs;

class Kw {
    static function test(a:Float, b:Float, c:Float):Void {
        trace('$a, $b, $c');
    }
    static function main():Void {
        test(1.1, 1.2, 1.3);

        var kw:KwArgs<Dynamic> = {a: 2.1, b: 2.2, c: 2.3};
        (untyped test)(kw);

        (untyped test)(({a: 3.1, b: 3.2, c: 3.3}:KwArgs<Dynamic>));
    }
}
```

- verbose and ugly...

# Function call with named arguments

let's add sugar :)

```
using PyHelpers;
...
test.call(c => 4.3, a => 4.1, b => 4.2);
...
var data = Pandas.read_csv.call(
    dataPath,
    sep => "\t",
    parse_dates => [0],
    header => 0
);
```

- <https://github.com/andyli/haxe-usage-survey/blob/master/src/PyHelpers.hx>
  - [macros](#) + [`using` static extension](#)

# Future work (my personal wish list)

- Output type annotations  
(python 3.5 / cython)
- Better extern generation, integrate [python/typedhed](#)
- Release Haxe libs to pypi  
(like <https://github.com/paulfitz/daff>)
- Re-implement python api in haxe?
- Python 2 support?



# Made in Haxe

- Games
  - [OpenFL showcase](#)
  - [Flambe showcase](#)
- [Prezi's article on using Haxe](#)
- [TiVo's talk on using haxe in the enterprise](#)  
(set-top box and mobile devices)
- [Net Wars](#) (Favourite Website Awards (FWA) featured)
- [LID Shirtshop](#) (interactive online shop)
- [Slickrock.io](#) (easily embeddable chatroom)
- [Verb](#) (a Haxe-compiled JS lib for creating and manipulating 3D NURBS)

# The super active Haxe community

- Haxe weekly roundup: [haxe.io](http://haxe.io)
- Twitter: [@haxelang](https://twitter.com/haxelang), [#haxe](https://twitter.com/haxe)
- Facebook: <https://www.facebook.com/haxe.org/>
- the World-Wide-Haxe conference (WWX)
  - <http://www.silexlabs.org/wrapping-up-wwx2015/>
- StackOverflow: [\[haxe\] tag](https://stackoverflow.com/questions/tagged/haxe)
- Github: [HaxeFoundation/haxe](https://github.com/HaxeFoundation/haxe)

# Interested in Haxe?

- Read => <http://haxe.org/>
- Try => <http://try.haxe.org/>
- Get =>
  - with an installer: <http://haxe.org/download/>, or
  - with a package manager:
    - `brew install haxe` on Mac
    - `choco install haxe` on Windows
    - [use a PPA](#) (ppa:haxe/releases) on Ubuntu
  - IDEs: <http://haxe.org/documentation/introduction/editors-and-ides.html>



Andy Li  
andyli

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

111  
Followers

274  
Starred

58  
Following

Organizations



Contributions

Repositories

Public activity

Follow

Popular repositories

- jQueryExternForHaxe** 53 ★  
Unleash the full power of jQuery in Haxe.
- hxOpenFrameworks** 36 ★  
Haxe binding to openFrameworks
- hxudp** 29 ★  
UDP socket for Haxe/C++
- hxLING** 28 ★  
An implementation of LING in Haxe.
- hxColor** 27 ★  
Haxe library for color conversion and color sc...

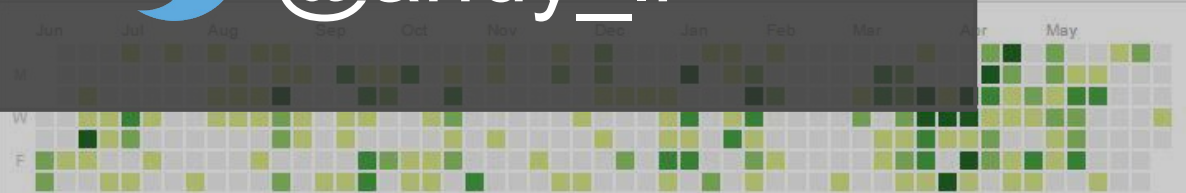
Repositories contributed to

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

**Get in touch!**

**@andy\_li**

Public cont



Summary of Pull Requests, issues opened, and commits. [Learn more.](#)

Less      More

Contributions in the last year  
**673 total**  
Jun 11, 2014 – Jun 11, 2015

Longest streak  
**11 days**  
April 7 – April 17

Current streak  
**0 days**  
Last contributed a day ago

Contribution activity

Period: 1 week

1 commit

Pushed 1 commit to caskroom/homebrew-cask Jun 10