

Introduction to



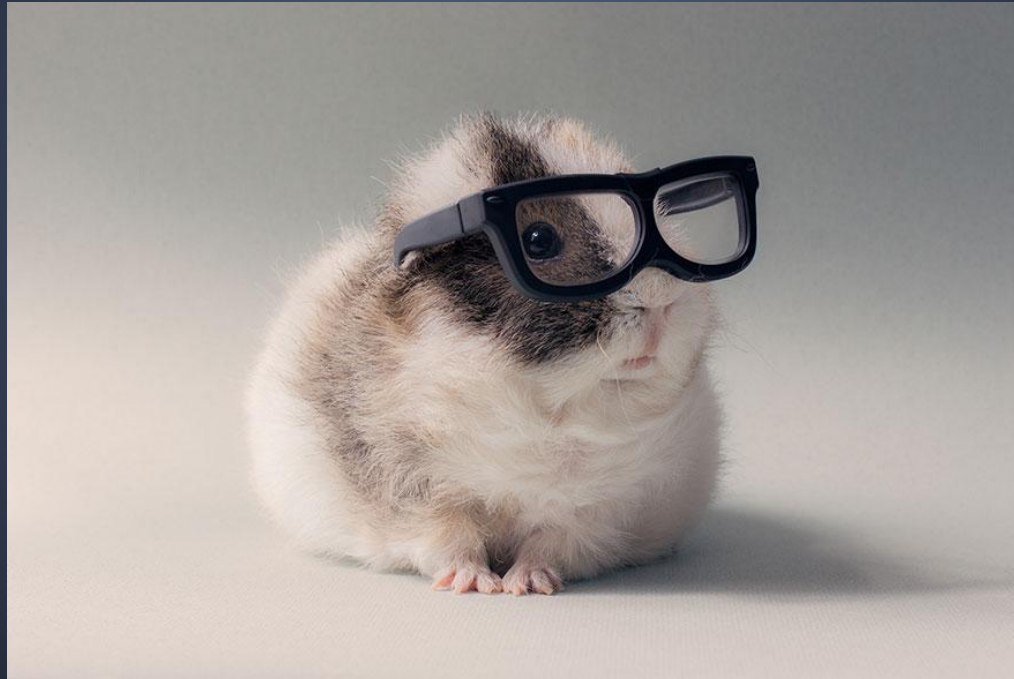
HAXE

by Kevin Purcell

Why Haxe?



Why Haxe?



What is the Haxe Toolkit?

- Cross-platform development toolkit
- Runs on Windows, Mac & Linux
- Haxe language
- Standard library
- Cross-compiler
- haxelib library manager
- Supplied with the Neko VM

A Brief History of Haxe

- Created by Nicolas Cannasse
- Dev started at Motion Twin in October 2005
- First beta released February 2006
- Initial support for AVM and Neko VM targets
- New language targets created by contributors
- Haxe Foundation formed November 2012
- World Wide Haxe Conference (WWX)
held annually in Paris by Silex Labs

A Brief History of Haxe

- 2006: haXe 1.0 (JavaScript)
- 2007: haXe 1.12 (ActionScript 3)
- 2008: haXe 2.0 (PHP)
- 2009: haXe 2.04 (C++)
- 2012: Haxe 2.09
- 2012: Haxe 2.10 (Java & C#)
- 2013: Haxe 3.0
- 2015: Haxe 3.2 (Python)

Haxe Targets



Haxe Targets

Flash	Bytecode	Games, Desktop, Mobile
Neko	Bytecode	Web, CLI
JavaScript	Source code	Games, Web, Desktop
ActionScript 3	Source code	Games, Mobile
PHP	Source code	Web
C++	Source code	Games, Desktop, Mobile, CLI
Java	Source code	Desktop, Mobile, CLI
C#	Source code	Desktop, Mobile
Python	Source code	Web, Desktop, CLI

The Haxe Language

One language to rule them all...



The Haxe Language

HelloWorld.hx

```
class HelloWorld {  
    static public function main() {  
        trace("Hello World!");  
    }  
}
```

```
$ haxe -main HelloWorld -js HelloWorld.js
```

The Haxe Language

HelloWorld.js

```
(function () { "use strict";  
var HelloWorld = function() { };  
HelloWorld.main = function() {  
    console.log("Hello World!");  
};  
HelloWorld.main();  
})();
```

The Haxe Language

- Strictly typed *

```
var doingTalk:Bool = true;
var minutes:Int = 30;
var subjects:Array<String> = [
    "The Haxe Language",
    "The Haxe Cross-Compiler",
    "haxelib"
];

var version:Map<String,String> = [
    "haxe" => "3.1.3",
    "neko" => "2.0.0"
];
```

The Haxe Language

- Strictly typed *
- Type inference

```
var doingTalk = true;
var minutes = 30;
var subjects = [
    "The Haxe Language",
    "The Haxe Cross-Compiler",
    "haxelib"
];

var version = [
    "haxe" => "3.1.3",
    "neko" => "2.0.0"
];
```

The Haxe Language

- Strictly typed *
- Type inference
- Object Oriented

```
class GuineaPig {  
    var name:String;  
    var age:Int;  
  
    public function new(name, age) {  
        this.name = name;  
        this.age = age;  
    }  
}  
  
var pig1 = new GuineaPig("Maisie", 1);  
var pig2 = new GuineaPig("Tilly", 2);
```

The Haxe Language

- Strictly typed *
- Type inference
- Object Oriented
- String interpolation

```
var x = 1;
var y = 2;

trace('$x + $y = ${x + y}');
// Outputs: 1 + 2 = 3

var lang = "haxe";

trace('Welcome to
    ${lang.toUpperCase()}!');
// Outputs: Welcome to HAXE!
```

The Haxe Language

- Strictly typed *
- Type inference
- Object Oriented
- String interpolation
- Iterators

```
var guineaPigs = [  
    "Maisie",  
    "Tilly",  
];  
  
for (guineaPig in guineaPigs) {  
    trace(guineaPig);  
}  
  
for (i in 0...guineaPigs.length) {  
    trace(guineaPigs[i]);  
}
```


The Haxe Language

- Strictly typed *
- Type inference
- Object Oriented
- String interpolation
- Iterators
- Array comprehension

```
var numbers = [for (i in 0...10) i];
trace(numbers);
// Outputs 0,1,2,3,4,5,6,7,8,9

var oddNumbers = [
    for (i in numbers)
        if (i % 2 == 1) i
];
trace(oddNumbers);
// Outputs: 1,3,5,7,9
```

The Haxe Language

- Strictly typed *
- Type inference
- Object Oriented
- String interpolation
- Iterators
- Array comprehension
- Pattern matching

```
var guineaPig = {
  name: "Basil",
  age: 5
};
var message = switch (guineaPig) {
  case {name: "Darcy", age: _}:
    "It's Darcy!";
  case {name: n, age: 5}:
    'Hello $n!';
  case _: "Who is this?";
}
trace(message); // Hello Basil!
```

The Haxe Language

- Strictly typed *
- Type inference
- Object Oriented
- String interpolation
- Iterators
- Array comprehension
- Pattern matching
- Conditional compilation

```
var target:String;
#if js
    target = "JavaScript";
#elseif cpp
    target = "C++";
#else
    target = "Unknown";
#end
trace(target);
```

The Haxe Language

- Strictly typed *
- Type inference
- Object Oriented
- String interpolation
- Iterators
- Array comprehension
- Pattern matching
- Conditional compilation
- Externs

```
extern class Math {  
    static var PI(default, null):Float;  
    static function floor(v:Float):Int;  
}  
  
var pi = Math.floor(Math.PI);  
$type(pi); // Int
```

The Standard Library

- Data structures
- Math
- Regular expressions
- JSON & XML
- HTTP
- ZIP
- Cryptography
- Unit testing
- File I/O
- File system
- MySQL & SQLite
- Native processes
- Target-specific APIs
- Networking (C++)
- DOM & Cookies (JS)
- Sessions (PHP)

The Haxe Cross-Compiler

- Compiles to target language or bytecode
- Single front-end parses Haxe to AST
- Multiple back-ends translate AST to specific targets
- Written in OCaml
- Dead code elimination
- Code completion
- Function inlining
- Resource embedding
- Metadata

Macros?

Make Haxe...



Macros!

...do MORE!



haxelib

- Library / package manager for Haxe
- Command line interface
- Central repository at <http://lib.haxe.org>
- A library consists of Haxe code
- Support for multiple versions of a library
- GitHub support
- Easy to submit a new library

Who Uses Haxe?



Prezi



nickelodeon

Disney



Coca-Cola



Hasbro



TOYOTA





massive

stencyl™




Developing in Haxe

- Text Editor
- FlashDevelop IDE
- IntelliJ IDEA Plugin
- Sublime Text Plugin
- Vaxe vim Bundle
- HIDE / Haxe Studio
- Atom Plugin
- Flash CC

Haxe for Games

- OpenFL
-  HaxeFlixel
-  HaxePUNK
- Flambe
- luxe
- Kha
- Heaps
- awe6
- Nape
- Box2D
- Away3D
- Haxor
- Pixi.js & Phaser
- Unity3D - HUGS, unihx

Haxe for Web

- JavaScript
-  *jQuery*
-  - haxe-js-kit, hxnodejs
-  ANGULARJS - angular.haxe
- PHP
- mod_neko (Neko on Apache)
- UFront MVC Framework (PHP/Neko & JS)
- Python

Haxe for Everything Else

- Desktop & Mobile Applications
 - HaxeUI
 - StablexUI
 - Waxe (wxWidgets wrapper)
 - node-webkit (NW.js)
- Command Line
 - mcli (Mini CLI)
 - MLib (MassiveLib)
 - nekotools

Useful Haxe Resources

- Haxe Website - <http://haxe.org>
- Manual - <http://haxe.org/manual>
- API Reference - <http://api.haxe.org>
- haxelib - <http://lib.haxe.org>
- Try Haxe - <http://try.haxe.org>
- Learn Haxe in Y Minutes -
<http://learnxinyminutes.com/docs/haxe>
- Haxe Roundup - <http://haxe.io>

Any Questions?



@grayhaze