



Typed JS with Haxe

aka WTF is Haxe?

or

Transpiring to JS with Haxe

or

"I don't like Javascript"

Brief History...



Circa 2005, created by Nicolas Cannasse (<https://twitter.com/ncannasse>)
Almost 11 years of history!

Initially the successor of the MTASC*[0] compiler
Heavily focused on Flash/Game Development at first
Has grown to be an incredible language+toolkit for cross platform development

Libs and frameworks for t **webdev** are growing everyday...

Awesome conference!
<http://wwx.silexlabs.org/2016/>

*[0] <http://www.mtasc.org/>

Strongly typed

```
var foo:String = "Hey"  
foo = 1 //Int should be String
```

Statically typed:

Type errors at compile-time
you get syntax checking for free, too

Type inference:

```
var foo = "Hi"  
$type(foo)  
=> String
```

Dynamic vs Static anyone? :)



Types, types, types.... and expressions.

In Haxe, everything boils down to a type or instance of a type or an expression.

Common Haxe Types:

Class, Enum, Structure, Function, Abstract...

Abstracts

Powerful compile-time feature allowing for powerful abstractions.

Only present at compile-time.

<http://try.haxe.org/#78EaD>

Enums & pattern matching

<http://haxe.org/manual/lf-pattern-matching-introduction.html>

<http://try.haxe.org/#70477>

More features!

Blazingly fast compilation (you often won't notice it actually compile),
compilation cache server, “using” keyword, interfaces, GADT, macros, typedefs.
Check the manual or have a look at `tink_lang`.

Also, besides JS, cross-compile to the following outputs:



Why Haxe for the web?

Because I can.

Because I don't like Javascript, **the language**

Static typing FTW, less runtime errors, if any

Shared code on the back/front

Shared code across tools (C#, Java, C++, Python, NekoVM)

Powerful preprocessor with macros

Powerful meta-programming (AST-level) with macros

More (sane) structure, better than CoffeeScript and TypeScript

Great for large codebases and teams!

What do I need to start?

Install / Compile Haxe! — available for all OSes.. <http://haxe.org/>
Plugins available for major editors and IDEs.

Get the manual: <http://haxe.org/manual>

Frontend: Compile to js, load the js, done.
Backend: compile to js, run with nodejs.

More?

Externs. Externs provide Typing information to Haxe.

No Externs? Write as you go, or use ``untyped``: <http://try.haxe.org/#d864A>

Advanced:

More control? **Macros**^[0]. Even more? Custom generator! (<https://github.com/explorin/es6-coretypes>)

Backend? Can compile to PHP, js (node), C#, Java, C++, Python and NekoVM bytecode.

Want to be pragmatic? Compile to js on Node or use rails-api on the backend. A lot of value to gain from Haxe in the frontend.

Sadly no Ruby target, yet...

^[0] <http://haxe.org/manual/macro.html> and <https://github.com/macronauts/haxe-macro-workshop-2015>

What's bad about Haxe?

Lack of documentation. **Getting better.**

Lack of docs = barrier of entry is higher than other more popular languages and stacks = smaller community = people might consider Haxe niche = FUD :(

[Cross-platform] eco-system still relatively small, comparing to Ruby, for example.
Come and contribute!

However, Haxe can leverage the eco-system of any of its target languages!

Resources:

<https://haxe.io/>
<http://haxe.org>
<https://github.com/HaxeFoundation>
<http://philippe.elsass.me/2014/11/haxe-working-with-javascript-libraries/>

Great libs and frameworks

<https://github.com/haxetink>
<http://thx-lib.org/>
<https://github.com/ufront/ufront>
<https://github.com/clemos/haxe-js-kit>
<https://github.com/benmerckx/monsoon>
<https://github.com/massiveinteractive/haxe-react>
<https://github.com/frabbit/angular.haxe>
<http://haxeui.org/>
<http://snowkit.org/>
<http://www.openfl.org/>
<https://github.com/ciscoheat/buddy>

...

Come and join the community!

Learn, contribute, spread the word!

#haxe on freenode

HaxeLang on GoogleGroups

#haxe on Twitter

WWX2016 in Paris, next May.

Me: **@fullofcaffeine** on Twitter.