



TypeScript
coding JavaScript
without the pain

@Sander_Mak

Luminis Technologies

WHAT'S WRONG WITH JAVASCRIPT?

- ❑ Dynamic typing
- ❑ Lack of modularity
- ❑ Verbose patterns (IIFE)



Ts

WHAT'S WRONG WITH JAVASCRIPT?

- ❑ Dynamic typing
- ❑ Lack of modularity
- ❑ Verbose patterns (IIFE)

In short:

JavaScript development scales badly



Ts

WHAT'S GOOD ABOUT JAVASCRIPT?

- ❑ It's everywhere
- ❑ Huge amount of libraries
- ❑ Flexible



Ts

WISHLIST



- ❑ Scalable HTML5 clientside development
- ❑ Modular development
- ❑ Easily learnable for Java developers
- ❑ Non-invasive (existing libs, browser support)
- ❑ Long-term vision
- ❑ Clean JS output (exit strategy)

Ts

WISHLIST



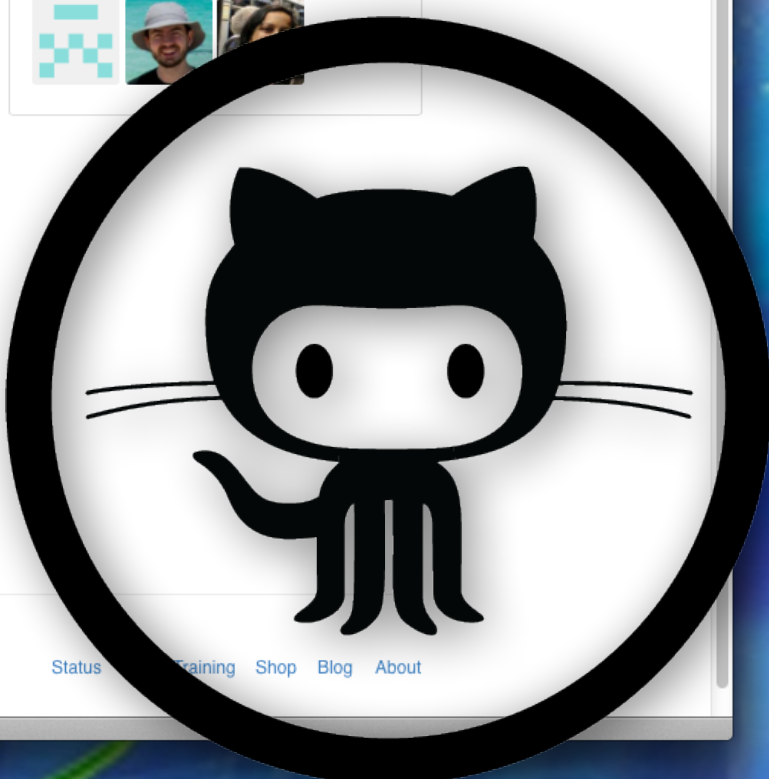
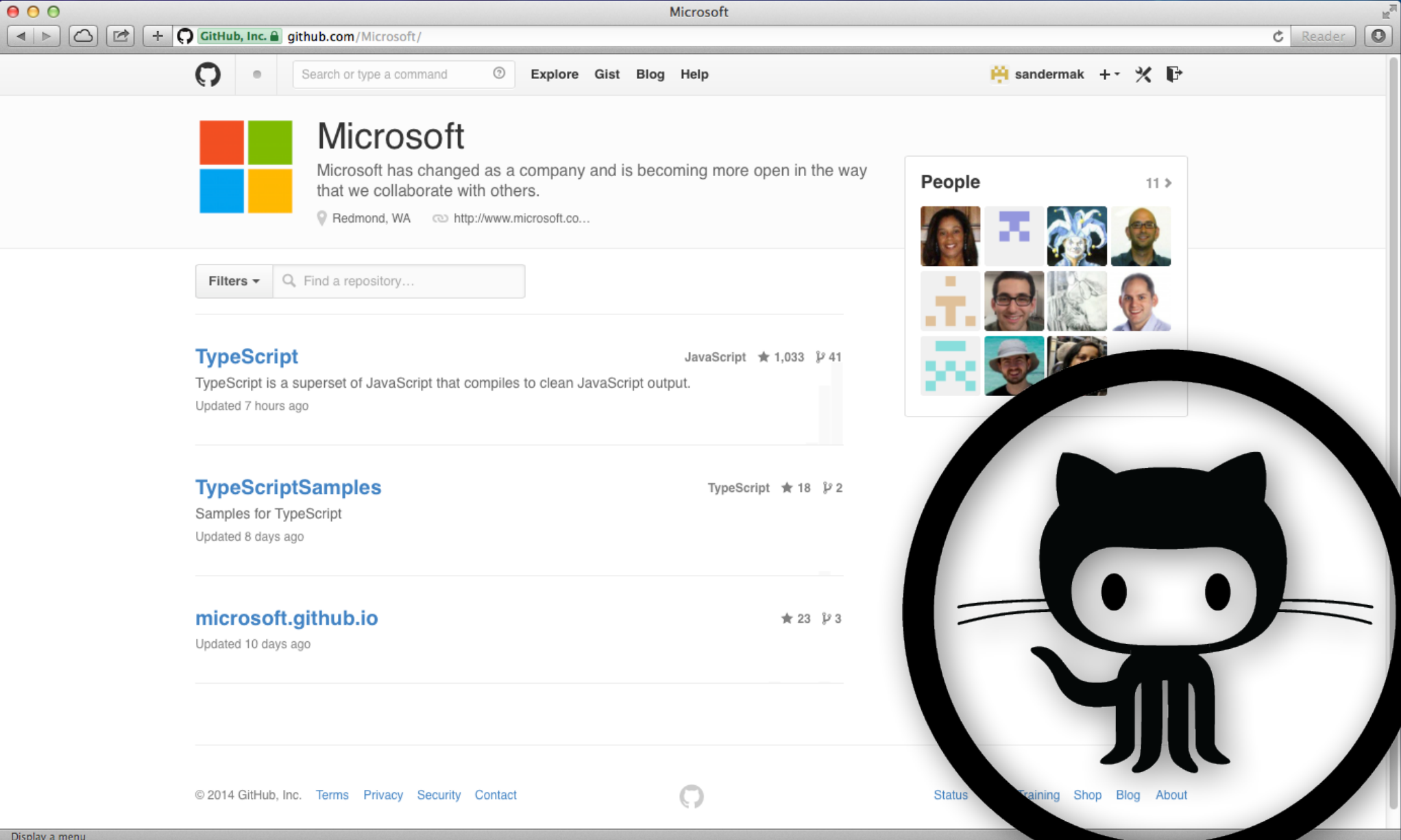
- ✓ Scalable HTML5 clientside development
- ✓ Modular development
- ✓ Easily learnable for Java developers
- ✓ Non-invasive (existing libs, browser support)
- ✓ Long-term vision
- ✓ Clean JS output (exit strategy)

Ts



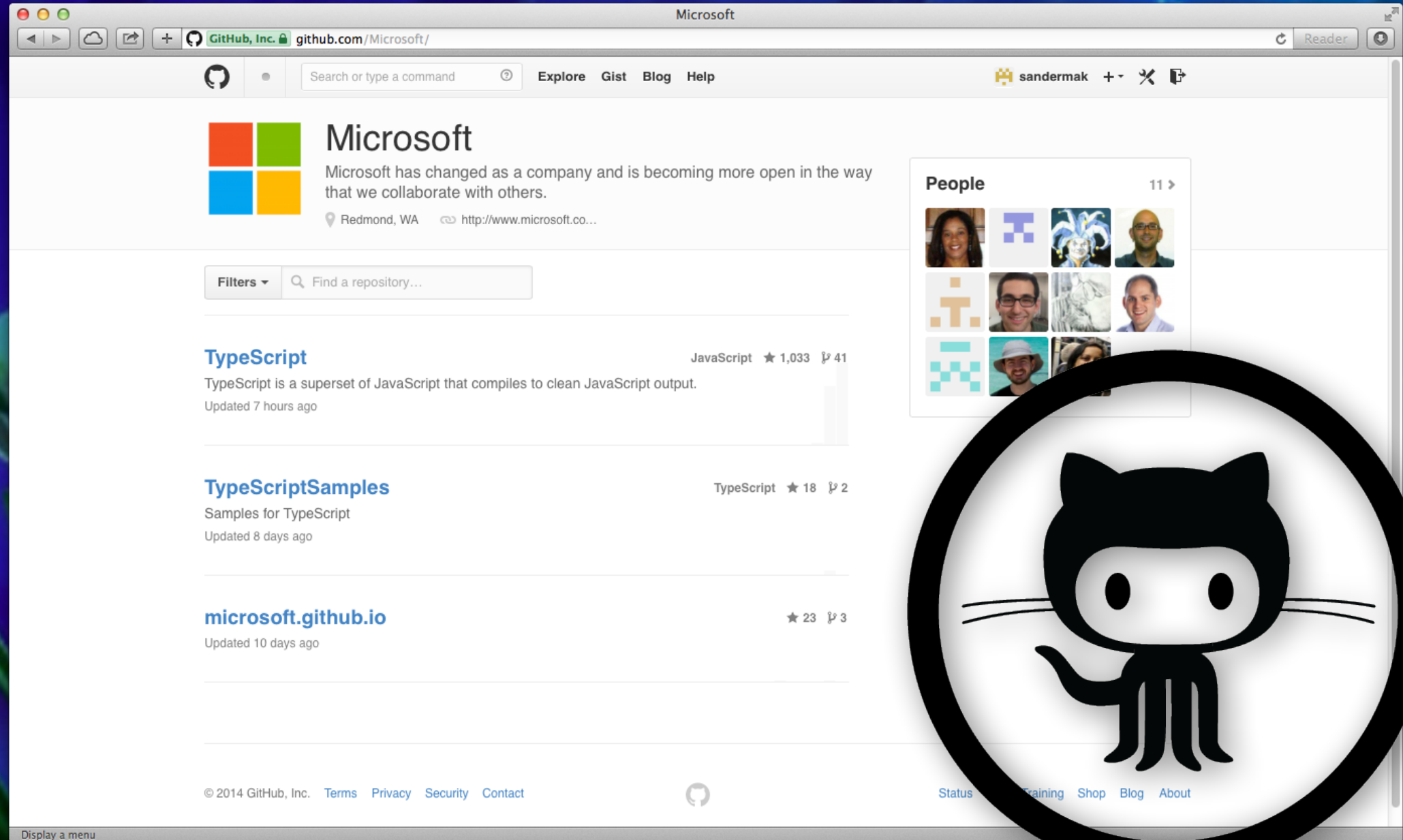
Microsoft


Apache 2.0 licensed






Apache 2.0 licensed



TYPESCRIPT

- ❑ Superset of JavaScript
- ❑ Optionally typed
- ❑ Compiles to ES3/ES5/ES2015
- ❑ No special runtime
- ❑ 1.0 in April 2014, current: 1.6



Ts

TYPESCRIPT

- ❑ Superset of JavaScript
- ❑ Optionally typed
- ❑ Compiles to ES3/ES5/ES2015
- ❑ No special runtime
- ❑ 1.0 in April 2014, current: 1.6

In short:

Lightweight productivity booster



Ts

GETTING STARTED

```
$ npm install -g typescript  
$ mv mycode.js mycode.ts  
$ tsc mycode.ts
```



Ts

GETTING STARTED

```
$ npm install -g typescript  
$ mv mycode.js mycode.ts  
$ tsc mycode.ts
```



May even find problems in existing JS!

Ts

OPTIONAL TYPES

Type annotations

```
> var a = 123  
> a.trim()
```

JS

TypeError: undefined is
not a function

runtime

```
> var a: string = 123  
> a.trim()
```

TS

Cannot convert 'number'
to 'string'.

compile-time

TS

OPTIONAL TYPES

Type annotations

```
> var a = 123  
> a.trim()
```

JS

TypeError: undefined is not a function

```
> var a: string = 123  
> a.trim()
```

TS

Cannot convert 'number' to 'string'.

Type inference

```
> var a = 123  
> a.trim()
```

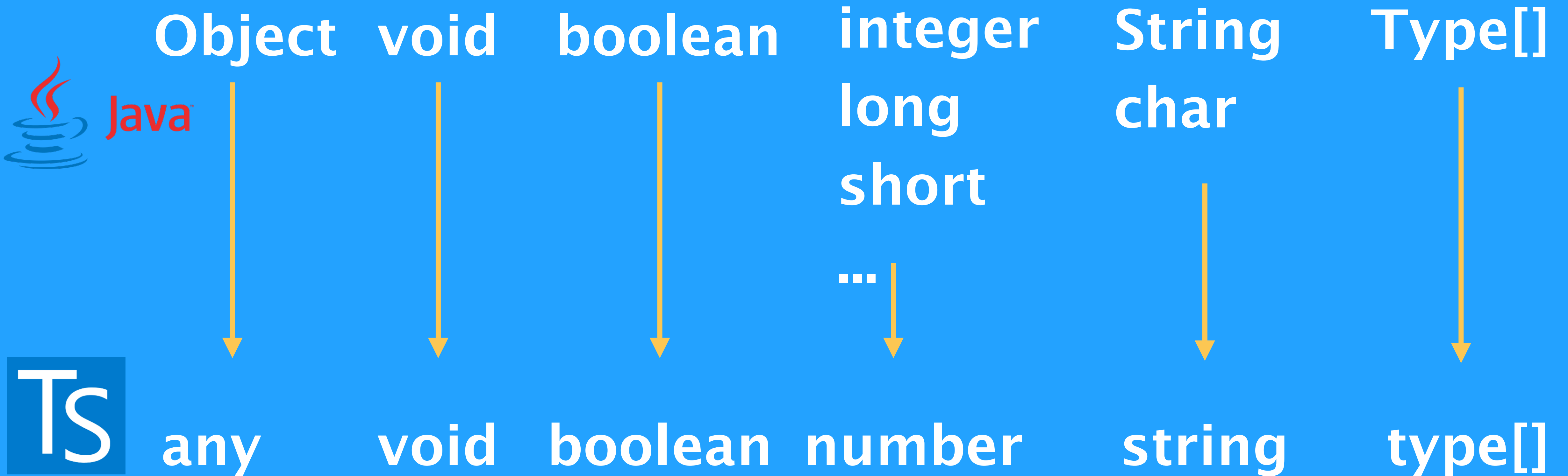
The property 'trim' does not exist on value of type 'number'.

Types disappear at runtime

TS



OPTIONAL TYPES



TS

EXPRESSIVE TYPES

Generics myList: List<T>

Tuple-types point: [number, number] = [1, 2]

Union types animal: Cat | Dog

Intersection order: Order & Shippable
types

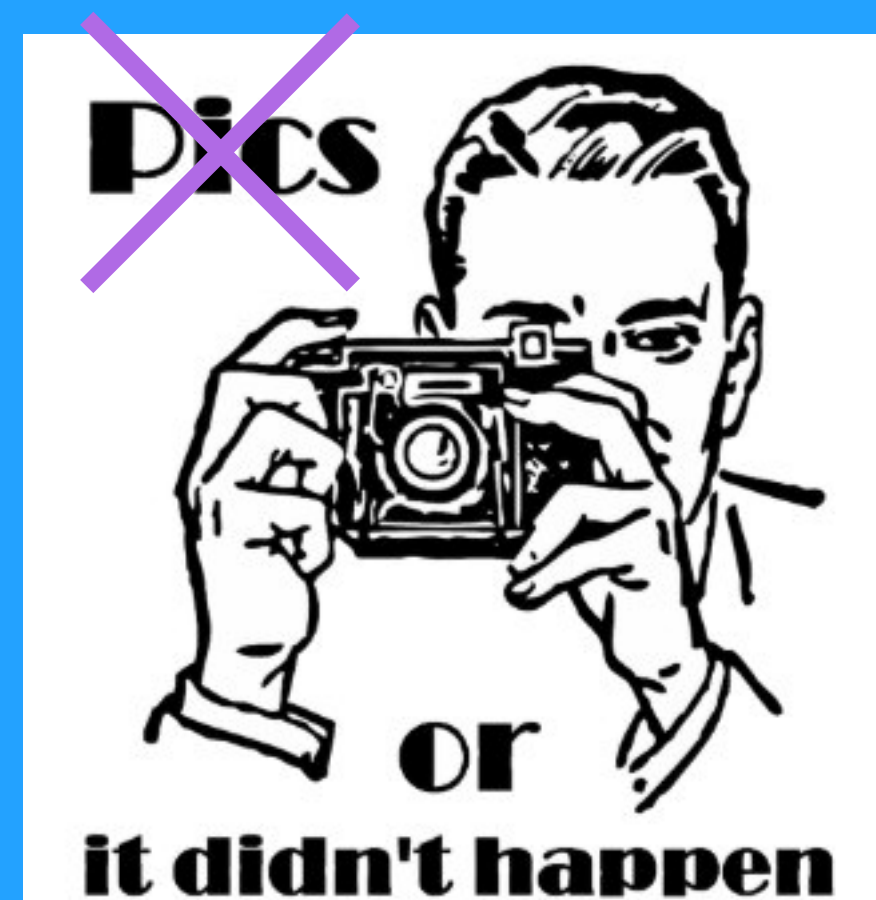
Aliasing type Animal = Cat | Dog

TS



DEMO: OPTIONAL TYPES

code



Code: <http://bit.ly/tscode>

Ts

INTERFACES

Use them to describe data returned in REST calls

```
$.getJSON('user/123').then((user: User) => {  
  showProfile(user.details)  
})
```



Ts



GOING ALL THE WAY

Force explicit typing with noImplicitAny

```
var ambiguousType;
```

```
ambiguousType = 1
```

```
ambiguousType = "text"
```

noimplicitany.ts

```
$ tsc --noImplicitAny noimplicitany.ts
```



Ts

GOING ALL THE WAY

Force explicit typing with noImplicitAny

```
var ambiguousType;  
  
ambiguousType = 1  
ambiguousType = "text"
```

noimplicitany.ts

```
$ tsc --noImplicitAny noimplicitany.ts
```

```
error TS7005: Variable 'ambiguousType' implicitly  
has an 'any' type.
```



Ts

TYPESCRIPT CLASSES

- ❑ Can implement interfaces
- ❑ Inheritance
- ❑ Instance methods/members
- ❑ Static methods/members
- ❑ Single constructor
- ❑ Default/optional parameters
- ❑ ES6 class syntax



DEMO: TYPESCRIPT CLASSES

code



Code: <http://bit.ly/tscode>

Ts

TYPE DEFINITIONS

How to integrate existing JS code?

- ❑ Ambient declarations
- ❑ Any-type :(
- ❑ Type definitions
- ❑ lib.d.ts
- ❑ Separate compilation:
`tsc --declaration file.ts`



Ts

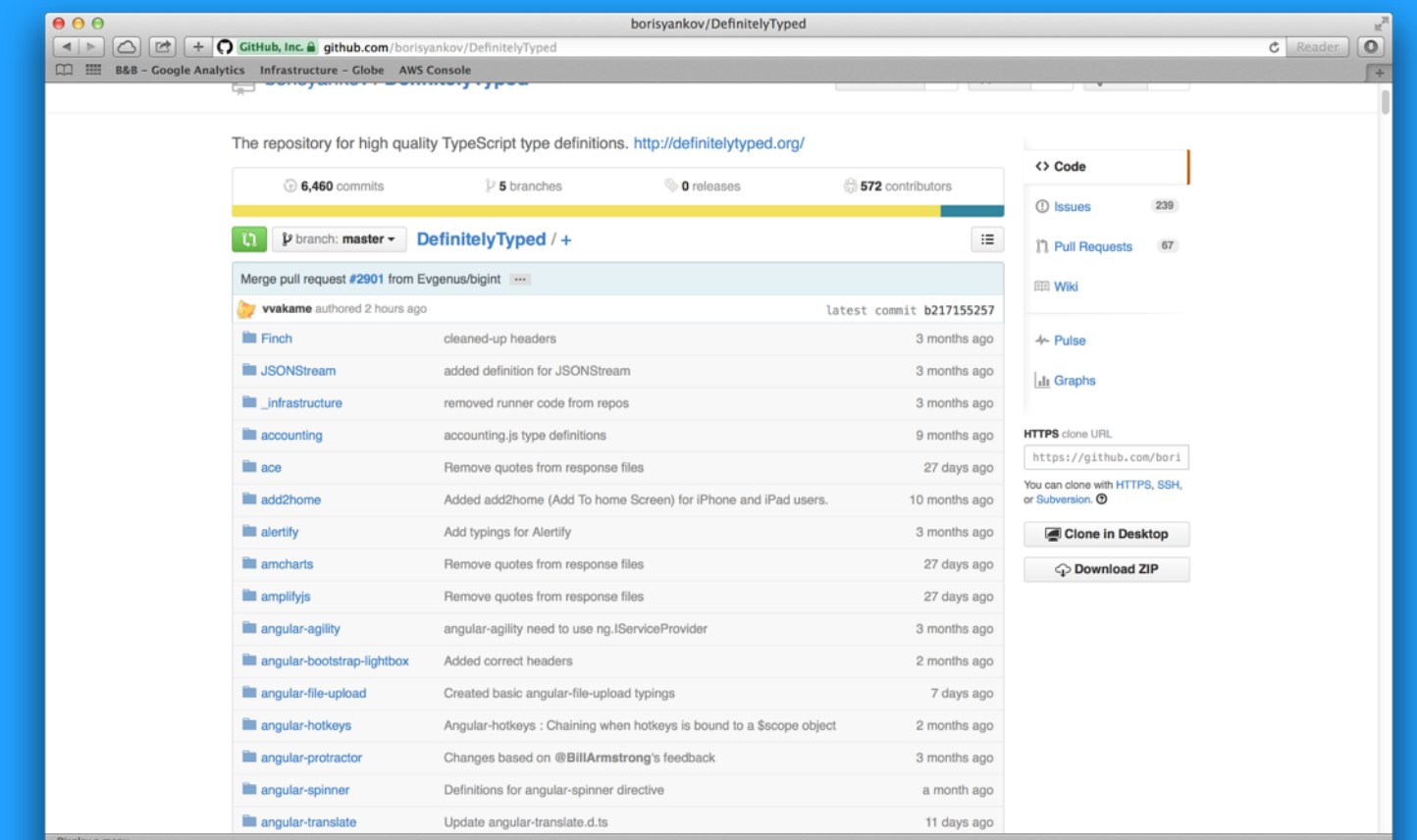
TYPE DEFINITIONS

How to integrate existing JS code?

- ❑ Ambient declarations
- ❑ Any-type :(
- ❑ Type definitions
- ❑ lib.d.ts
- ❑ Separate compilation:
tsc --declaration file.ts

DefinitelyTyped.org

Community provided .d.ts files for popular JS libs



Ts

MODULES

Internal:

Namespacing

External:

E.g. CommonJS/AMD
Combine with loader
(RequireJS)

TypeScript 1.5: ES6 modules

Ts

A dark blue silhouette of a city skyline with various building shapes, located in the bottom right corner of the slide.

DEMO: PUTTING IT ALL TOGETHER



Ts

DEMO: PUTTING IT ALL TOGETHER

Go left:

TypeScript
jQuery
IntelliJ
Live-coding

Go right:

TypeScript
Angular2-alpha
Atom
Walkthrough

Ts

A dark blue silhouette of a city skyline with various building shapes, located at the bottom right of the slide.

OUR STACK



+



ANGULARJS

+



=



Awesome

TS

CONCLUSION

TypeScript allows for gradual adoption

Internal modules
Classes/Interfaces
Some typing

External modules
Type defs
More typing

Generics
Type defs
-noImplicitAny

Ts

CONCLUSION

Some downsides:

- ❑ Still need to know some JS quirks
- ❑ Introduces compile times into frontend workflow



Ts

CONCLUSION

- ❑ High value, low cost improvement over JavaScript
- ❑ Safer and more modular
- ❑ Solid path to ES2015 and further



Ts

RESOURCES

Code: <http://bit.ly/tscode>

Learn: <http://typescript.tips>

@Sander_Mak

Luminis Technologies

Ts

A dark blue silhouette of a city skyline with various building shapes, located in the bottom right corner of the slide.