



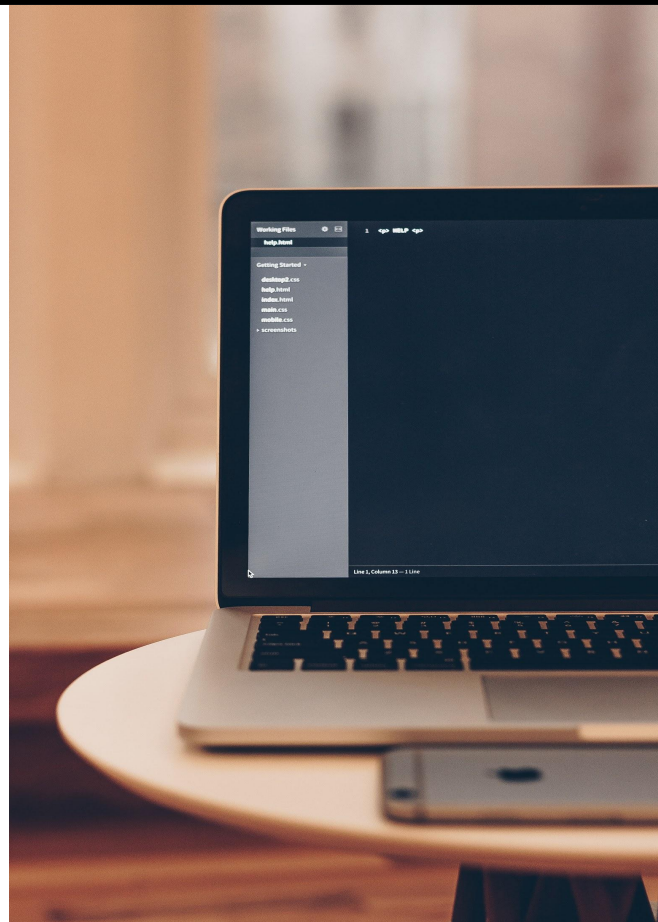
# **Strong Strings**

**why string is weak and  
what to do about it**

# Hello!

**I'm Jason Killian**

I'm a frontend-focused  
software engineer at  
Palantir.



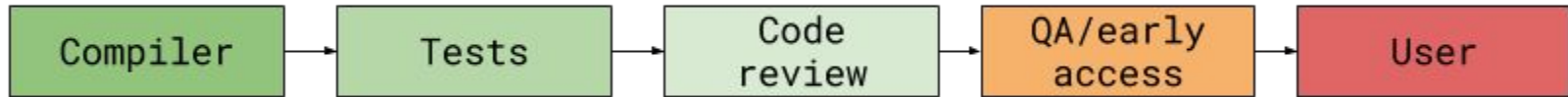
# **Philosophical Question Time!**



**1.**

**Why do we even  
bother with  
TypeScript?**

# Move your bugs to the left



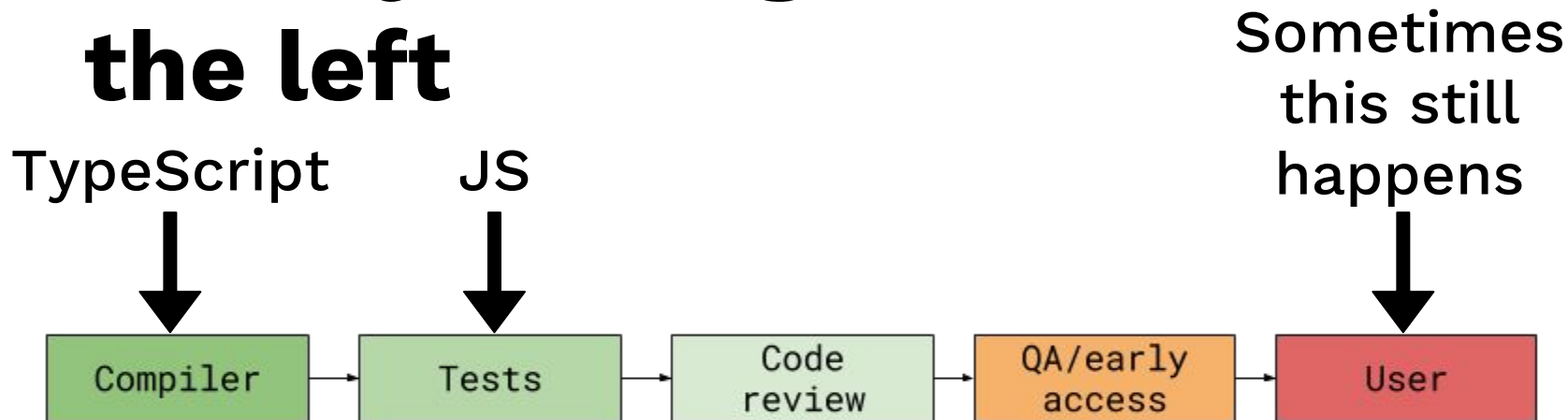
# Move your bugs to the left

TypeScript

JS



# Move your bugs to the left



# Game Time!

The good, the bad, and the ugly





```
const avgLength = (strings: string[]) => strings.length > 0  
  ? strings.reduce((a, b) => a + b.length, 0) / strings.length  
  : 0;
```

```
const avgLength = (strings: string[]) => strings.length > 0  
  ? strings.reduce((a, b) => a + b.length, 0) / strings.length  
  : 0;
```



**Good!**

11



12

```
const ssnLastFourDigits = (ssn: string) =>  
  ssn.slice(-4);
```

```
const ssnLastFourDigits = (ssn: string) =>  
  ssn.slice(-4);
```



**Bad!**

```
const getCommentText = (commentId: string) =>  
  | commentService.getComment(commentId).then(c => c.text);
```

```
const getCommentText = (commentId: string) =>  
  commentService.getComment(commentId).then(c => c.text);
```



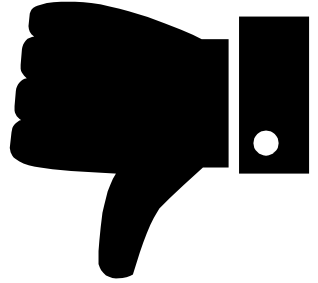
**Bad!**



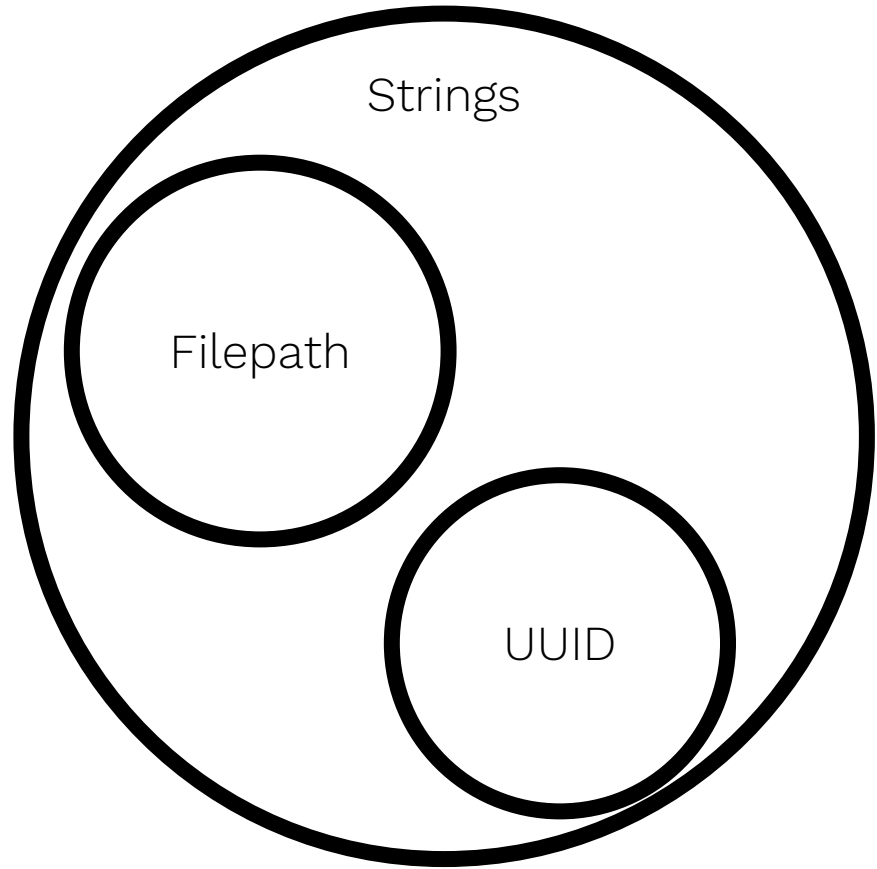
**2.**

**What makes  
string bad?**

**string isn't a  
specific enough  
type in most  
cases**



**When is a  
string  
more  
than a  
string?**



# When is a string more than a string?



## Okay uses of string

- A user-entered comment
- Text to display in a warning dialog
- A label for a button

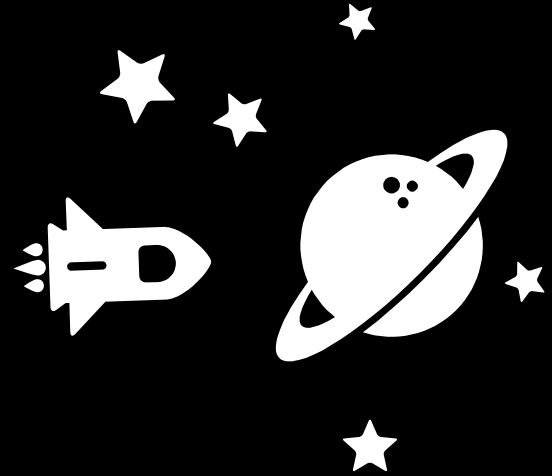
## Bad uses of string

- An address
- A UUID
- An SSN
- A filepath

**3.**

**How do we fix  
string?**

# Structural vs. Nominal



# Type alias to the rescue!

Try it out.

```
type FilePath = string;  
const readFile = (file: FilePath) => { /* ... */ };  
readFile("random non-filepath string"); // compiles :(
```

## Type alias to the rescue!

Try it out.



```
type FilePath = string;  
const readFile = (file: FilePath) => { /* ... */ };  
readFile("random non-filepath string"); // compiles :(
```

**Type alias to the  
rescue (or not).**



**Bad!**

Try it out.

# Wrapped types?

Try it out.

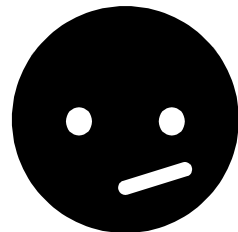
```
type FilePath = {  
  filePath: string,  
};  
  
const readFile = (path: FilePath) => {  
  path.filePath // do something with it  
  // etc.  
};  
  
const someFilePathObj = { filePath: "/path/to/file" };  
readFile(someFilePathObj);
```

## Wrapped types?

Try it out.

```
type FilePath = {  
  filePath: string,  
};  
  
const readFile = (path: FilePath) => {  
  path.filePath // do something with it  
  // etc.  
};  
  
const someFilePathObj = { filePath: "/path/to/file" };  
readFile(someFilePathObj);
```

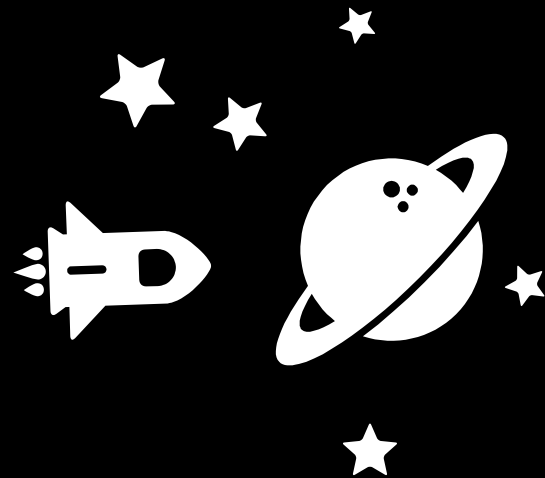
# Wrapped types?



Meh.

Try it out.

# Type brands



# Type brands let us have psuedo-nominal types in TypeScript.

```
type FilePath = string & { _FilePathBrand: any };
```

# **Live Demo Time!**

Let's see a real example of using type brands in action.

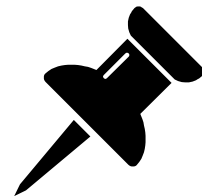
# We can even have inheritance-based patterns

```
type FilePath = string & { _FilePathBrand: any };  
type RelativeFilePath = FilePath & { _RelativeFilePathBrand: any };  
type AbsoluteFilePath = FilePath & { _AbsoluteFilePathBrand: any };
```



# **Live Demo Time!**

Let's see a real example of using type brands in action.



# Type guards and more!

- We can create type guard and factory functions to improve ergonomics and give us more compiler safety

# **Live Demo Time!**

Let's see a real example of using type brands in action.

# Addendum: number, boolean...



- Other literal types suffer from the same problem as `string`
- We can apply the same branding techniques to these types

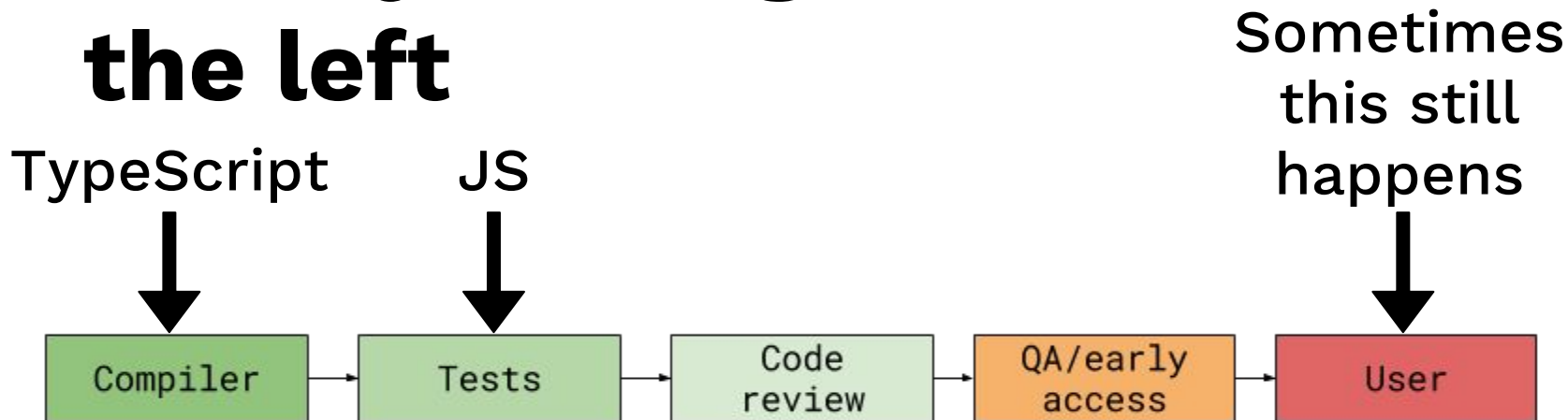
```
const repeatString = (str: string, numTimes: number) =>  
  Array(numTimes).fill(str).join("");
```

```
const repeatString = (str: string, numTimes: number) =>  
  Array(numTimes).fill(str).join("");
```



**Bad!**

# Move your bugs to the left

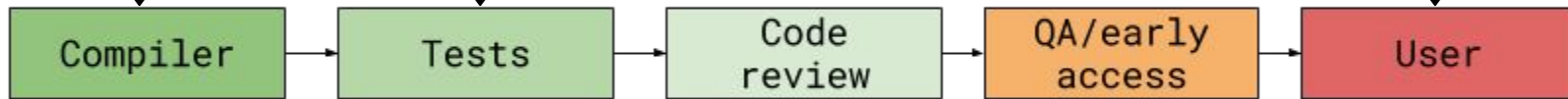


# Move your bugs to the left

TypeScript


JS

~~Sometimes  
this still  
happens~~



Your code now!



A photograph taken from the perspective of someone inside a wooden boat, looking forward towards the horizon. The boat's wooden planks and gunwales are visible in the foreground. The water is very calm, reflecting the soft, warm light of the sunset sky. In the distance, there are silhouettes of trees and reeds along the shoreline. The overall mood is peaceful and contemplative.

Go onwards!  
Make your code correct.  
Teach others what you have learned.

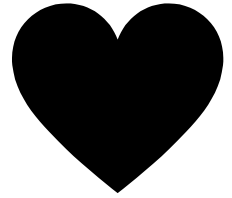
# Thanks!



**Any questions?**

You can find me at

- @the\_jkillian
- [github.com/jkillian](https://github.com/jkillian)



# Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by [SlidesCarnival](#)
- Photographs by [Unsplash](#)
- Muscle clipart from [Clker](#)



**Questions?**