1 Intro

2 Problem

2.1 Conceptual Obfuscation

2.1.1 Overarching concepts hidden

- Language allows bad practice by working even with "improper" code a lot of time

- Most js is pretty simple, allowing devs to get by without real knowledge of the language

1. Environment

    - No compiler allows mistakes to slip through
        - Also no standard
    - Browser as an evironment
        - different browsers use different APIs
        - No set standard
        - web is always changing

3 New version

- Much of ES6 is simply a new way to write the same code

- This concept is known as syntactic sugar

3.1 Syntactic Sugar

- Syntactic sugar: the idea of superfluous syntax changes, without meaningful semantic changes underneath

    - Like sugar, often added to help taste, but not necessary for nutrition

3.1.1 Benefits

- Makes it easier to write code

3.1.2 Problems

- not necessarily easier to read - especially if you haven't learned the new way

- The old way isn't defunct - now even more variety and less standards in place

- Slow adoption actually gives a path forward for programs whose sole purpose is to convert the new version into the old version

4 Specifics

4.1 Const/Let

Made easier: variable declaration

- One of the true improvements in ES6
- Fixes a major pain point in js
  - While other changes also address pain points, js was alone in its variable declaration.
- JS can now work more reliably as a result.

4.2 Class

Made difficult: prototypal inheritance

- The worst offender in unnecessary obfuscation
- Wraps an already-misunderstood concept in a new syntax
- While the previous syntax was confusing, so was the concept.
  - The concept is the same
  - A new syntax does not make the concept easier
    - in fact, it is harder. What is actually going on does not correspond with the new syntax

4.3 Arrow function

Made difficult: this keyword

- Unnecessary
- Saves a bit of typing in its best case
- Saves dealing with another difficult concept at worst

5 Conclusion