

# State of the Open edX Frontend



**Ari Rizzitano, edX**

**May 30, 2018**

**Open edX Conference, Montréal, QC**



**Ari Rizzitano**



**software engineer**



**@arizzitano**



**arizzitano@edx.org**

# Meet the Frontend

**Technology**

**Architecture**

**Extension**

# Meet the Frontend

**What's the  
frontend?**

**Code related to the  
Open edX user interface,  
including display, layout,  
interactivity, animation,  
form submission,  
validation, testing, etc...**

ES5

 BACKBONE.JS

 mako

UNDERSSCORE.JS

 **jQuery**  
write less, do more.

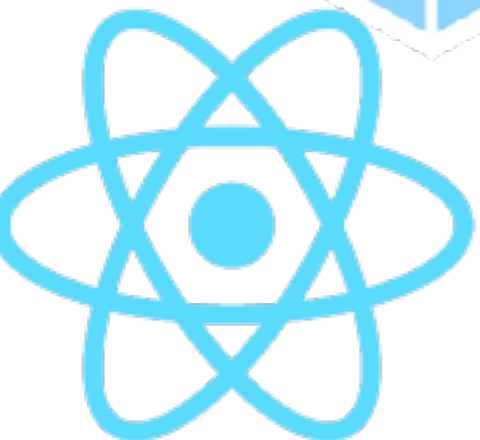
What's the  
frontend?

 Sass



webpack

ES2015



React

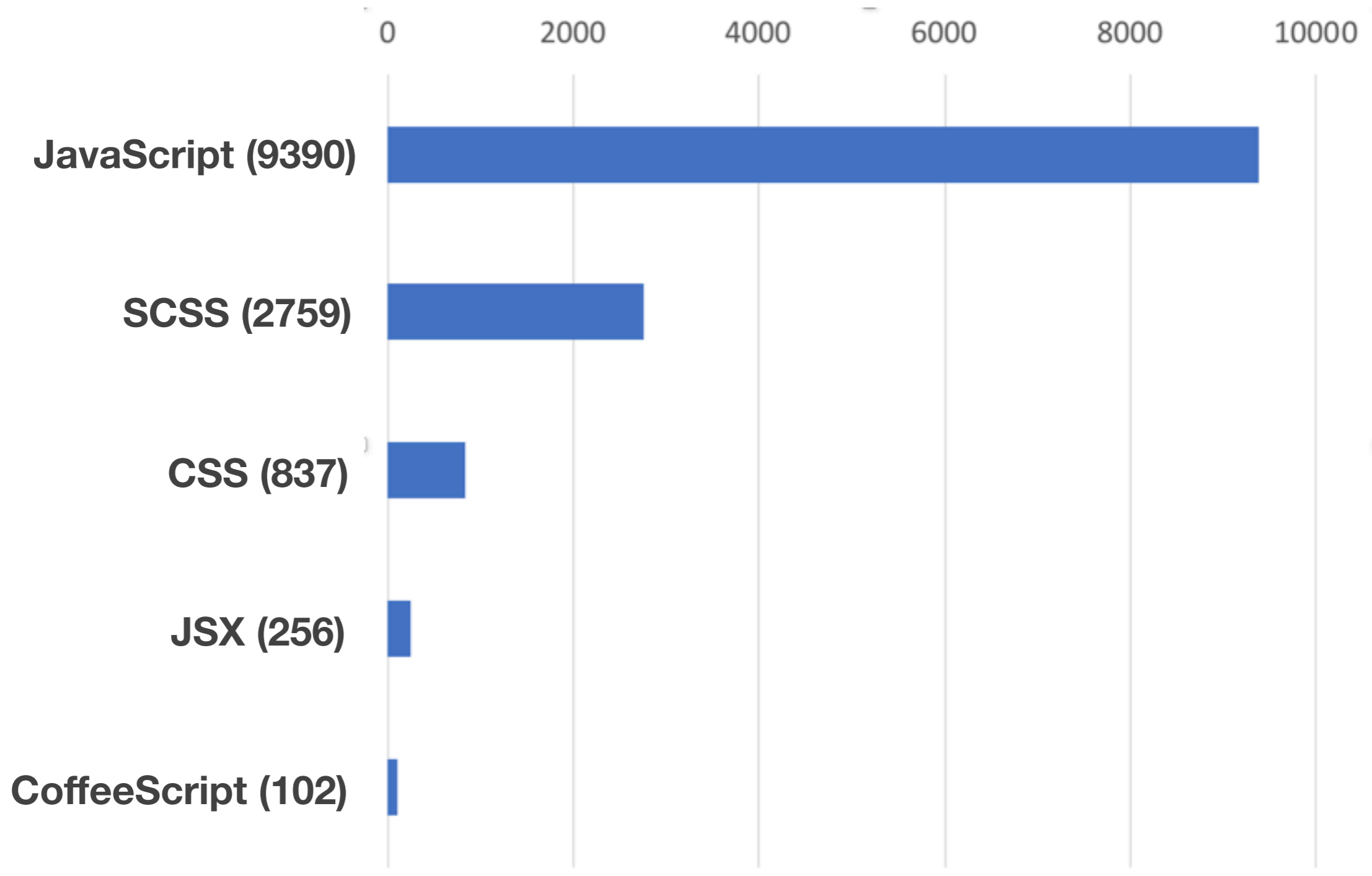


Redux

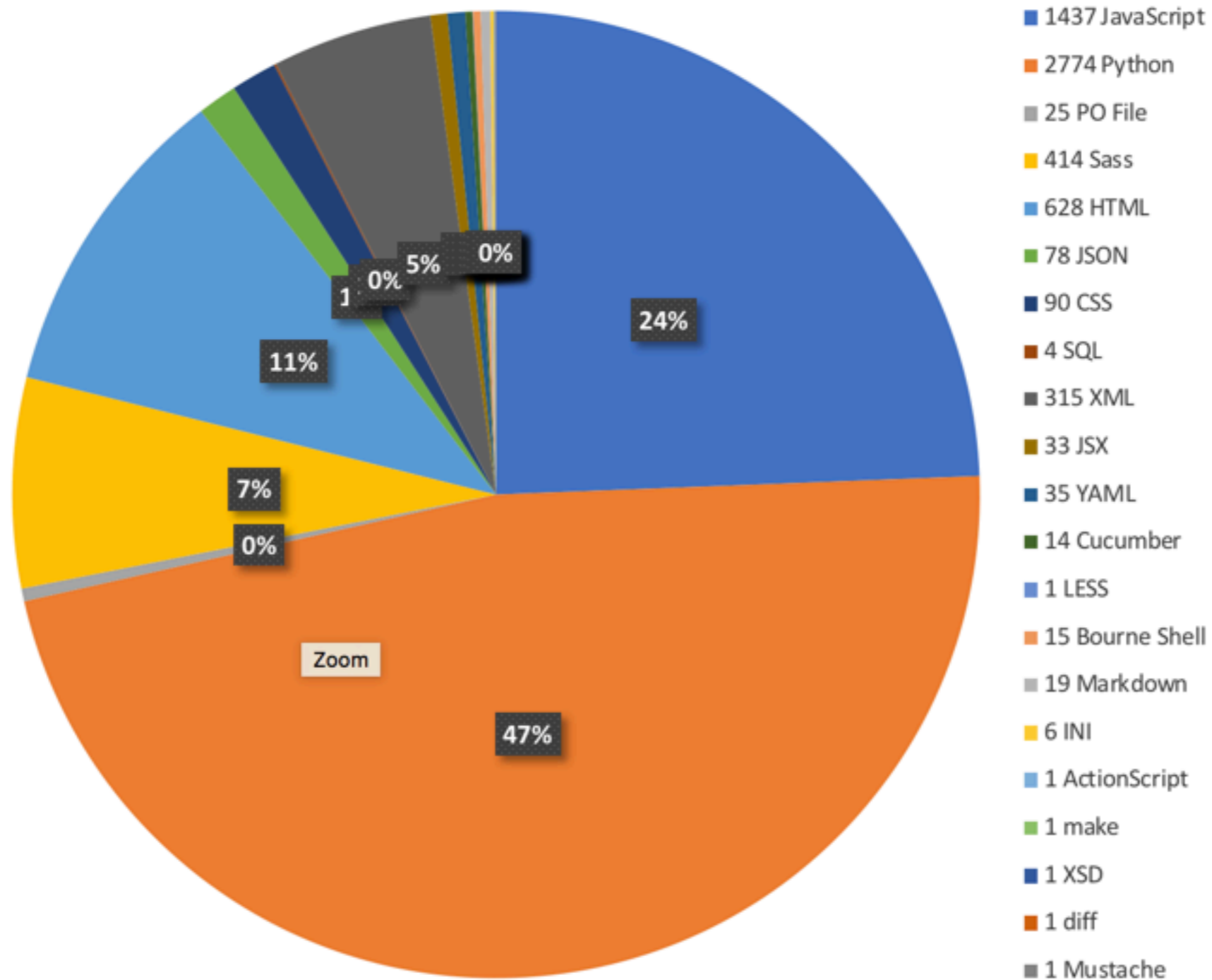
**Open edX  
code that runs  
in browsers.**



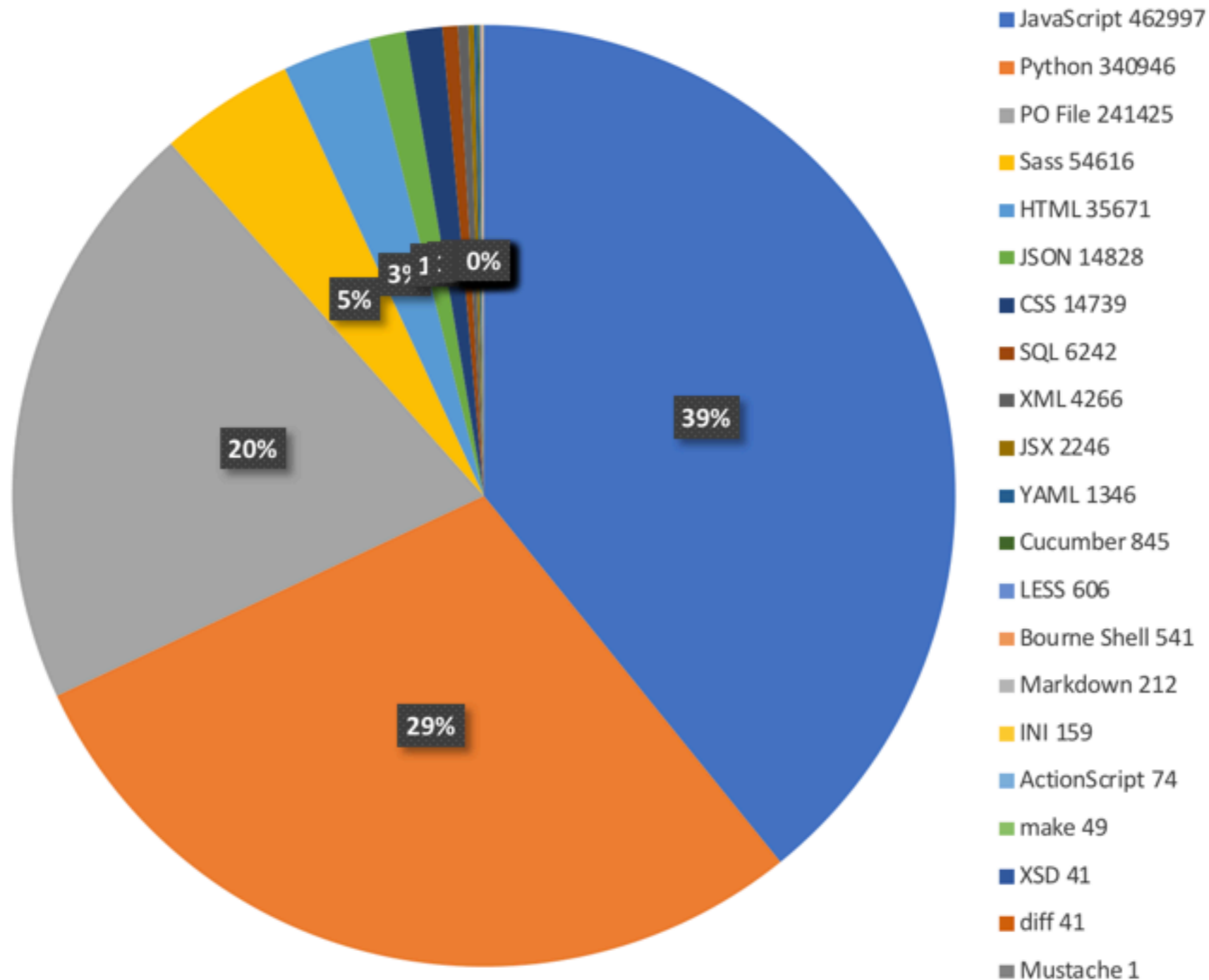
# File Count (all edx repos)



# File Count (edx-platform)



# Lines of Code (edx-platform)



**It's not just a  
pretty face.**

**Technology**

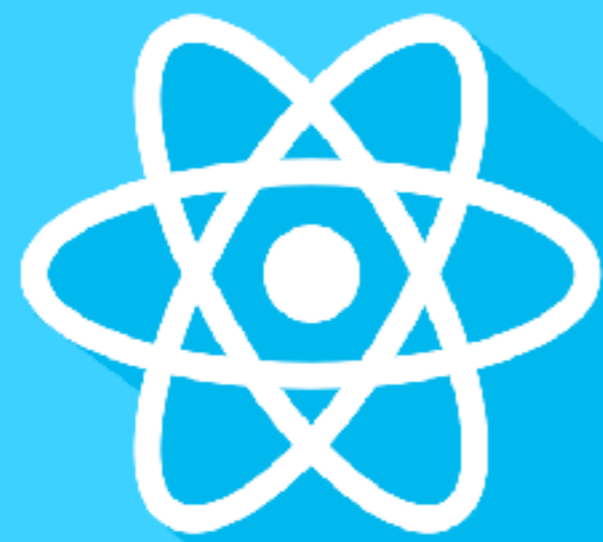
**ES2015**



**webpack**

# Webpack

- Enables modern JS and ES2015
- Used in 15 repositories to build static assets, processing hundreds of files
- Replacing Require and `update_assets`
- Work is underway to process XBlock assets with Webpack

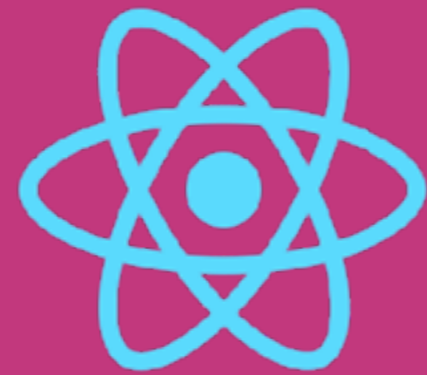


**React**



**Redux**

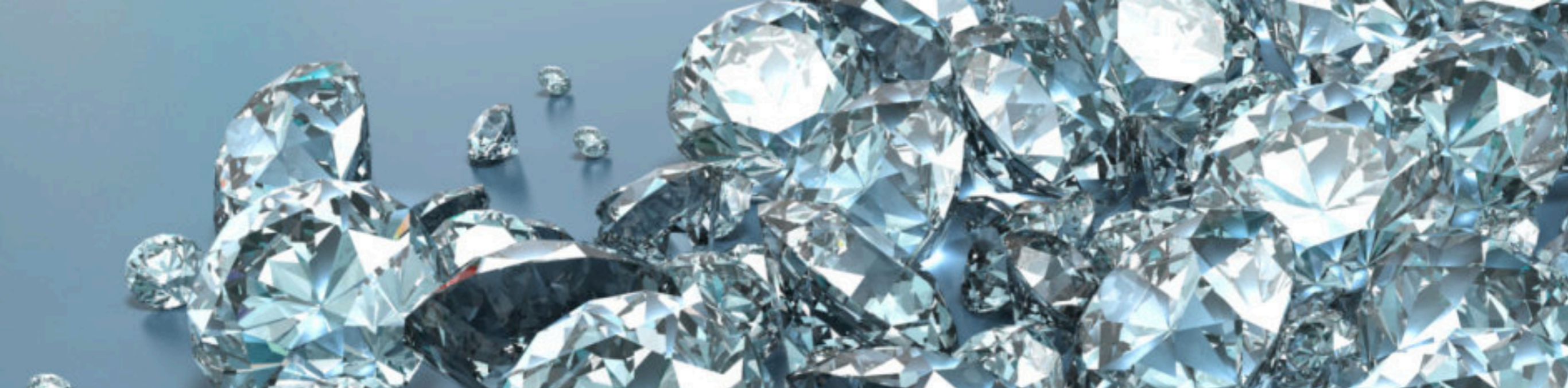




# React

- React is being used to build all new UI
- ReactDOM enables devs to embed React code within Django/Mako templates
- Redux as data layer, tested with Jest + Enzyme
- Now using an MIT license 🙌

**New projects**




# Accessible UI components





# Paragon

- edX's accessible component library
- Written in React with Bootstrap, released via npm
- Generic, reusable, embeddable chunks of UI: dropdowns, buttons, inputs, etc.
- <https://github.com/edx/paragon>

A photograph of a store display featuring several shelves of cowboy boots. The boots are arranged in rows, showcasing a variety of styles, colors, and materials. Some boots are plain, while others have intricate patterns or designs. A small sign on the top shelf indicates a size of 8.5 for men. The background is dark, making the boots stand out.

# Maintainable styling system

# edx-bootstrap

- Open edX-flavored Bootstrap package
- Contains a first pass at an initial example theme
- Used by all new Open edX frontend code to style new UI
- <https://github.com/edx/edx-bootstrap>



# Frontend app starter kit





# front-end-cookie-cutter & create-edx-react-app

- Reference code and cli generator for new frontend applications
- React, Redux, Webpack, edx-bootstrap, Paragon
- Facilitates consistency across new projects
- <https://github.com/edx/front-end-cookie-cutter-application>
- <https://github.com/edx/create-edx-react-app>





# First separate frontend

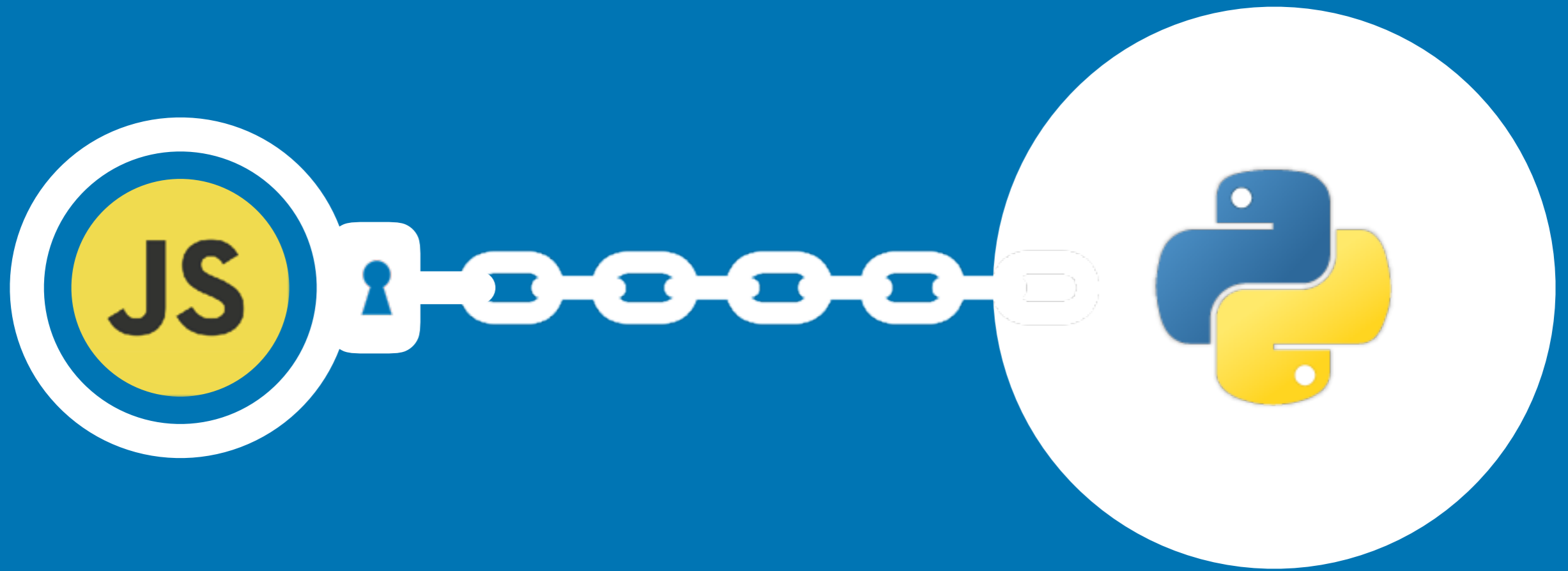




# studio-frontend

- edX's first separately developed and released frontend repo
- Studio UI built with React, Paragon, and Bootstrap
- Installed and rendered inline in edx-platform
- <https://github.com/edx/studio-frontend>

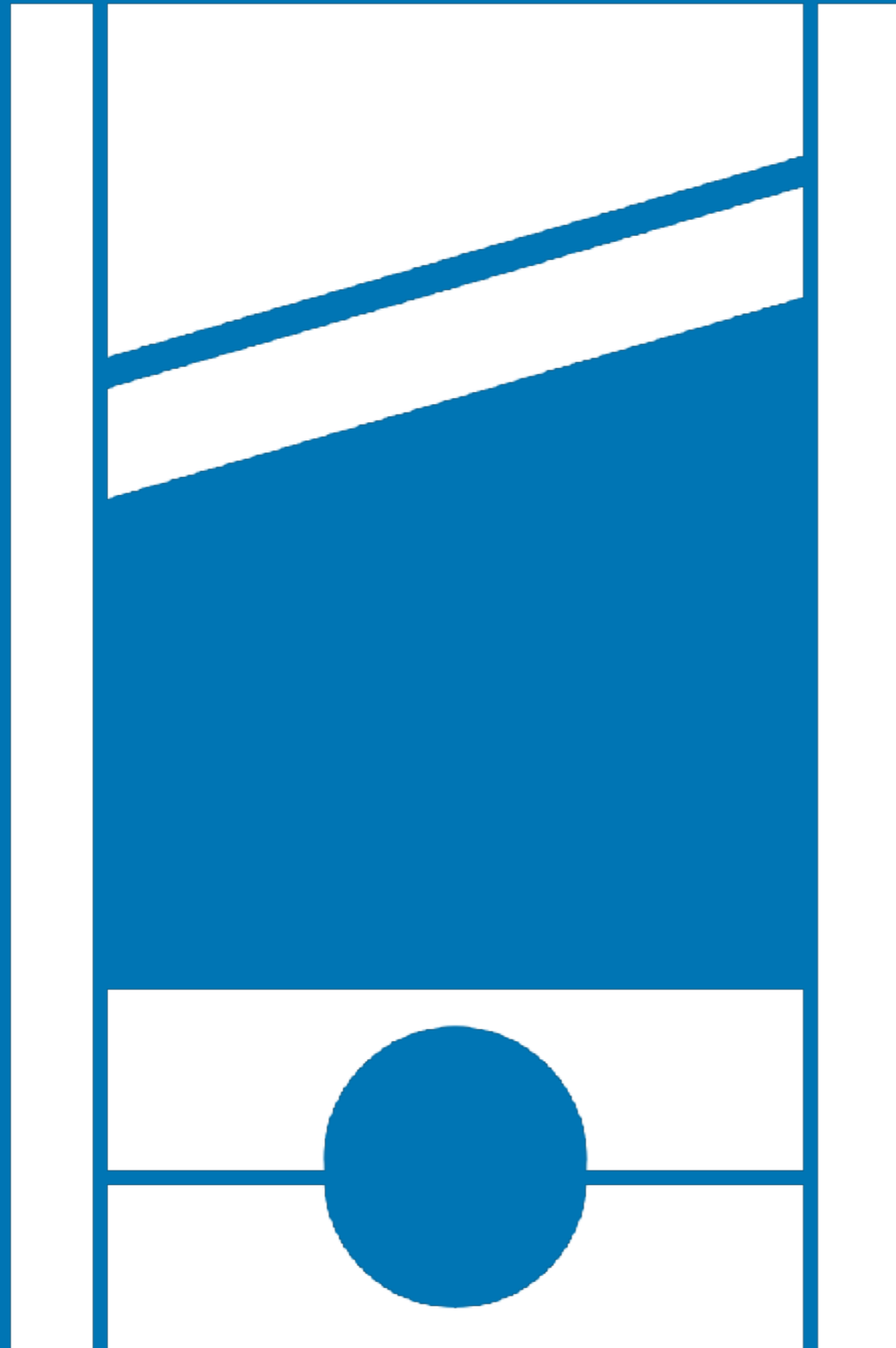
**Architecture**



**Our JS depends  
heavily on Python**

# Limitations

- JS is built, served, and deployed with Python
- Devs must hack around system limitations to use the latest and greatest frontend tools (e.g. `django-webpack-loader`)
- Full-scale page conversions are time-consuming and costly

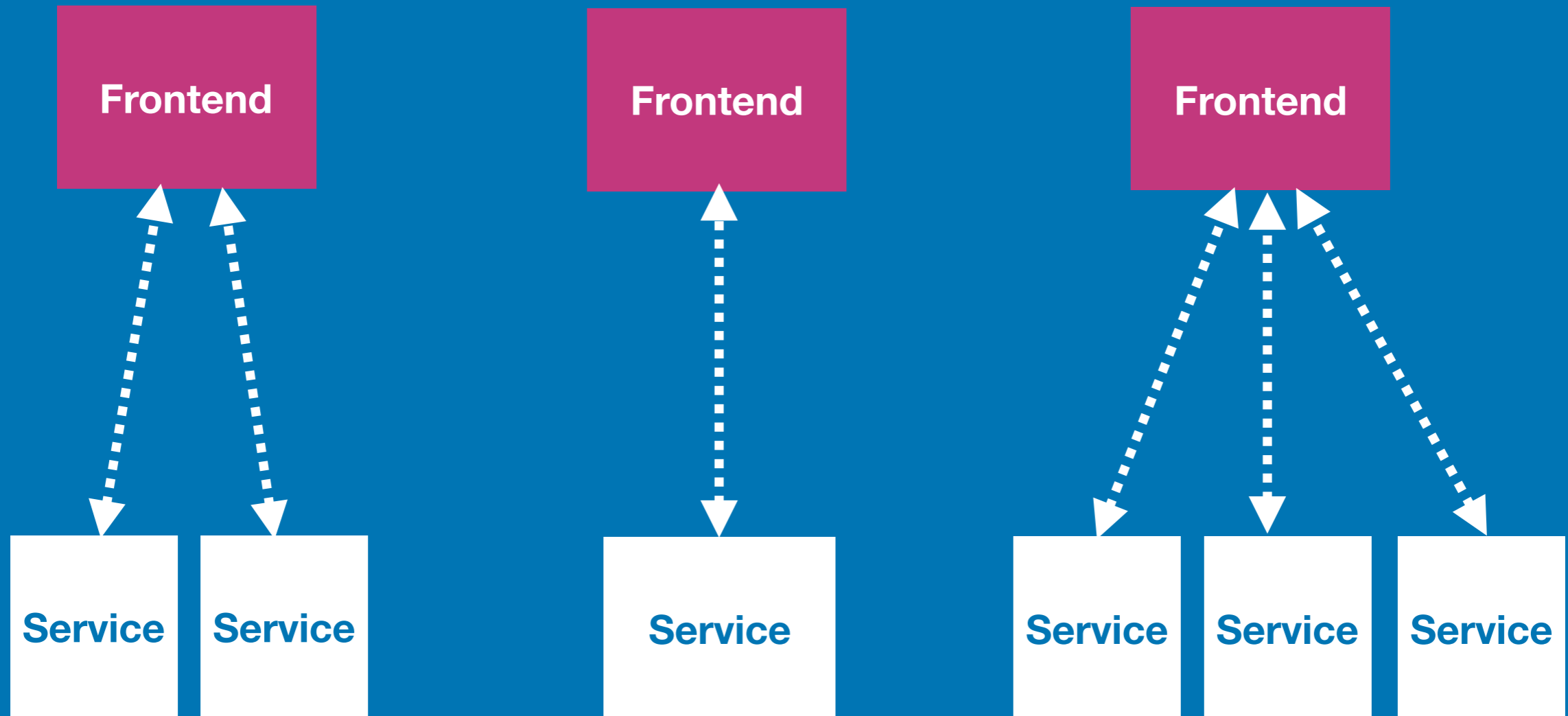


**It's time to  
separate  
frontend  
from  
backend**

# Frontend/Backend Split

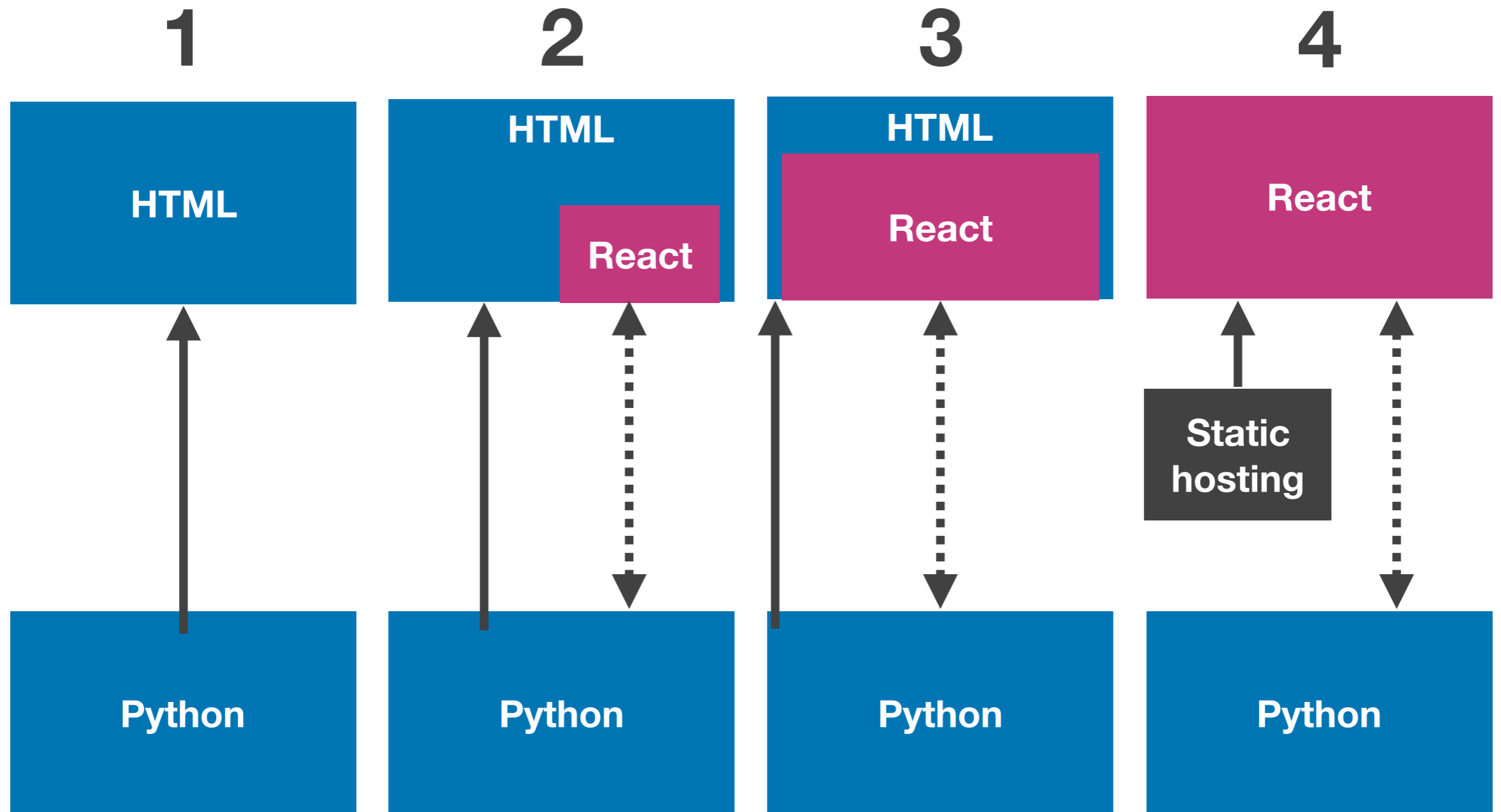
- Independent build, release, and deployment based on shared infrastructure
- Faster dev and delivery cycles
- Greenfield environment for modern tech

# Architecture Goal





# Incremental Path



**“Decoupled  
React frontends  
are easy! We’ve  
already done it.”**

**—you, probably**

**“Decoupled  
React frontends  
are easy! We’ve  
already done it**

**Theming**

**Feature  
velocity**

Decoupled

React frontends

are easy! We've

already done it

**Theming**

Feature  
velocity

accept

**BACKWARDS**

**COMPATIBILITY**

ive

already done


The remaining

**Extension**



**Customize the user  
interface of the  
Open edX platform**

# Style



Welcome to  
exampleX



*Welcome to  
exampleX*

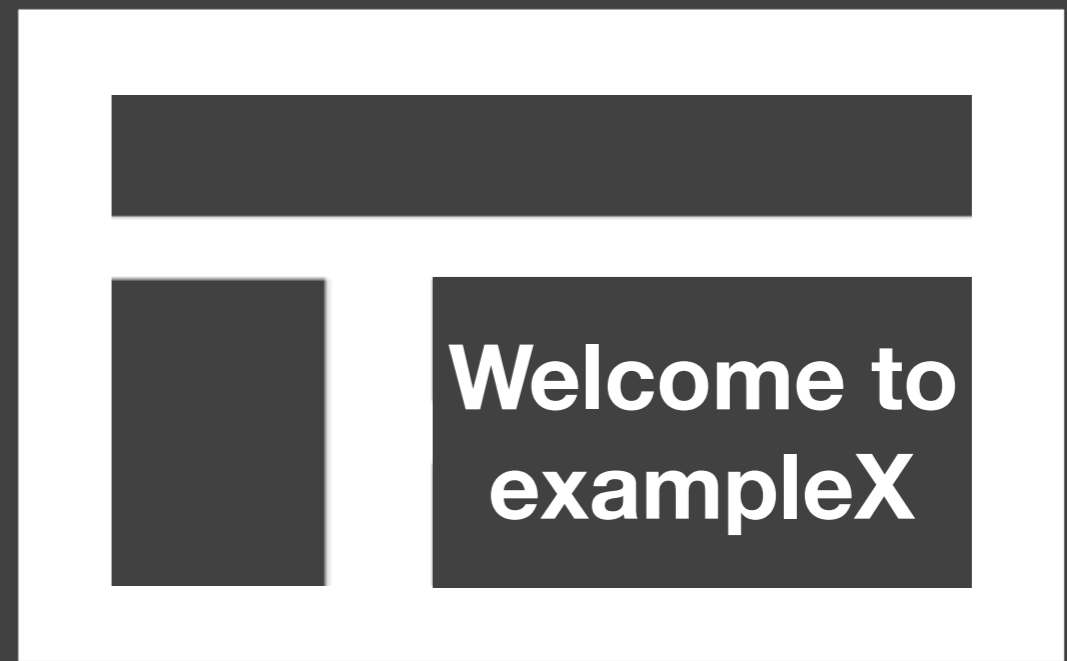
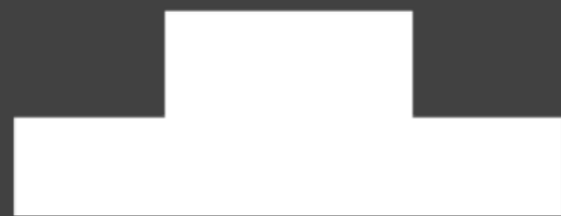
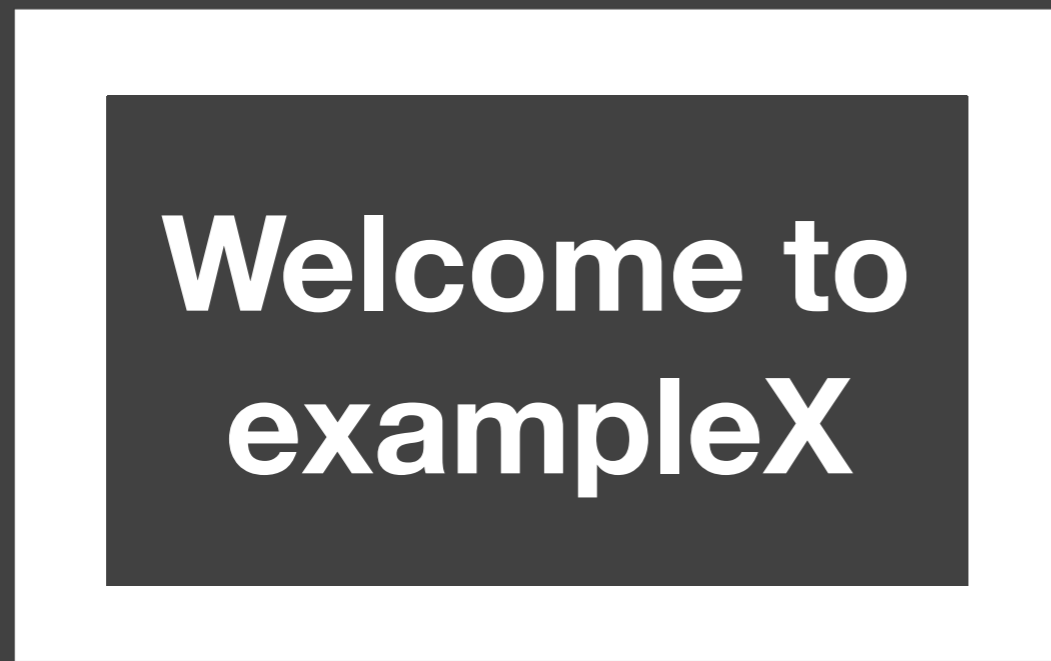
**Scope: look-and-feel and branding elements**

**Examples: colors, logos, fonts**

**Current Tech: CSS/Sass partials, Django templates**



# Interface



**Scope: interface elements**

**Example: header elements, tabs, etc.**

**Current Tech: Django templates, Web Fragments**

# Content



**Welcome to  
exampleX**



**exampleX  
welcomes you**

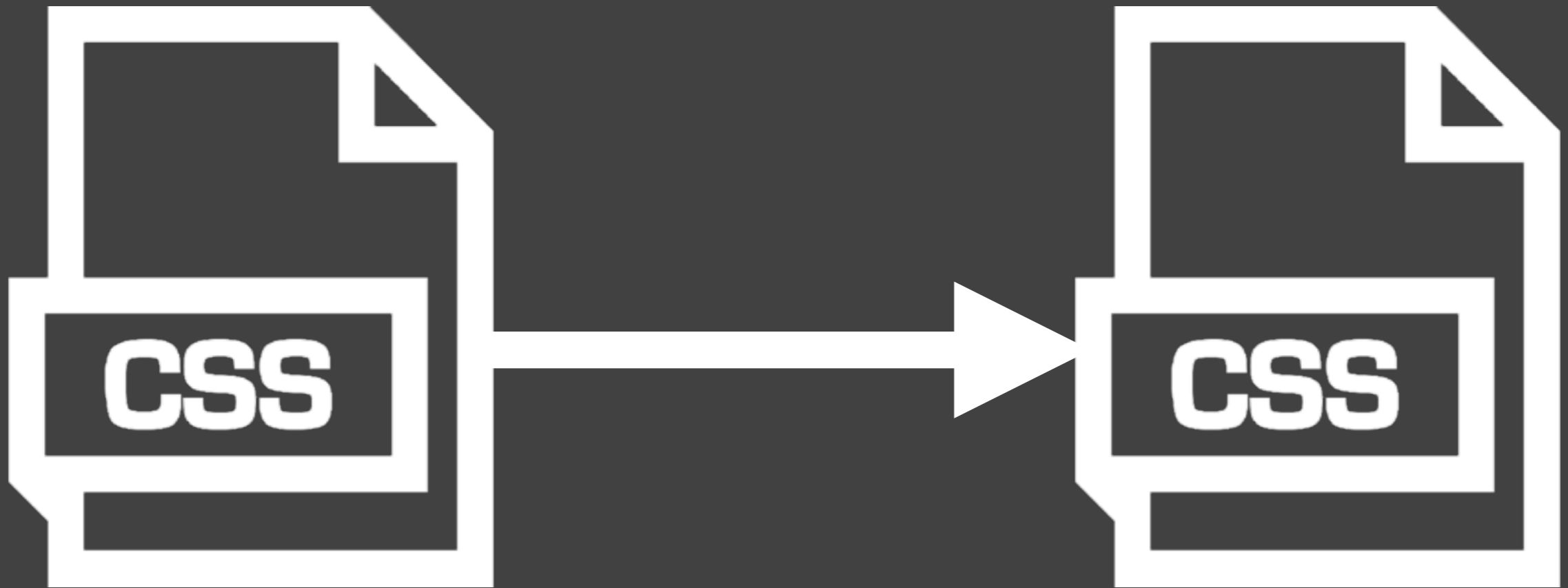
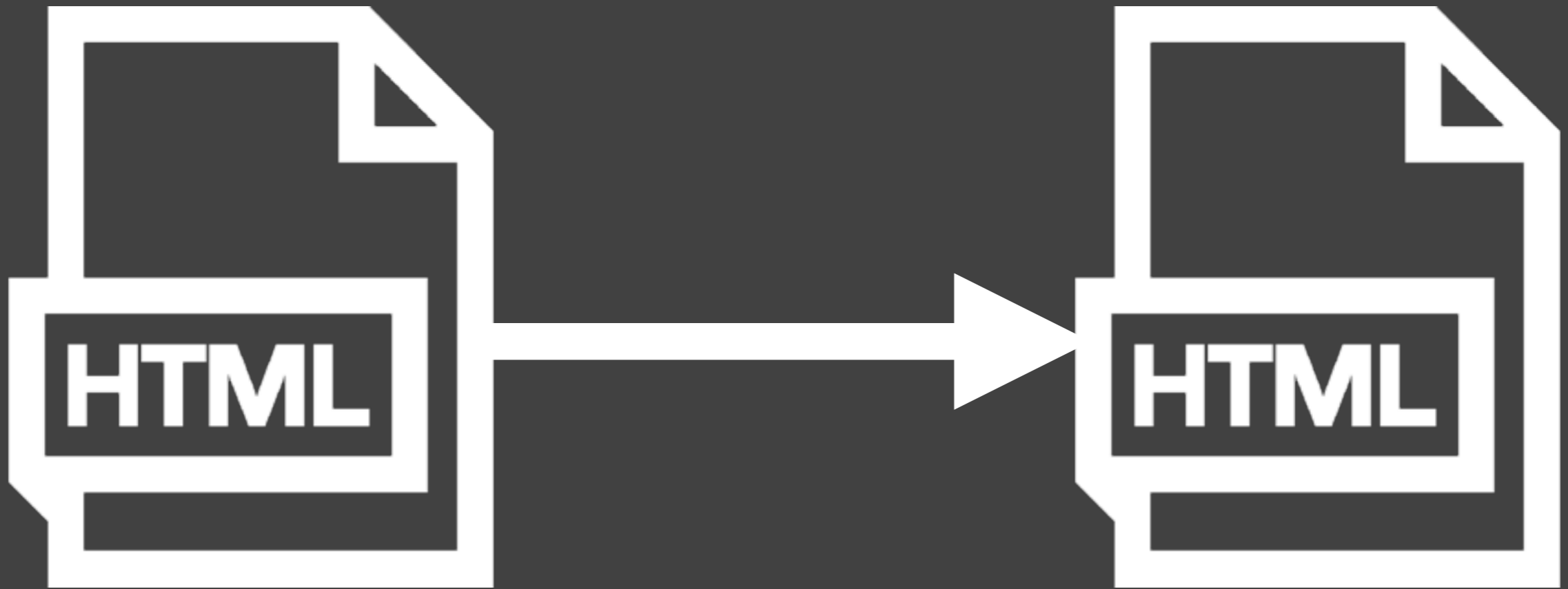
**Scope: static semantic content, such as strings**

**Examples: onboarding text**

**Current Tech: Django templates**

**Solution:  
theming**

**Comprehensive &  
Stanford &  
Microsites &  
Pattern Library**



**Template overrides  
are optimized for the  
Django ecosystem**

**APIs**







**Core contributors break  
theming *constantly*  
without being aware of it**



**Ironically, we also  
struggle to make big  
innovations.**

**Goal: enable style,  
interface, and content  
customization  
*independently* with  
clear APIs**

**First stop:  
style theming**

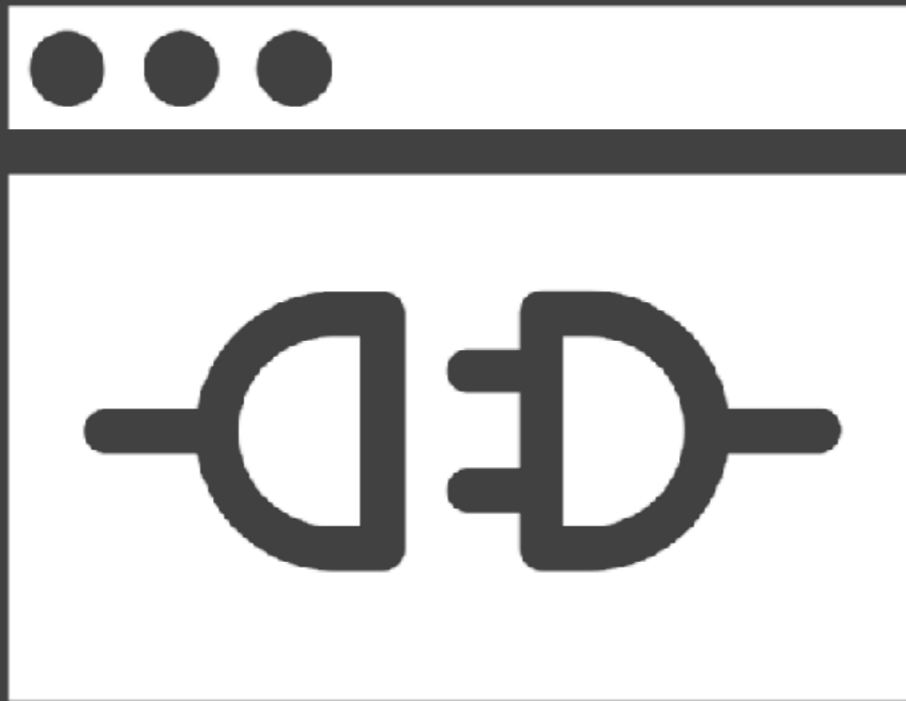


Welcome to  
exampleX



*Welcome to  
exampleX*

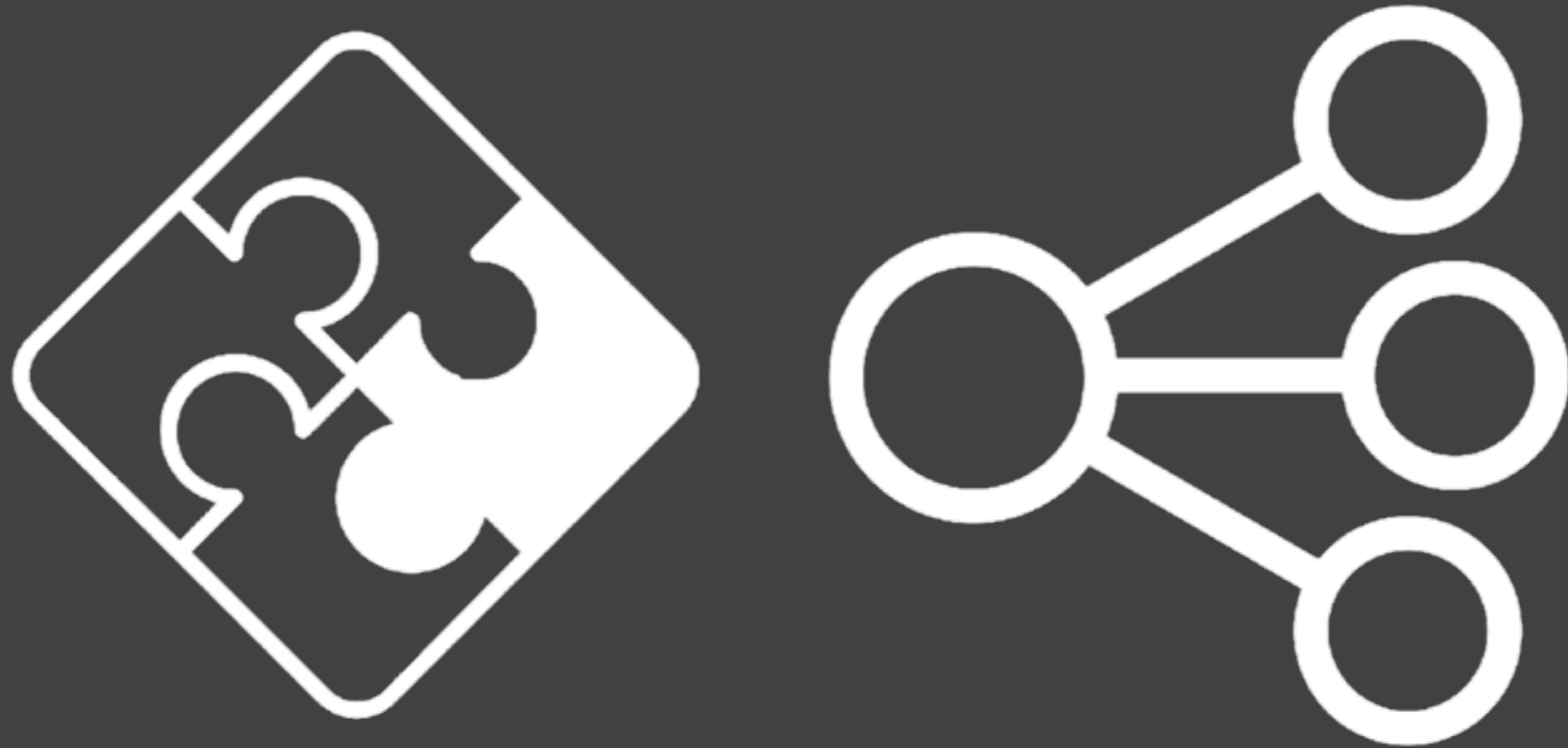
**Style theming will cover look  
and feel changes and  
branding overrides**



**Official style theming API  
based on Bootstrap,  
supported in new and  
legacy applications**



**edX core contributor team  
will use this system and  
keep breaking changes to  
a minimum**



**Interface modification and  
content overrides will be  
handled separately.  
How? TBD!**





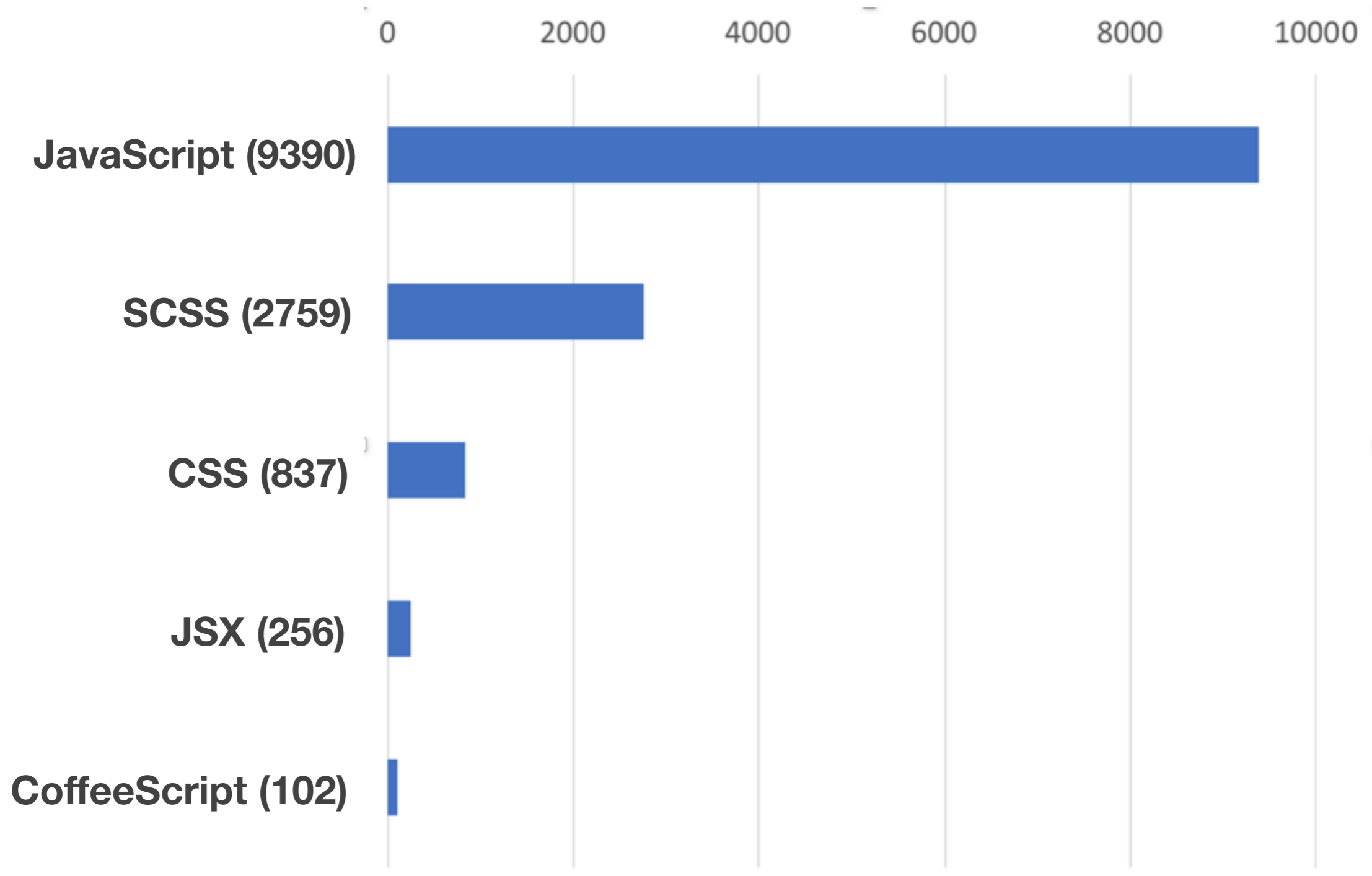
**Existing systems' support  
for style theming will be  
deprecated over the course  
of several named releases**

**Let's chat  
about it.**

**GET**

**INVOLVED**

# File Count (all edx repos)





**#front-end**



**<https://github.com/edx>**



**10:45am tomorrow**

**Thank you!**