



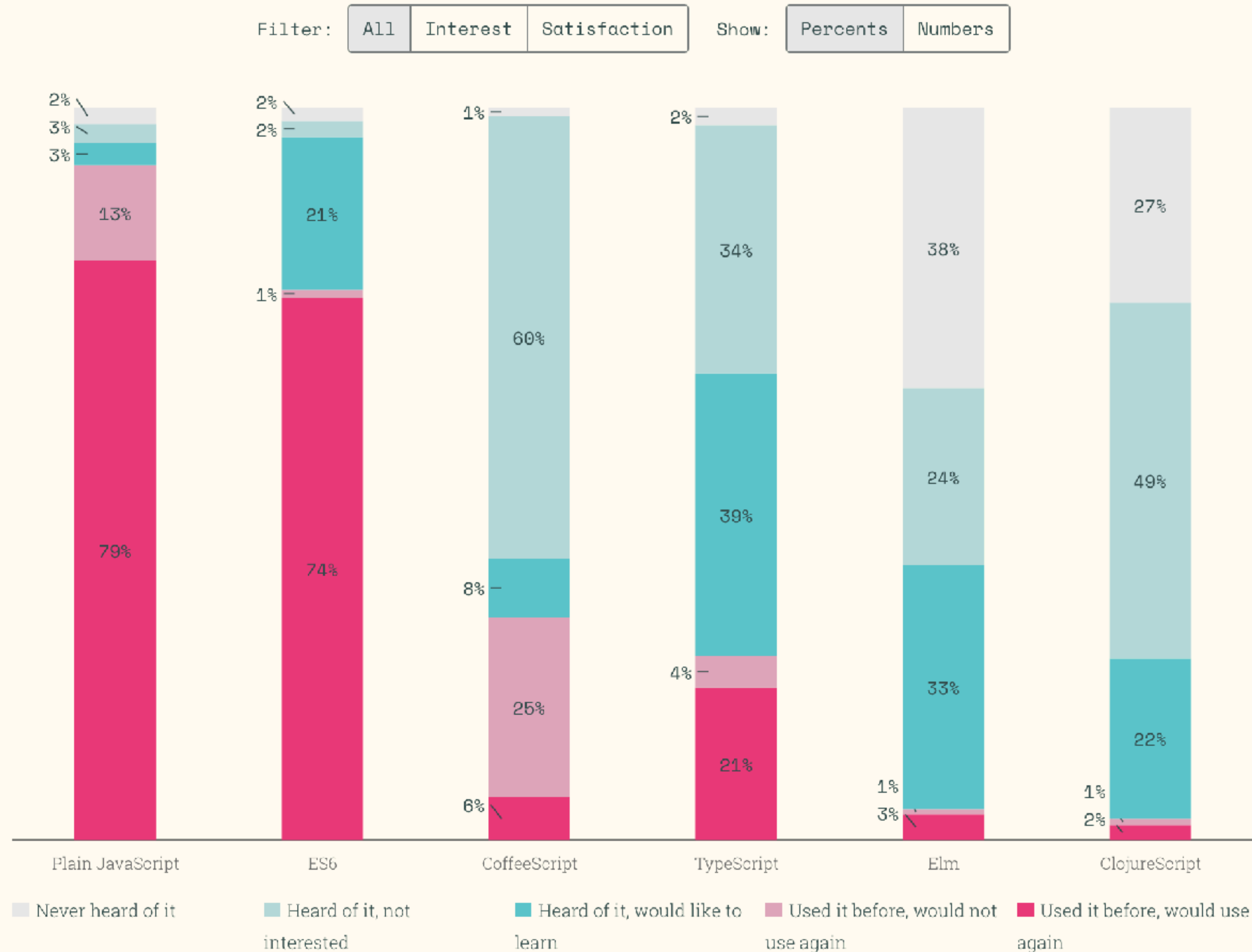
JavaScript för 2017 och framåt

ES6, webb-API:er

**Diskutera: Vad betyder frontend
och backend idag?**

Läget för JS år 2017

JavaScript Flavors



“It’s interesting to look at ES6: a large proportion of developers have already jumped on the bandwagon, and almost all (89%) of those who haven’t yet want to learn it as well. [...]

While plain JavaScript has the largest usage numbers, in terms of satisfaction the big winner here is ES6, and I think it’s safe to say it’s now the default way to write JavaScript apps.”

– <http://stateofjs.com/2016/flavors/>

Presentational markup

All basfunktionalitet och logik som driver
sajten bör byggas med vanilla JS (av
frontendare!) för att vara maximalt möjlig att
påverka i framtiden

Linting

- Precis som med linting av CSS så är det viktigt här av samma skäl
- Vanligaste lintern för JS är idag ESLint
- Setup funkar i princip likadant som för Stylelint

Transpilers

- Babel
- Traceur

Att transpilera ES6 till ES5 är även det, för vissa webbläsare, alldeles för avancerat (IE9 har inte helt stöd om kompatibilitetsläge)

Dynamisk typning

Paradigm

- Deklarativ
- Imperativ
- Funktionell

```
<article>
  <header>
    <h1>Declarative Programming</h1>
    <p>Sprinkle Declarative in your
verbiage to sound smart</p>
  </header>
</article>
```

```
$("#btn").click(function() {  
    $(this).toggleClass("highlight")  
    $(this).text() === 'Add Highlight'  
        ? $(this).text('Remove  
Highlight')  
        : $(this).text('Add Highlight')  
})
```

`const add = (a, b) \Rightarrow a + b;`

TypeScript

Nya teknologier/API:er

Fetch

Promise

Session/Local Storage

Service Worker

ES6-primer

II FE

let + const

querySelector

Fat arrow

Template string literals

Default parameters

Använd konsolen för att logga:
`warn, error, table, variabler (%d...)`

addEventListener

Event propagation

- bubbling
- capturing

Event handlers

forEach

Deconstructing

Rest parameter

Spread parameter

Chrome Debugger

Namespace

- Om komplext projekt, använd alltid dedikerat namespace och inga globala variabler

Import/export av moduler

- import
- require
- AMD
- UMD
- CommonJS

Synkrona och asynkrona behandlingar

- `foreach...`
- `map`
- `filter`
- `reduce`
- Map och andra funktioner som inte skriver över tidigare värden

Block/Function Scoping

Class

Closures/hoisting

NodeList

Framtiden för JS

Mer funktionell
programmering

Utbredd testning

Lättare FE-bindning för
basala saker: Riots, Rivets...