PRELUDE

HOW THE WEB WORKS

?

HOW THE WEB WORKS

HTTP REQUEST

HTTP RESPONSE

CLIENT

SERVER

HOW THE WEB WORKS

WEB CLIENT

WEB SERVER

HTTP SERVER

APP
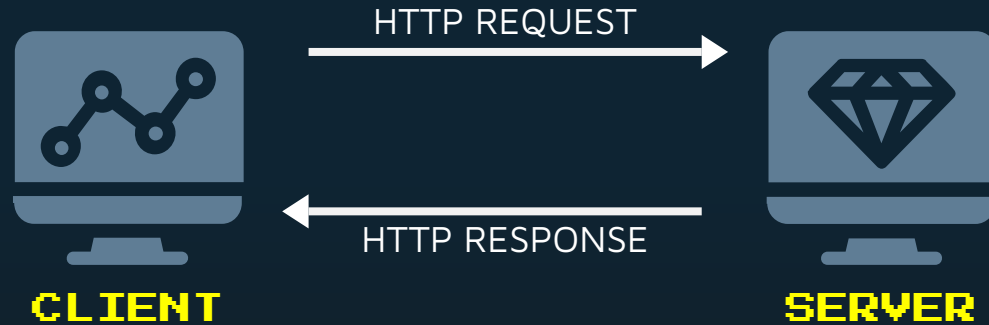
FILES
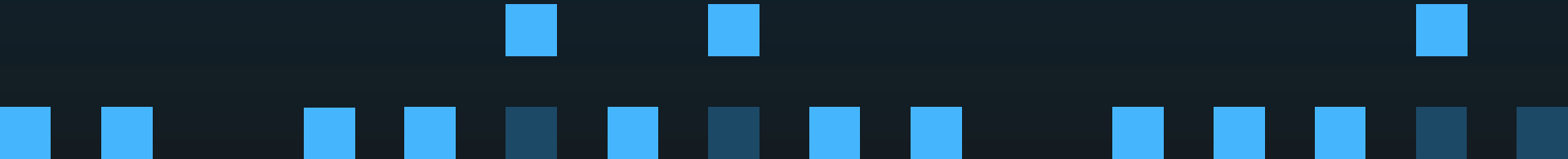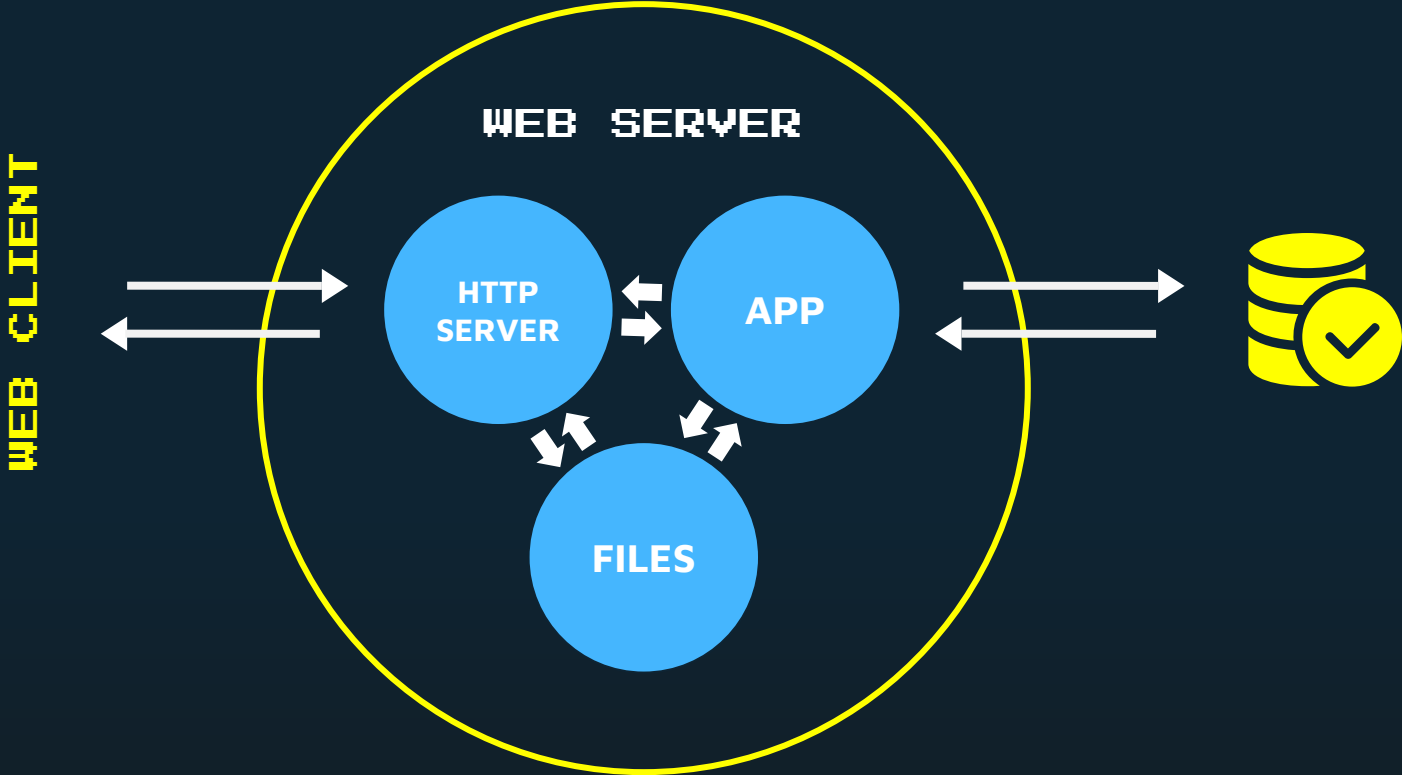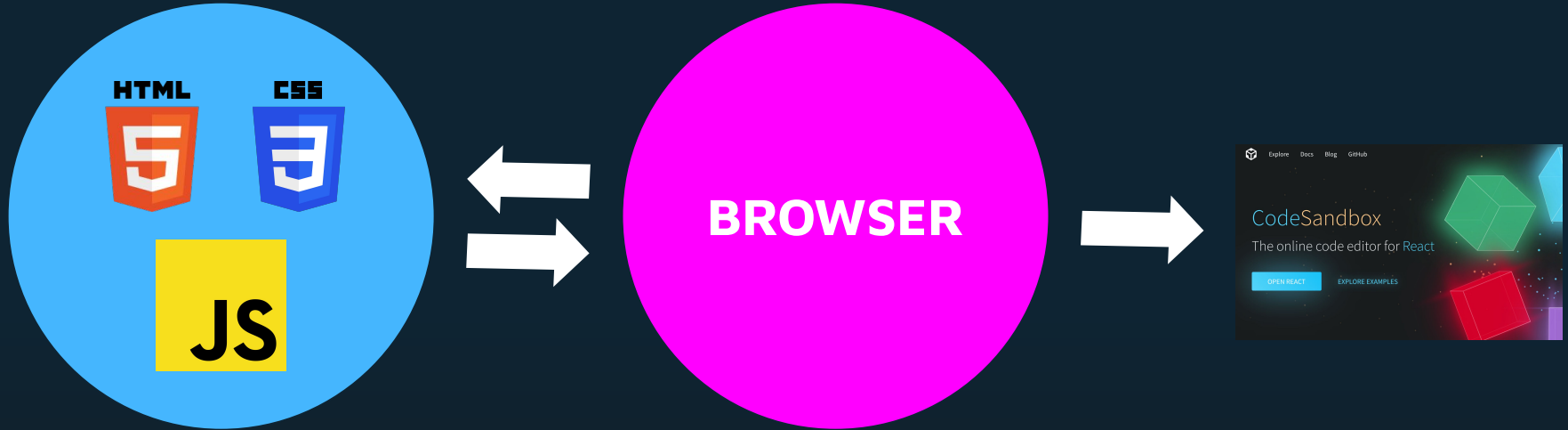
STATIC WEBSITES

DYNAMIC WEBSITES

GET DATA

BUILD WEBSITE

HTML CSS JS

BROWSER

TEMPLATE

EJS

HANDLEBARS

PUG
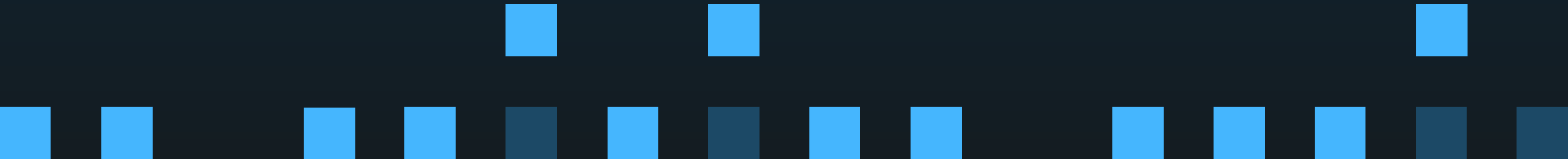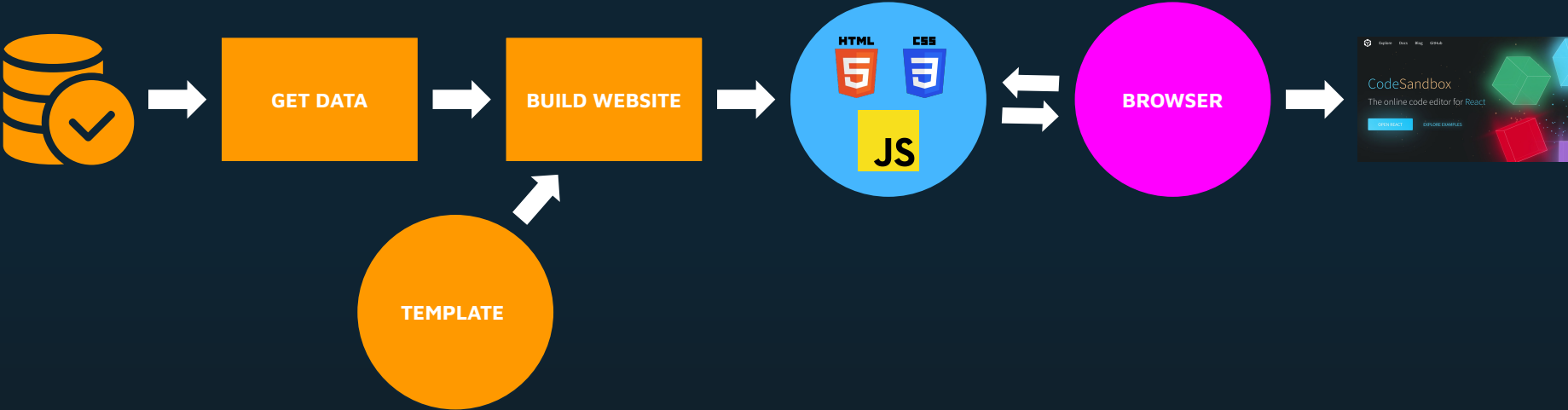
API-POWERED WEBSITES

BUILDING API

GET DATA

JSON

TEMPLATE

BROWSER

BUILD WEBSITE

CodeSandbox
The online code editor for React
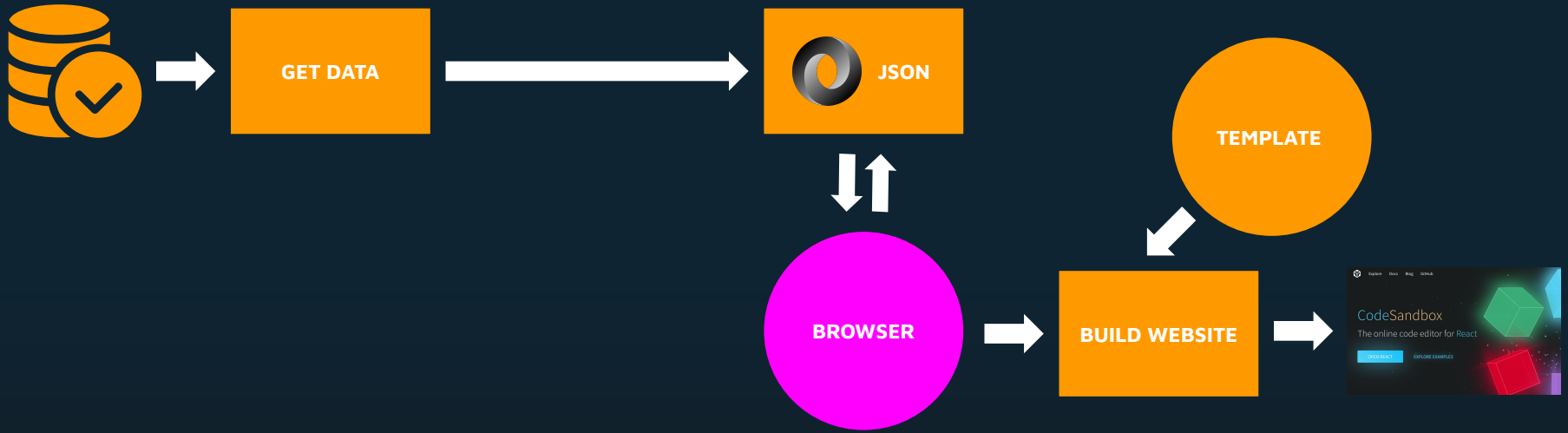
CONSUMING API
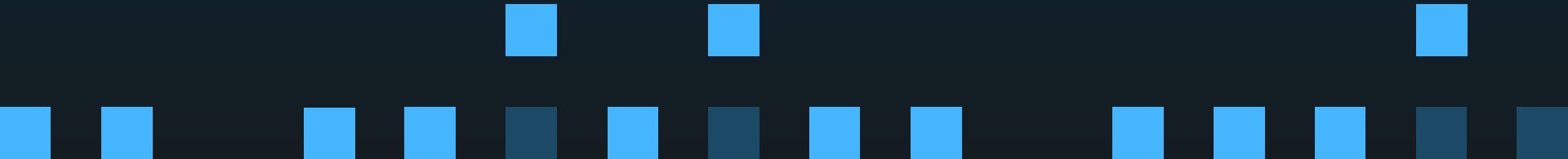
SINGLE PAGE APPLICATIONS (SPA)

The server returns **one single HTML** document along with **Javascript** code

This Javascript code will change (re-render) the page **dynamically**
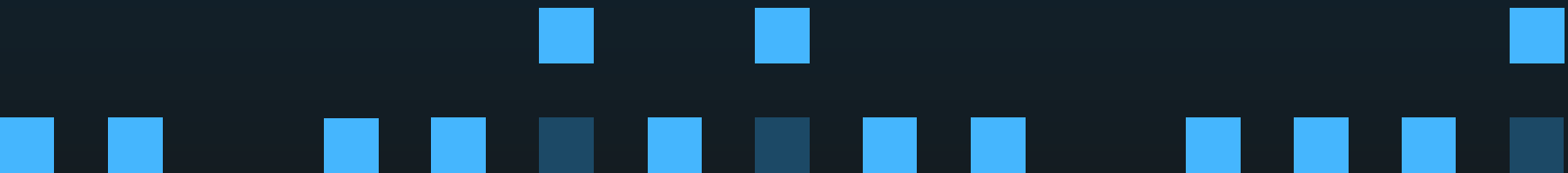
INTRODUCTION

What is Svelte?

Svelte is a **compiler** that generates minimal and highly optimized Javascript code

"*Svelte converts your app into ideal JavaScript at* **build time**, *rather than interpreting your application code at run time*"

https://svelte.dev

"*Svelte is a radical new approach to **building user interfaces**. Whereas traditional frameworks like React and Vue do the bulk of their work in the browser, Svelte shifts that work into a **compile step that happens when you build your app**"*

https://svelte.dev

"It is similar to JavaScript frameworks such as React and Vue, which share a goal of making it easy to build slick interactive user interfaces"

https://svelte.dev

	2016	2017	2018	2019	2020	
React	93%	93%	91%	89%	89%	Svelte
Vue.js	87%	91%	91%	88%	88%	React
Angular	68%	66%	84%	87%	85%	Vue.js
Ember	47%	41%	45%	78%	82%	Alpine.js
			41%	38%	78%	Preact
				30%	78%	LitElement
					67%	Stimulus
					42%	Angular
					27%	Ember
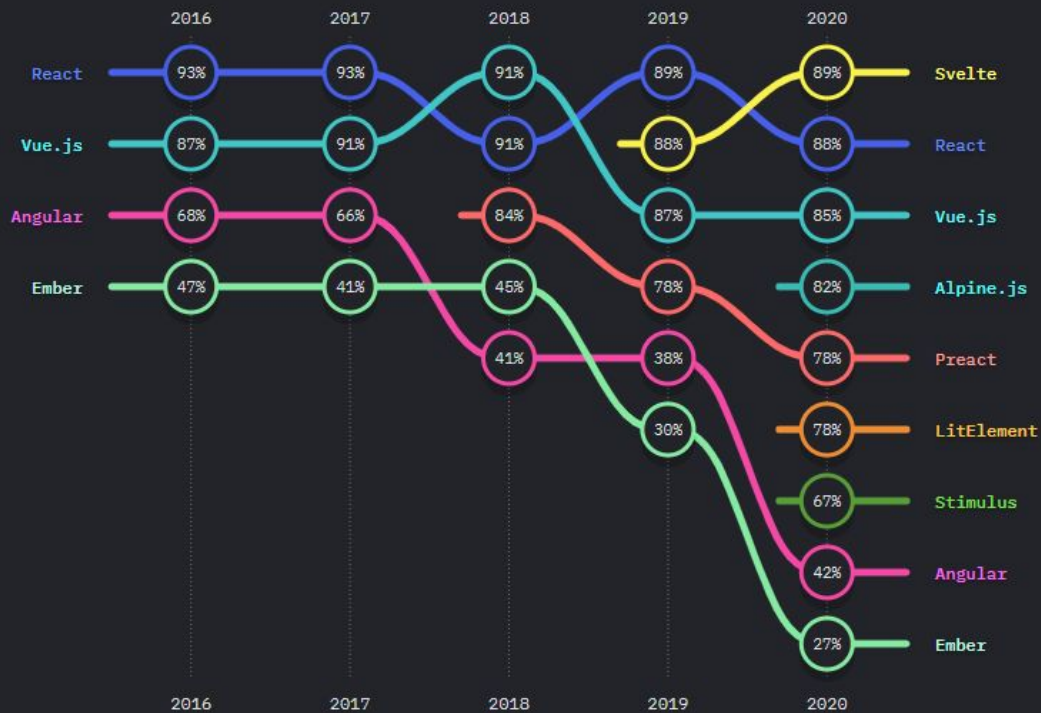
Technologies with less than 10% awareness not included. Each ratio is defined as follows:

- Satisfaction: **would use again** / (**would use again** + **would not use again**)

https://2020.stateofjs.com/en-US/technologies/front-end-frameworks

	Loved vs. Dreaded	Want		66,202 responses

Svelte	71.47%	28.53%
ASP.NET Core	71.47%	28.53%
FastAPI	70.04%	29.96%
React.js	69.28%	30.72%
Vue.js	64.41%	35.59%
Express	62.07%	37.93%
Spring	59.82%	40.18%
Ruby on Rails	57.55%	42.45%
Angular	55.82%	44.18%

https://insights.stackoverflow.com/survey/2021#most-loved-dreaded-and-wanted-webframe-love-dread
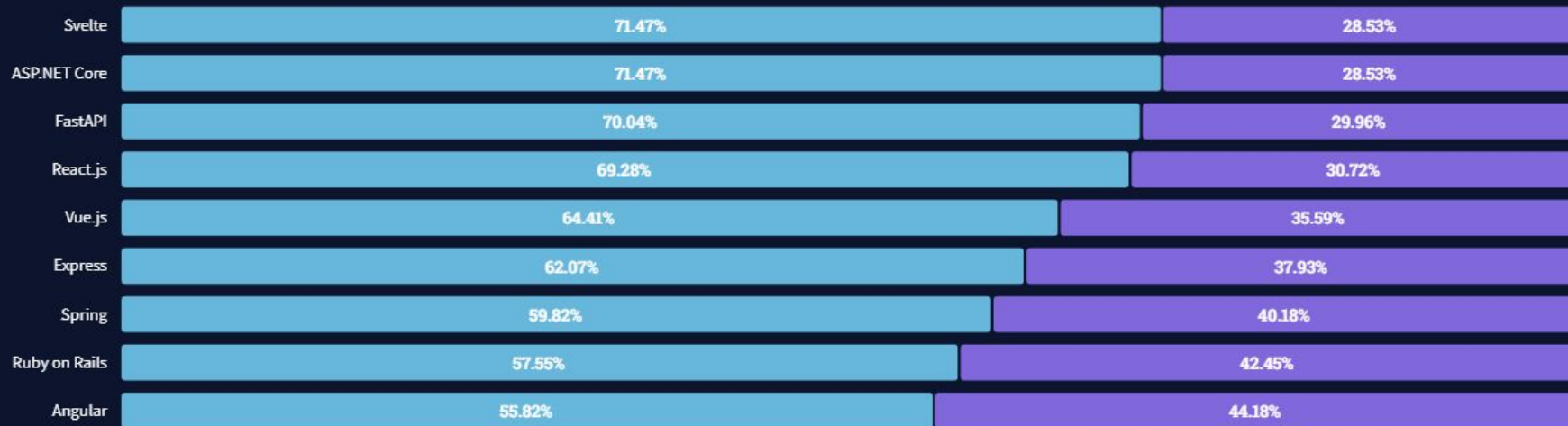
```html
<html>
    <head>
        <meta charset="UTF-8">
        <title>DOM</title>

        <!-- assets here -->
    </head>
    <body>
        <div class="container">
            <h3>Hello World!</h3>
        </div>
    </body>
</html>
```
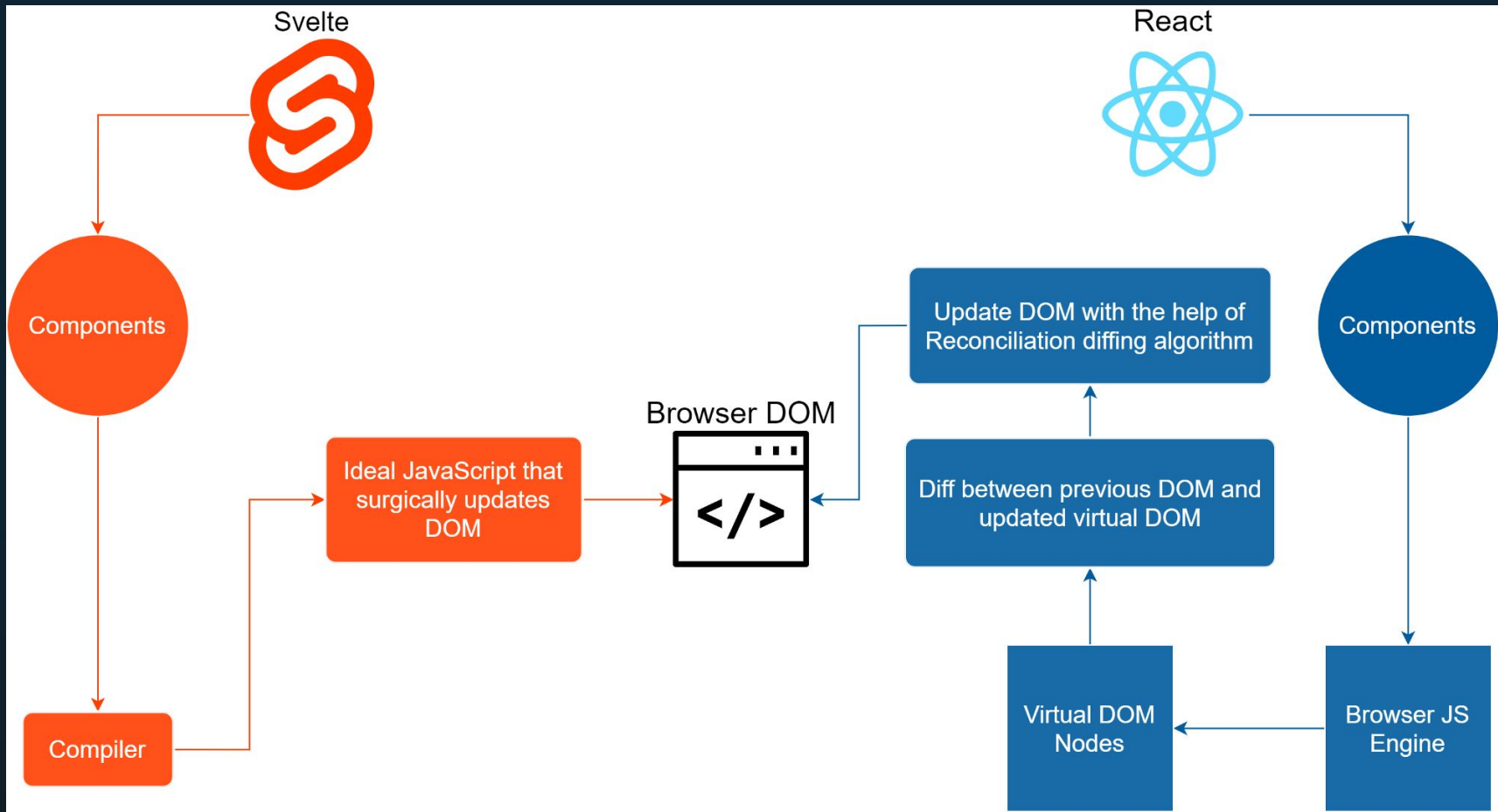
```
└HTML
   ├HEAD
   │  ├#text:
   │  ├META charset="UTF-8"
   │  ├#text:
   │  ├TITLE
   │  │  └#text: DOM
   │  ├#text:
   │  ├#comment: assets here
   │  └#text:
   ├#text:
   └BODY
      ├#text:
      ├DIV class="container"
      │  ├#text:
      │  ├H3
      │  │  └#text: Hello World!
      │  └#text:
      └#text:
```

https://medium.com/jspoint/understanding-basics-of-dom-265b73d958d1

Svelte

Components

Ideal JavaScript that surgically updates DOM

Compiler

Browser DOM

`</>`

React

Update DOM with the help of Reconciliation diffing algorithm

Diff between previous DOM and updated virtual DOM

Components

Virtual DOM Nodes

Browser JS Engine

https://blog.bitsrc.io/react-vs-sveltejs-the-war-between-virtual-and-real-dom-59cbebbab9e9

Duration in milliseconds ± standard deviation (*slowdown = duration / fastest*)

Name Duration for...	vanillajs	svelte-v3.44.1	vue-v3.2.21	angular-v13.0.0	react-v17.0.1
<u>create rows</u> creating 1,000 rows	77.20.6 (1.00)	92.61.7 (1.20)	99.71.9 (1.29)	113.42.4 (1.47)	116.91.7 (1.51)
replace all rows updating all 1,000 rows (5 warmup runs).	79.20.6 (1.00)	93.40.8 (1.18)	88.90.6 (1.12)	101.91.4 (1.29)	106.01.0 (1.34)
partial update updating every 10th row for 1,000 rows (3 warmup runs). 16x CPU slowdown.	157.12.7 (1.00)	170.53.0 (1.09)	174.719.0 (1.11)	165.91.9 (1.06)	209.31.9 (1.33)
select row highlighting a selected row. (no warmup runs). 16x CPU slowdown.	20.01.1 (1.00)	29.31.1 (1.46)	32.71.0 (1.63)	55.72.8 (2.79)	92.31.8 (4.62)
swap rows swap 2 rows for table with 1,000 rows. (5 warmup runs). 4x CPU slowdown.	47.00.5 (1.00)	48.90.3 (1.04)	47.90.6 (1.02)	324.51.3 (6.91)	320.91.4 (6.83)
remove row removing one row. (5 warmup runs).	19.50.2 (1.00)	20.20.4 (1.03)	21.20.2 (1.09)	20.00.3 (1.02)	22.30.3 (1.14)
create many rows creating 10,000 rows	758.818.1 (1.00)	972.631.4 (1.28)	963.76.8 (1.27)	1,088.018.1 (1.43)	1,342.325.3 (1.77)
append rows to table appending 1,000 to a table of 1,000 rows. 2x CPU slowdown.	170.71.6 (1.00)	205.82.4 (1.21)	197.13.3 (1.15)	230.11.5 (1.35)	240.43.2 (1.41)
clear rows clearing a table with 1,000 rows. 8x CPU slowdown.	49.60.5 (1.00)	70.60.9 (1.42)	66.70.6 (1.35)	139.73.9 (2.82)	77.11.2 (1.55)

https://krausest.github.io/js-framework-benchmark/2022/table_chrome_97.0.4692.71.html

Memory allocation in MBs ± standard deviation

Name	vanillajs	svelte-v3.44.1	vue-v3.2.21	angular-v13.0.0	react-v17.0.1
ready memory Memory usage after page load.	**1.2** (1.00)	**1.2** (1.00)	**1.4** (1.13)	**1.8** (1.45)	**1.4** (1.19)
run memory Memory usage after adding 1000 rows.	**1.2** (1.00)	**2.1** (1.81)	**3.1** (2.66)	**4.0** (3.45)	**4.4** (3.79)
update eatch 10th row for 1k rows (5 cycles) Memory usage after clicking update every 10th row 5 times	**1.2** (1.00)	**2.1** (1.81)	**3.1** (2.68)	**4.1** (3.48)	**4.9** (4.18)
replace 1k rows (5 cycles) Memory usage after clicking create 1000 rows 5 times	**1.3** (1.00)	**2.2** (1.71)	**3.2** (2.49)	**4.4** (3.40)	**4.7** (3.65)
creating/clearing 1k rows (5 cycles) Memory usage after creating and clearing 1000 rows 5 times	**0.9** (1.00)	**1.1** (1.19)	**1.4** (1.53)	**2.4** (2.57)	**1.9** (2.00)
geometric mean of all factors in the table	**1.00**	**1.46**	**1.98**	**2.73**	**2.68**

https://krausest.github.io/js-framework-benchmark/2022/table_chrome_97.0.4692.71.html

SETUP

AND FIRST STEPS

1

Install Node.js

https://nodejs.org/en

Install **LTS** version. Installation follows like always (next, next, next…)

2

Install **npx** (a code runner) globally with **npm** (already included with Node.js) typing in your terminal

npm i -g npx

3

Create your Svelte project using **degit**
(a project scaffolding tool)

npx degit sveltejs/template app-name

Previous step will create a directory
called **app-name**. Go there and type

npm i

and then

npm run dev

You'll see something like this

Your application is ready~! 🚀

- Local: http://localhost:8080
- Network: Add `--host` to expose

Visit that address in your favorite web browser

Easy-peasy. Isn't it? 😘

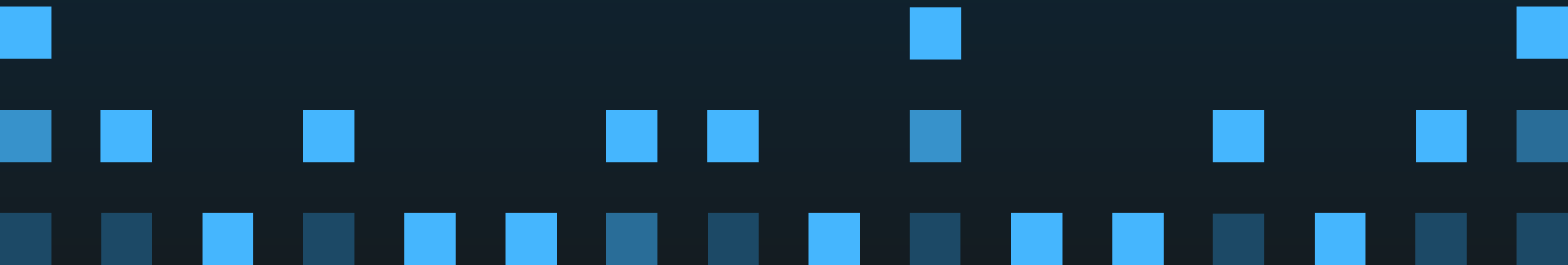# SVELTE: THE BASICS

* Props

* Reactivity

* Events

* Components

* Control Flow

LET'S PRACTICE

THANKS

Daniel Solano

dfsolanol@unal.edu.co