

JavaScript - Ramverk

En snabb introduktion till ramverksdjungeln

Dagens agenda

- Snabb överblick över populära ramverk
- Att bygga en todo-lista på *olika* sätt
- En snabb introduktion till react
- Tid för frågor kring projektet

Orientera er i olika ramverk, olika sätt att skapa webbsidor

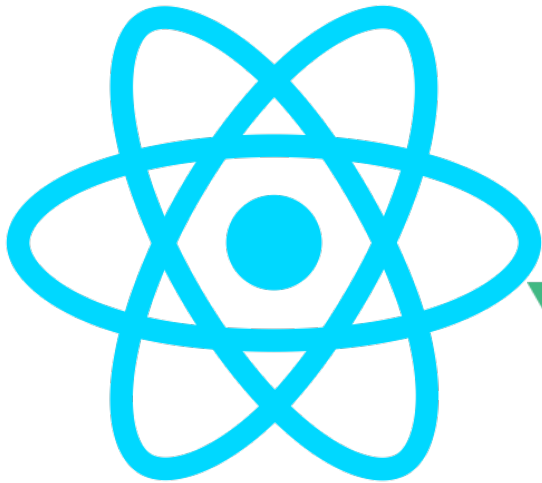


Syfte med dagens föreläsning

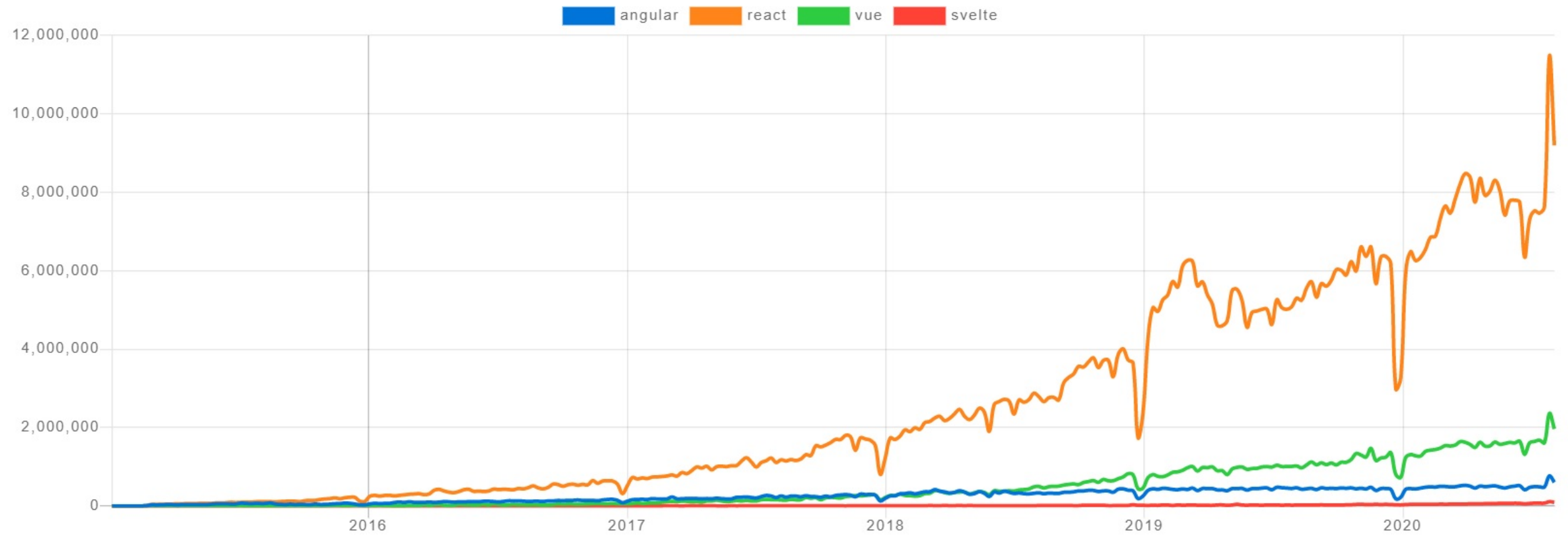
Ramverk

- Slippa återupppfinna hjulet (designmönster)
- Komma igång snabbare (tid = pengar)
- Ändamålsenligt (rätt verktyg)
- Inkluderad funktionalitet (router, data store)
- Koncept: komponenter (parent, child, properties)
- Utvecklingsmiljö

Vilket ramverk?



Downloads in past All time ▾



Front-end Frameworks

As usual React and Vue lead the pack, but Svelte is quickly establishing itself as a very serious contender for the front-end crown.

Rankings

Export

Share

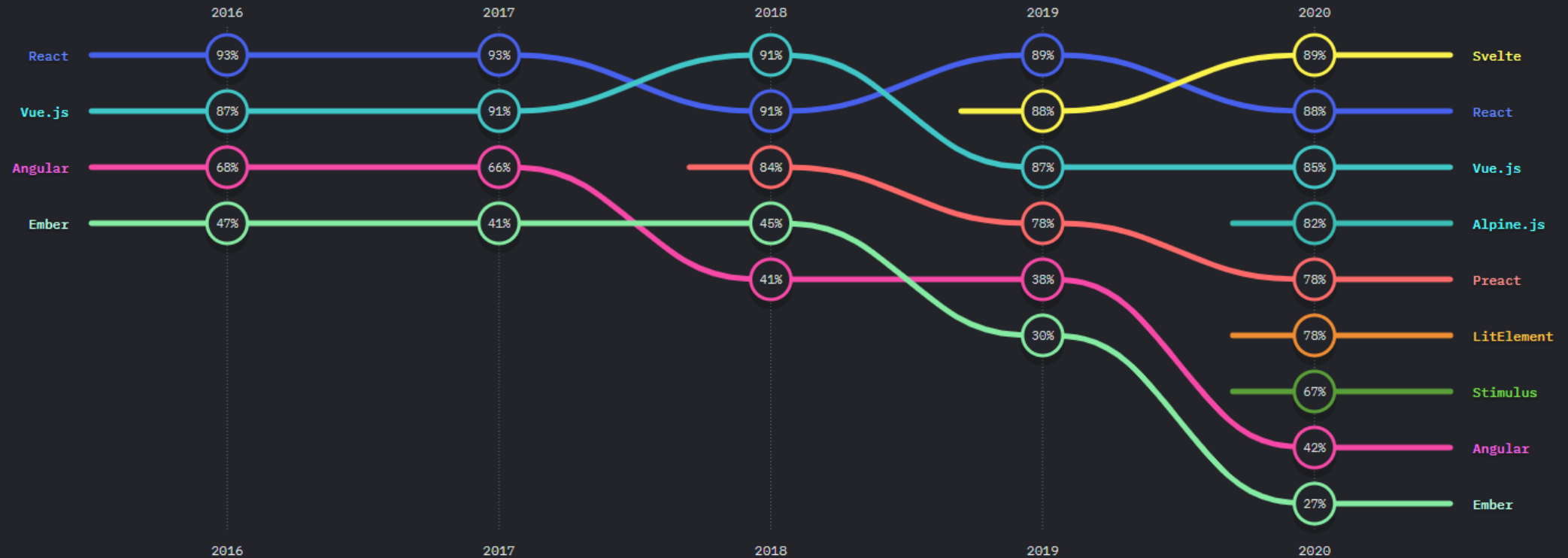
Satisfaction

Interest

Usage

Awareness

Satisfaction, interest, usage, and awareness ratio rankings.



Front-end Frameworks

As usual React and Vue lead the pack, but Svelte is quickly establishing itself as a very serious contender for the front-end crown.

Rankings

Export

Share

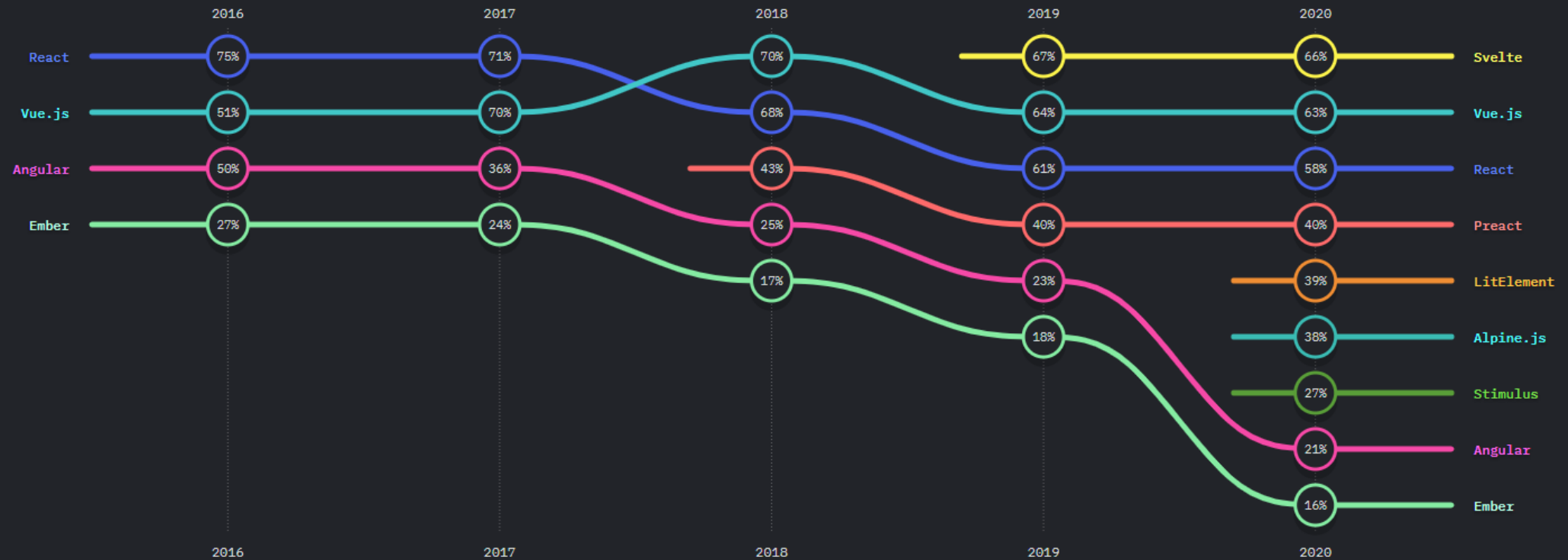
Satisfaction

Interest

Usage

Awareness

Satisfaction, interest, usage, and awareness ratio rankings.



Front-end Frameworks

As usual React and Vue lead the pack, but Svelte is quickly establishing itself as a very serious contender for the front-end crown.

Rankings

Export

Share

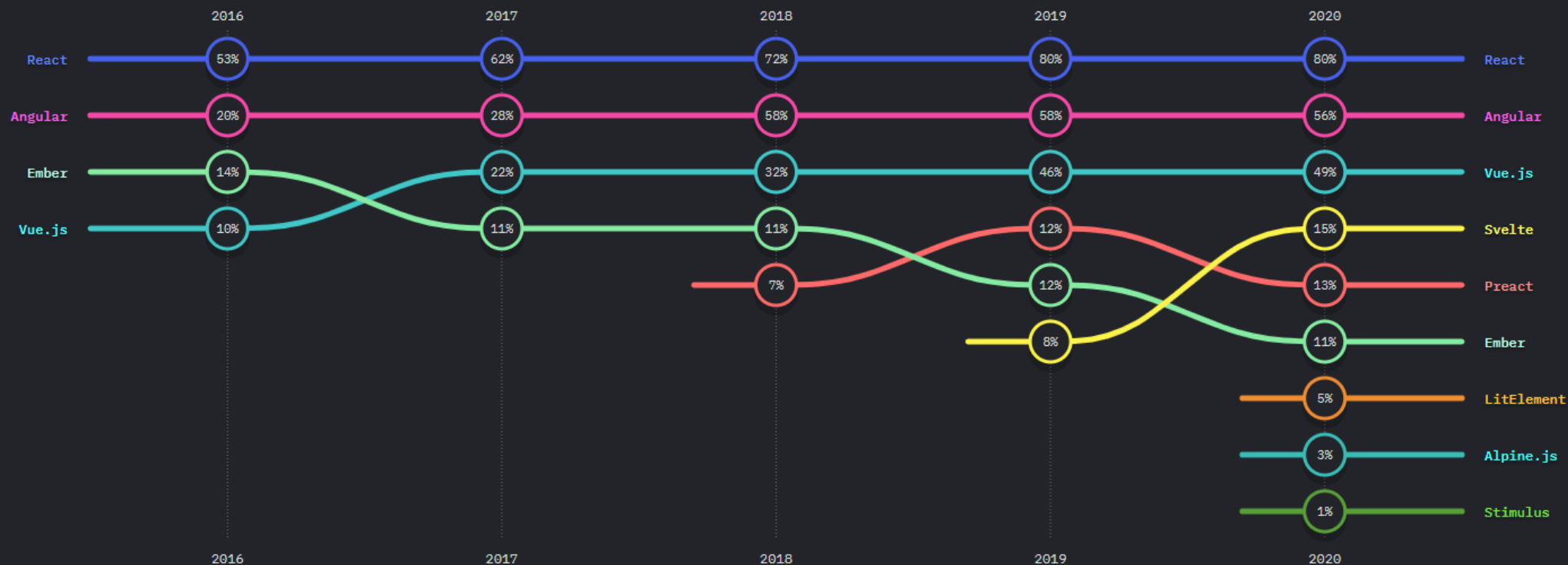
Satisfaction

Interest

Usage

Awareness

Satisfaction, interest, usage, and awareness ratio rankings.



Front-end Frameworks

As usual React and Vue lead the pack, but Svelte is quickly establishing itself as a very serious contender for the front-end crown.

Rankings

Export

Share

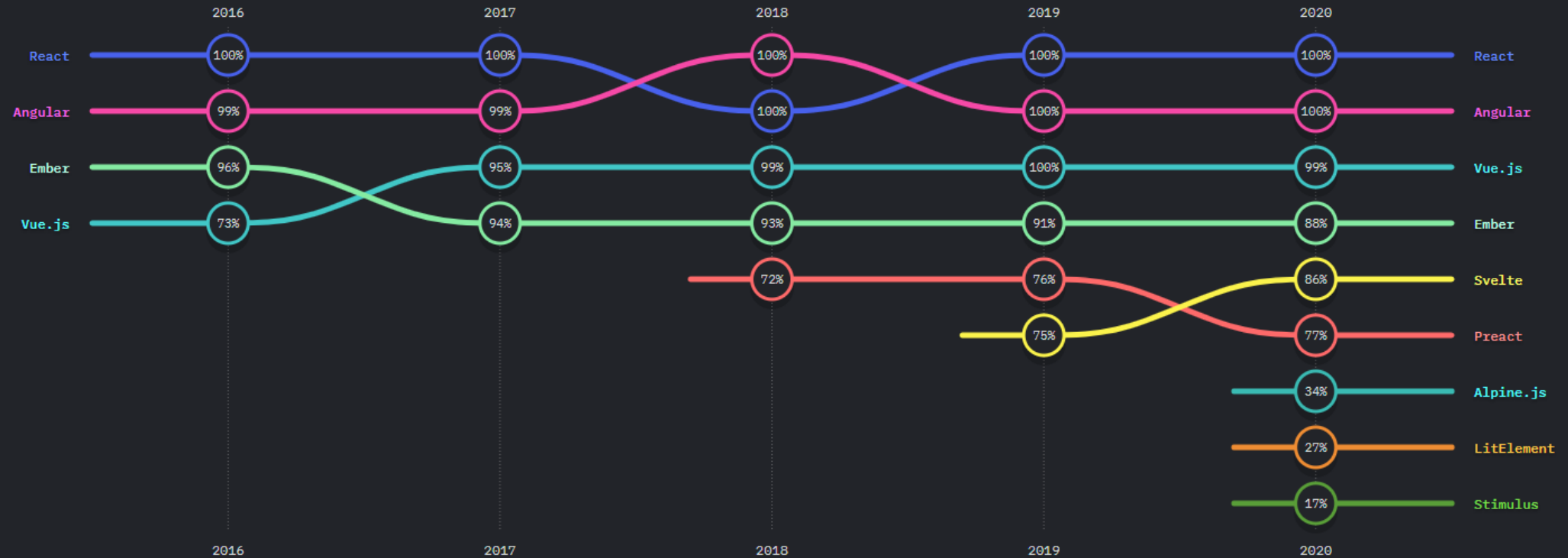
Satisfaction

Interest

Usage

Awareness

Satisfaction, interest, usage, and awareness ratio rankings.



Vad säger andra?

- Angular vs React vs Vue: Which Framework to Choose in 2021
 - <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>
- React vs Angular vs Vue.js – What to choose in 2020?
 - <https://medium.com/techmagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d>

Utvecklingsmiljö?

- <https://nodejs.org/en/download/>
 - **nodejs** (JavaScript utanför webbläsaren)
 - **npm** (pakethanterare för JavaScript)
- Byggsystem
 - <https://webpack.js.org>
 - <https://parceljs.org>

Todo-lista



Dagens exempel vi kommer att prata kring

Welcome to * Todo List!

Todos

First item



1 todo left to do.



Vanilla JS



JS

```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Todos</title>
6     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
7     <link href="style.css" rel="stylesheet">
8   </head>
9   <body>
10    <div class="container">
11      <h1>Welcome to Vanilla JS Todo List!</h1>
12      <h2>Todos</h2>
13      <input type="text" id="new-todo" placeholder="What needs to be done?" class="form-control">
14      <ul id="todos" class="list-group"><ul>
15        <span id="nr-of-todos"></span>
16    </div>
17
18    <script src="script.js"></script>
19  </body>
20 </html>
```

Welcome to
* Todo List!
Todos

What needs to be done?

First item

X

1 todo left to do.


```
1  var todos = [{title: "First item", id: 1}];
2  var highestId = 1;
3
4  document.querySelector("#new-todo").addEventListener("keyup", function(e) {
5      if (event.keyCode == 13) {
6          let todo = {
7              title: event.target.value,
8              id: ++highestId
9          };
10         todos.push(todo);
11         event.target.value = "";
12         renderTodos();
13     }
14 });
15
16 function renderTodos() {
17     document.querySelector("#todos").innerHTML = "";
18     for (let i = 0; i < todos.length; i++) {
19         document.querySelector("#todos").innerHTML += `
20         <li class="list-group-item">
21             <label>${todos[i].title}</label>
22             <button class="btn btn-sm btn-danger" data-id="${todos[i].id}">X</button>
23         </li>
24         `;
25     }
26     let deleteBtns = document.querySelectorAll(".btn-danger");
27     for (let i = 0; i < deleteBtns.length; i++) {
28         deleteBtns[i].addEventListener("click", function(event) {
29
30             todos = todos.filter(todo => todo.id != event.target.getAttribute("data-id"));
```

Welcome to * Todo List!

Todos



1 todo left to do.

```

index.html x JS script.js # style.css
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Todos</title>
6     <link rel="stylesheet" href="https://s
7     <link href="style.css" rel="stylesheet
8   </head>
9   <body>
10    <div class="container">
11      <h1>Welcome to vanilla JS Todo List</h1>
12      <h2>Todos</h2>
13      <input type="text" id="new-todo" r
14      <ul id="todos" class="list-group">
15      <span id="nr-of-todos"></span>
16    </div>

```

```

index.html JS script.js # style.css
1 var todos = [{title: "First item", id: 1}];
2 var highestId = 1;
3
4 document.querySelector("#new-todo").addEventListener("keyup", function(e) {
5   if (event.keyCode === 13) {
6     let todo = {
7       title: event.target.value,
8       id: ++highestId
9     };
10    todos.push(todo);
11    event.target.value = "";
12    renderTodos();
13  }
14 });
15
16 function renderTodos() {
17   document.querySelector("#todos").innerHTML = "";
18   for (let i = 0; i < todos.length; i++) {
19     document.querySelector("#todos").innerHTML += `
20     <li class="list-group-item">
21       <label>${todos[i].title}</label>
22       <button class="btn btn-sm btn-danger" data-id="${todos[i].id}">X</button>
23     </li>
24     `;
25   }
26   let deleteBtns = document.querySelectorAll(".btn-danger");
27   for (let i = 0; i < deleteBtns.length; i++) {
28     deleteBtns[i].addEventListener("click", function(event) {
29
30       todos = todos.filter(todo => todo.id !== event.target.getAttribute("data-id"));
31       renderTodos();
32     });
33   }
34   let todoString = todos.length === 1 ? "todo" : "todos";
35   document.querySelector("#nr-of-todos").innerHTML = `
36   <strong>${todos.length}</strong> ${todoString} left to do.
37   `;
38 }
39
40 renderTodos();

```

Welcome to
* Todo List!
Todos

What needs to be done?

First item

X

1 todo left to do.

Welcome to * Todo List!

Todos

First item



1 todo left to do.

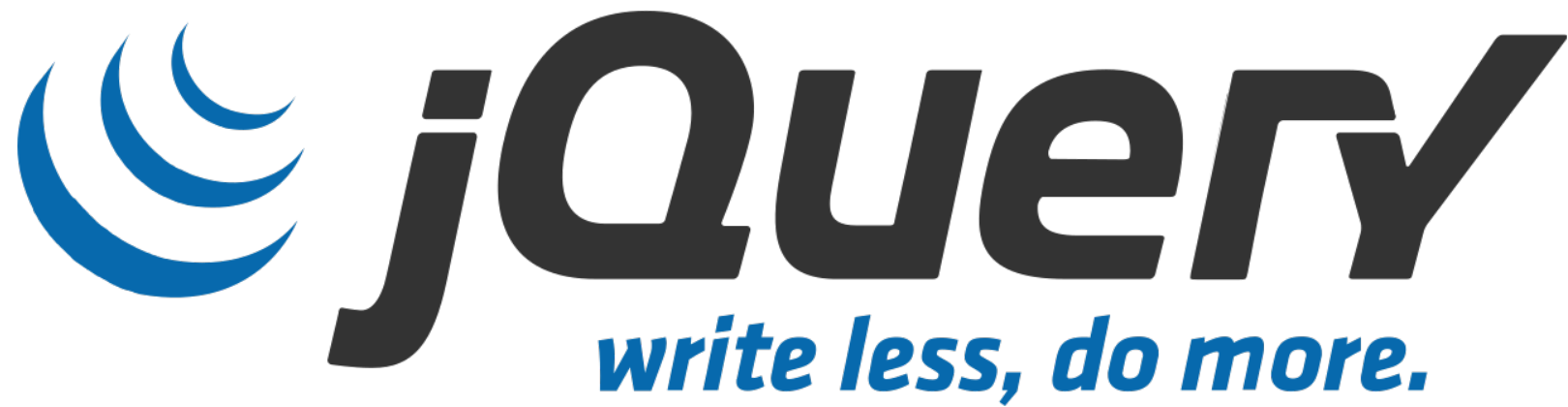
**[http://webshare.mah.se/tsanti/
VT2019/DA355A/todo/js/index
.html](http://webshare.mah.se/tsanti/VT2019/DA355A/todo/js/index.html)**



Todo list - Vanilla JS style

Vanilla JS

- Behöver **ingen** extern kod
- **Större kodbas** (eventuell upprepning)
- Svårt att täcka många **edge** cases
- **Omständigt** att lösa vissa vanliga problem (routing



```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Todos</title>
6     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
7     <link href="style.css" rel="stylesheet">
8   </head>
9   <body>
10    <div class="container">
11      <h1>Welcome to Vanilla JS Todo List!</h1>
12      <h2>Todos</h2>
13      <input type="text" id="new-todo" placeholder="What needs to be done?" class="form-control">
14      <ul id="todos" class="list-group"><ul>
15        <span id="nr-of-todos"></span>
16    </div>
17
18    <script src="script.js"></script>
19  </body>
20 </html>
```

Welcome to
* Todo List!
Todos

What needs to be done?

First item

X

1 todo left to do.

```
1  var todos = [{title: "First item", id: 1}];
2  var highestId = 1;
3
4  $("#new-todo").on("keyup", function(event) {
5      if (event.keyCode == 13) {
6          let todo = {
7              title: event.target.value,
8              id: ++highestId
9          };
10         todos.push(todo);
11         $(this).val("");
12
13         renderTodos();
14     }
15 });
16
17 function renderTodos() {
18     $("#todos").html("");
19     $.each(todos, function(index, todo) {
20         $("#todos").append(`
21         <li class="list-group-item">
22             <label>${todo.title}</label>
23             <button class="btn btn-sm btn-danger" data-id="${todo.id}">X</button>
24         </li>
25         `)
26     });
27
28     let todoString = todos.length == 1 ? "todo" : "todos";
29     $("#nr-of-todos").html(`
30     <strong>${todos.length}</strong> ${todoString} left to do.
```



```

index.html x JS script.js # style.css
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Todos</title>
6     <link rel="stylesheet" href="https://stac
7     <link href="style.css" rel="stylesheet">
8   </head>
9   <body>
10    <div class="container">
11      <h1>Welcome to vanilla JS Todo List!<
12      <h2>Todos</h2>
13      <input type="text" id="new-todo" plac
14      <ul id="todos" class="list-group"></u
15      <span id="nr-of-todos"></span>
16    </div>

```

Welcome to
* Todo List!
Todos

What needs to be done?

First item



1 todo left to do.

```

index.html JS script.js
1 var todos = [{title: "First item", id: 1}];
2 var highestId = 1;
3
4 $("#new-todo").on("keyup", function(event) {
5   if (event.keyCode == 13) {
6     let todo = {
7       title: event.target.value,
8       id: ++highestId
9     };
10    todos.push(todo);
11    $(this).val("");
12
13    renderTodos();
14  }
15 });
16
17 function renderTodos() {
18   $("#todos").html("");
19   $.each(todos, function(index, todo) {
20     $("#todos").append(`
21       <li class="list-group-item">
22         <label>${todo.title}</label>
23         <button class="btn btn-sm btn-danger" data-id="${todo.id}">X</button>
24       </li>
25     `);
26   });
27
28   let todoString = todos.length == 1 ? "todo" : "todos";
29   $("#nr-of-todos").html(`
30     <strong>${todos.length}</strong> ${todoString} left to do.
31   `);
32 }
33
34 $("body").on("click", ".btn-danger", function() {
35   todos = todos.filter(todo => todo.id != $(this).attr("data-id"));
36   renderTodos();
37 });
38
39 renderTodos();

```

Welcome to * Todo List!

Todos

First item



1 todo left to do.

**[http://webshare.mah.se/tsanti/
VT2019/DA355A/todo/jquery/i
ndex.html](http://webshare.mah.se/tsanti/VT2019/DA355A/todo/jquery/index.html)**

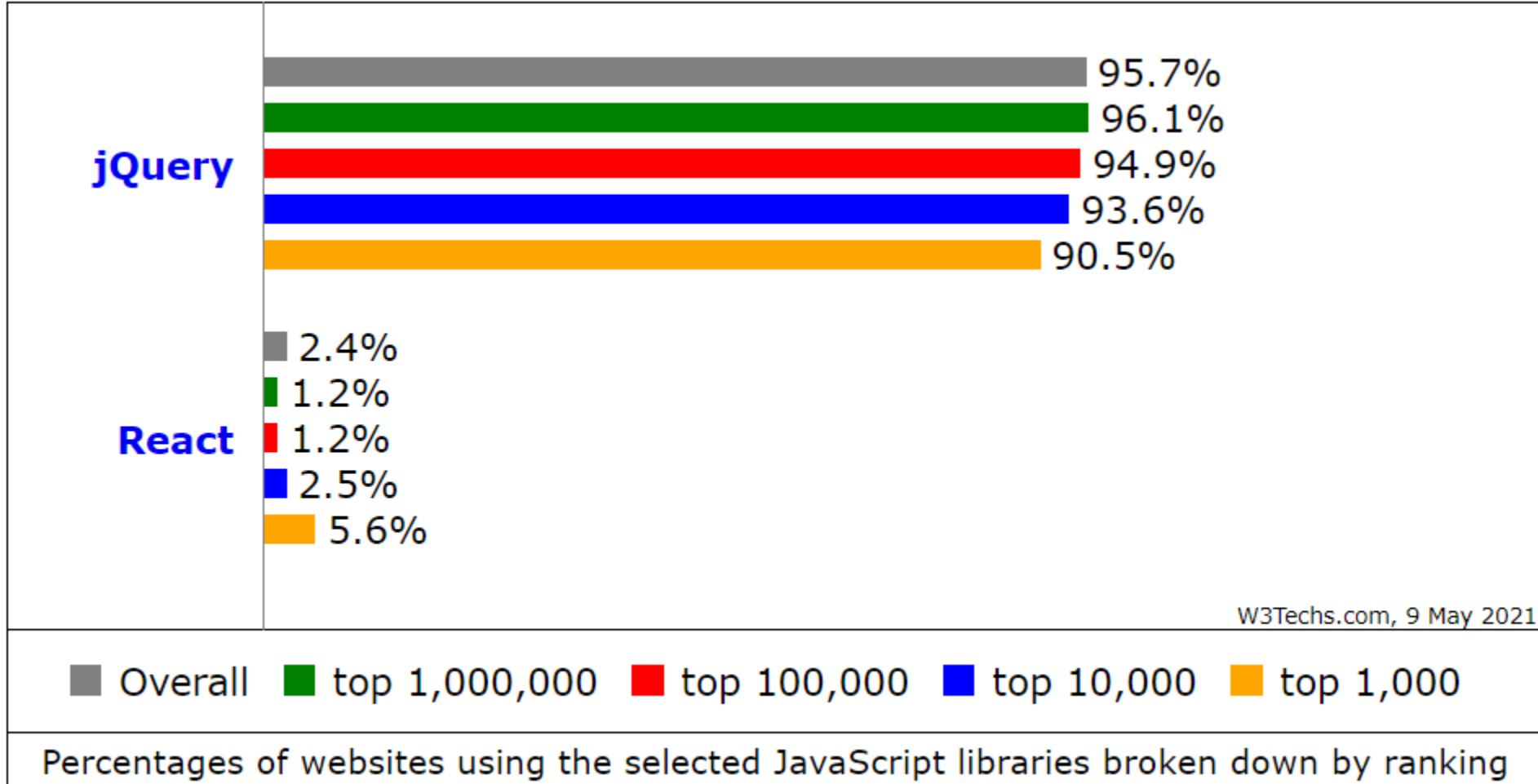


Todo list - jQuery style

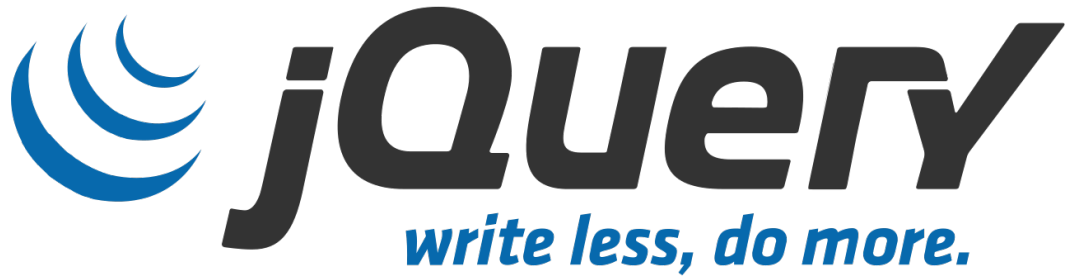
PIECE OF CAKE



MMMMM CAKE

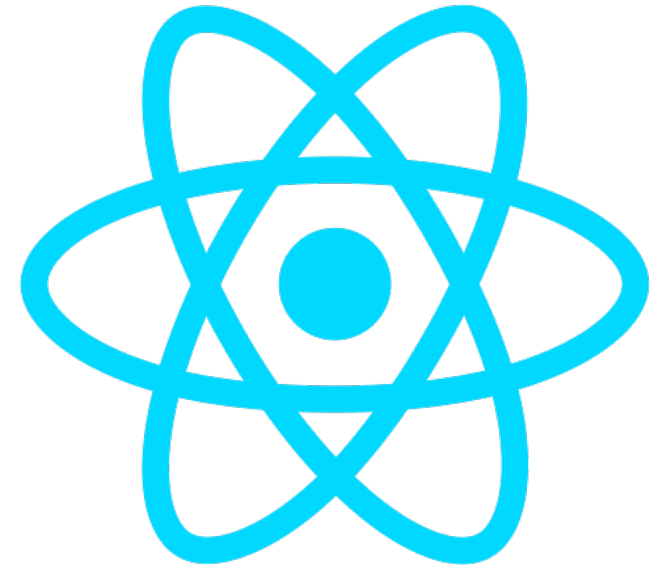


<https://w3techs.com/technologies/comparison/js-jquery,js-react>



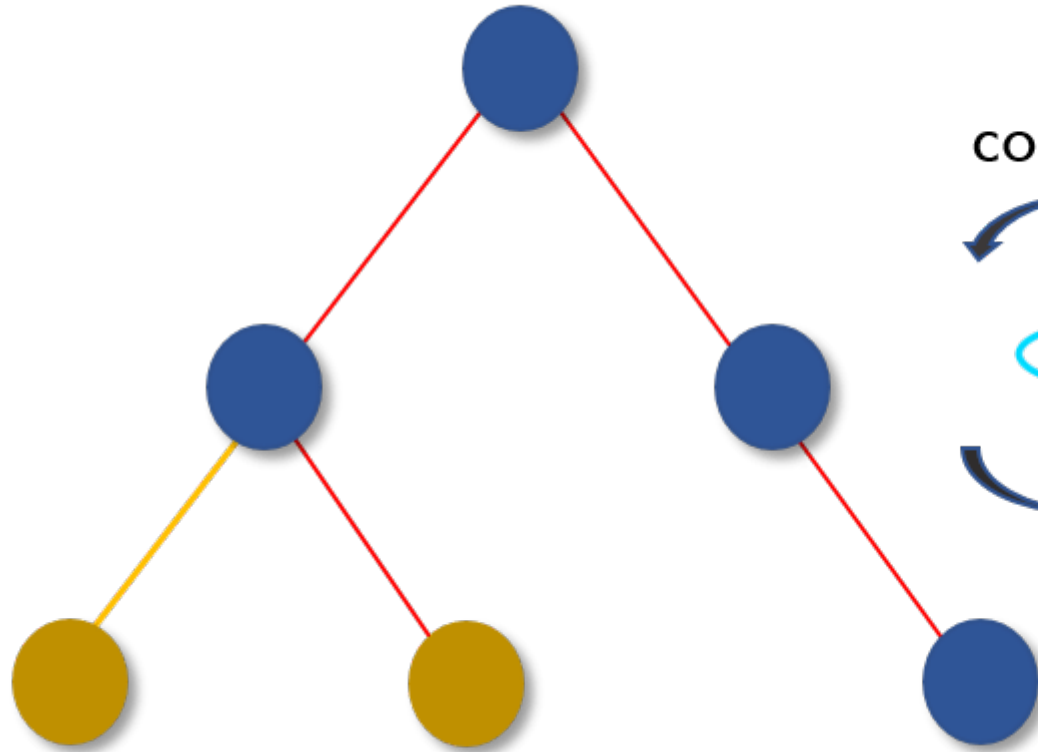
- **Ett bibliotek**
- **Större kodbas** (men mindre än Vanilla JS)
- jQuery täcker en del **edge cases**
- Stort **community** (enkelt att hitta guider)
- Många **plugins** för vanliga problem

Varför använder vi JS-ramverk?

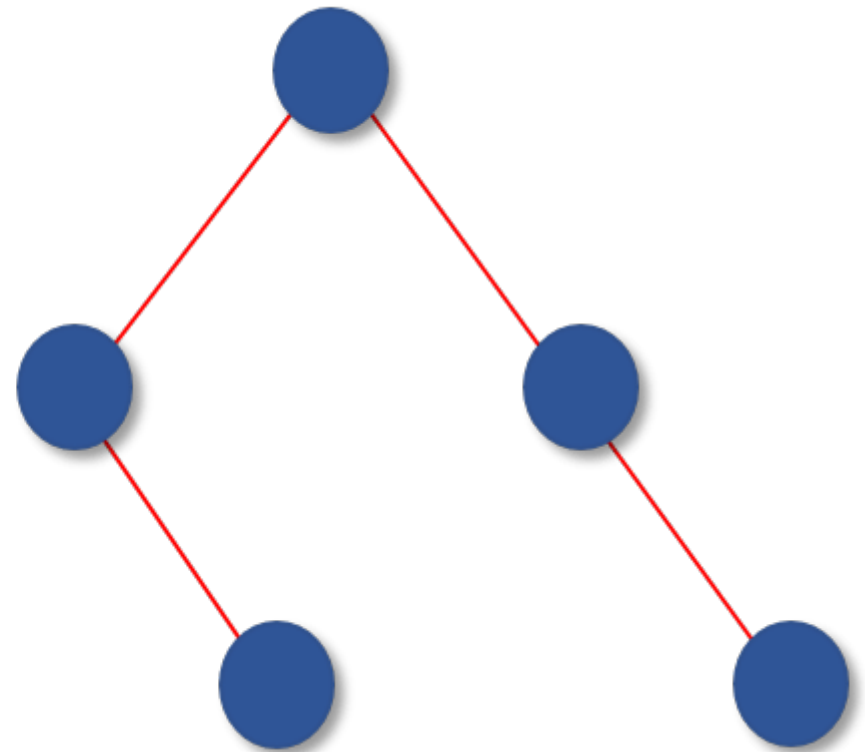


**KEEPING THE UI
IN SYNC
WITH THE STATE
IS HARD**

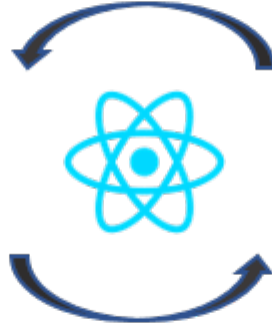
Virtual DOM



Real DOM

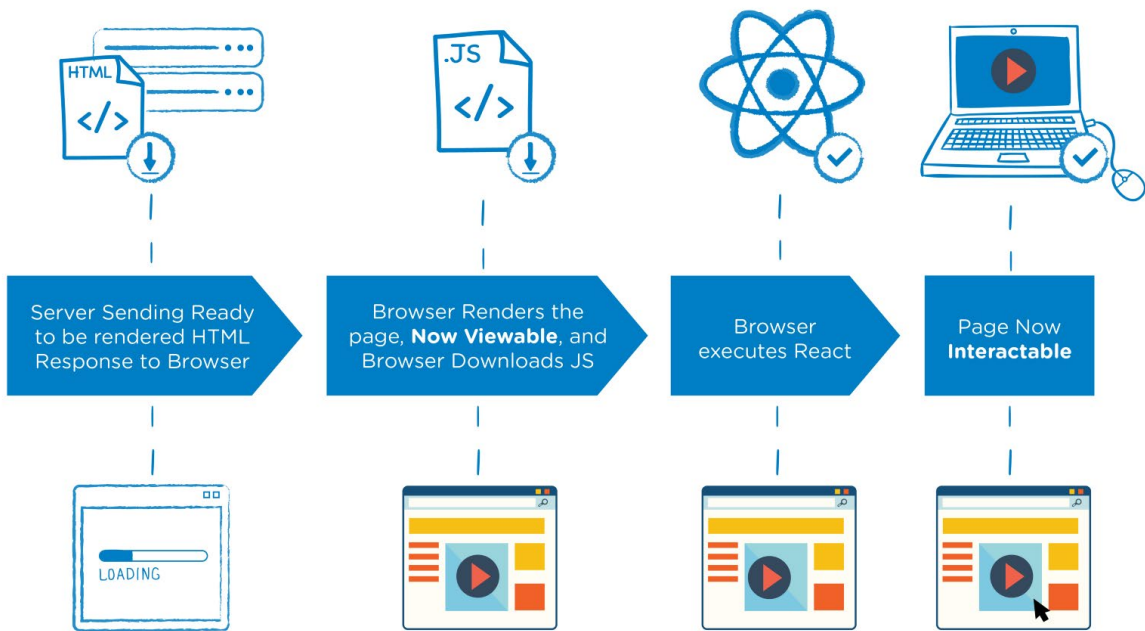


compare

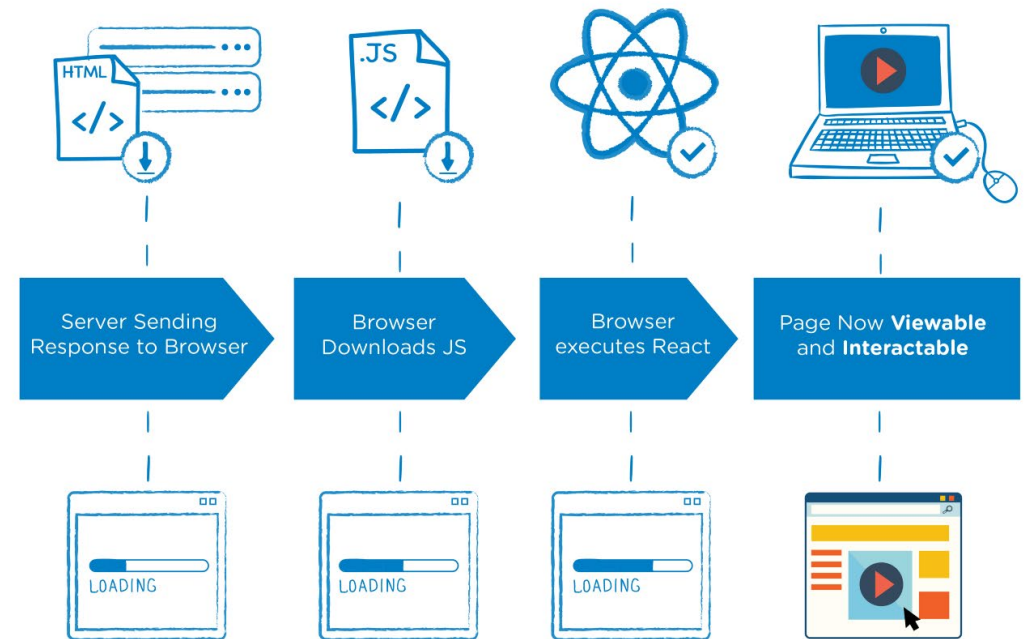




SSR



CSR



Att använda ett ramverk vs. ett bibliotek

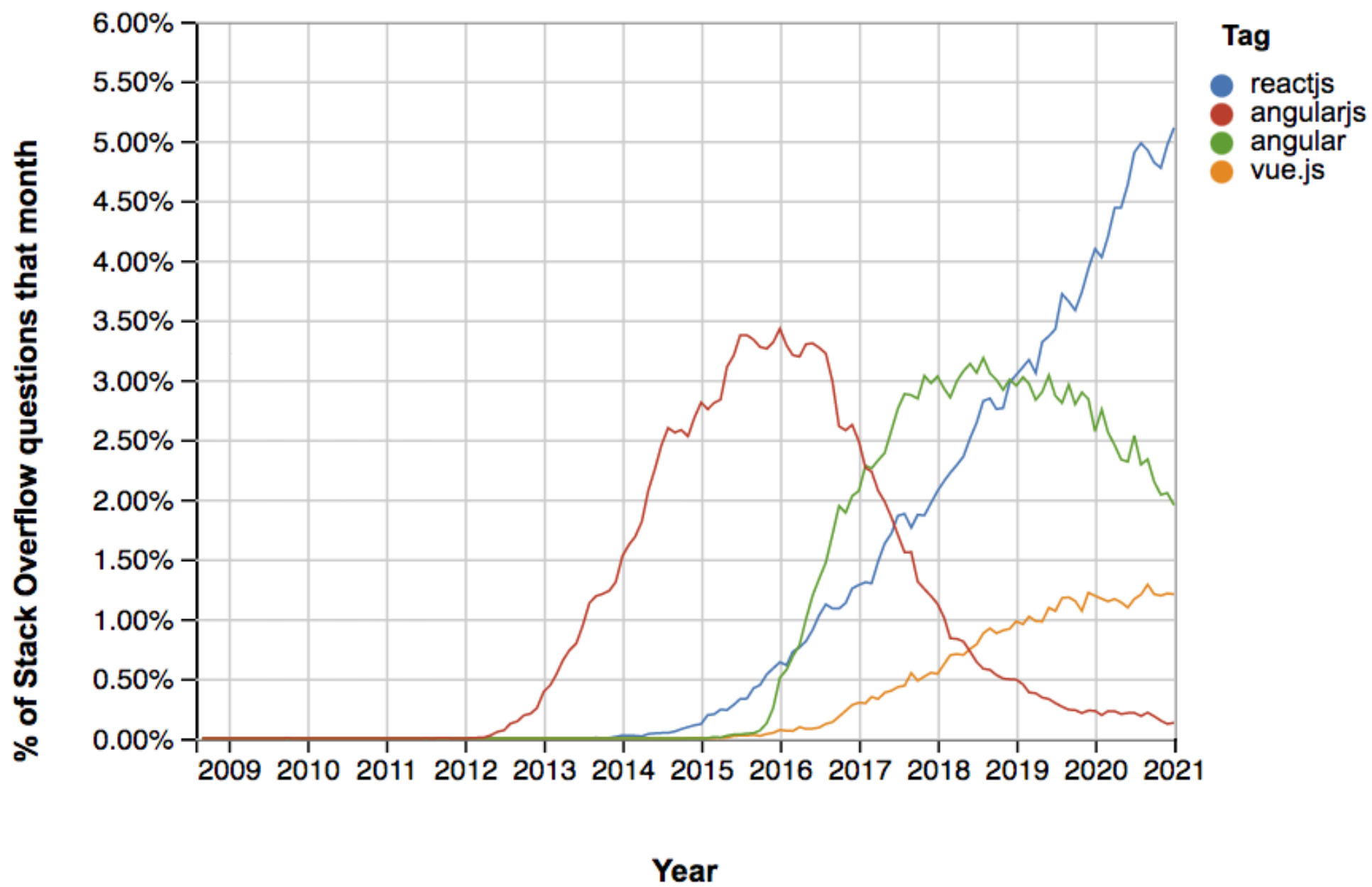


jQuery.js

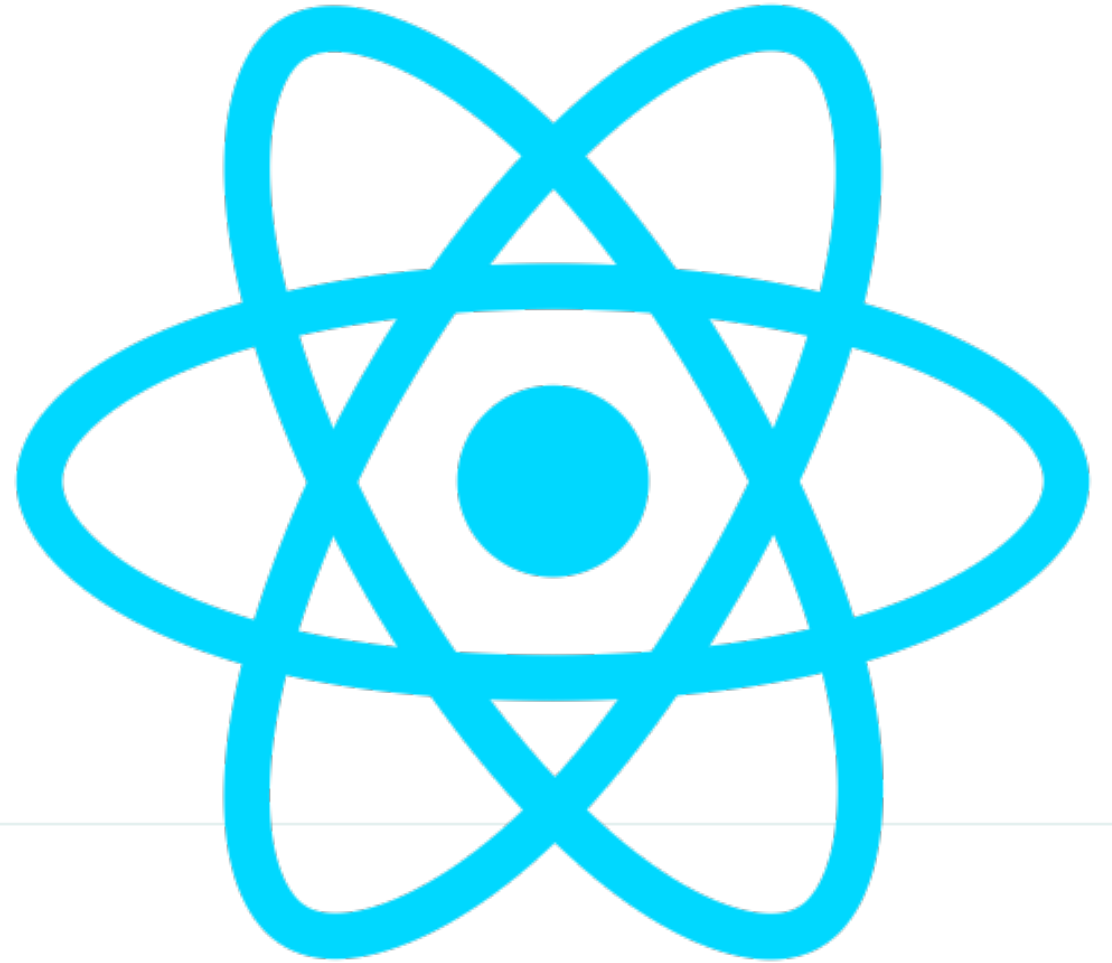


```
▸ dist
▸ e2e
▸ node_modules
▸ src
⚙ .editorconfig
🔍 .gitignore
{} angular.json
{} package-lock.json
{} package.json
📖 README.md
{} tsconfig.json
{} tslint.json
```

	Angular	React	Vue
Initial release	2010	2013	2014
Official site	angular.io	reactjs.org	vuejs.org
Current version	11	17.x	3.0.x
Used by	Google, Wix	Facebook, Uber	Alibaba, GitLab



Hur kommer man igång med react



Create React App

Create React App is a comfortable environment for **learning React**, and is the best way to start building **a new single-page application** in React.

It sets up your development environment so that you can use the latest JavaScript features, provides a nice developer experience, and optimizes your app for production. You'll need to have Node ≥ 6 and npm ≥ 5.2 on your machine. To create a project, run:

```
npx create-react-app my-app  
cd my-app  
npm start
```


Komponenten

Welcome to Angular Todo List!

Todos

What needs to be done?

First item

x

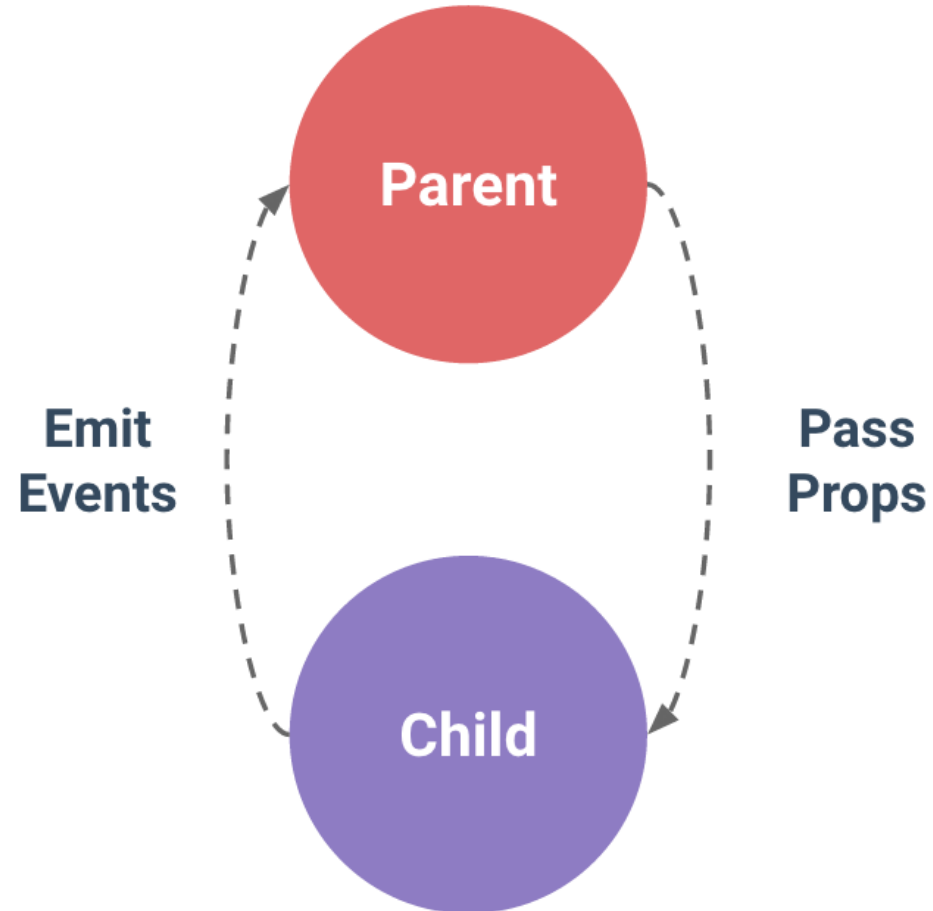
1 todo left to do.

App

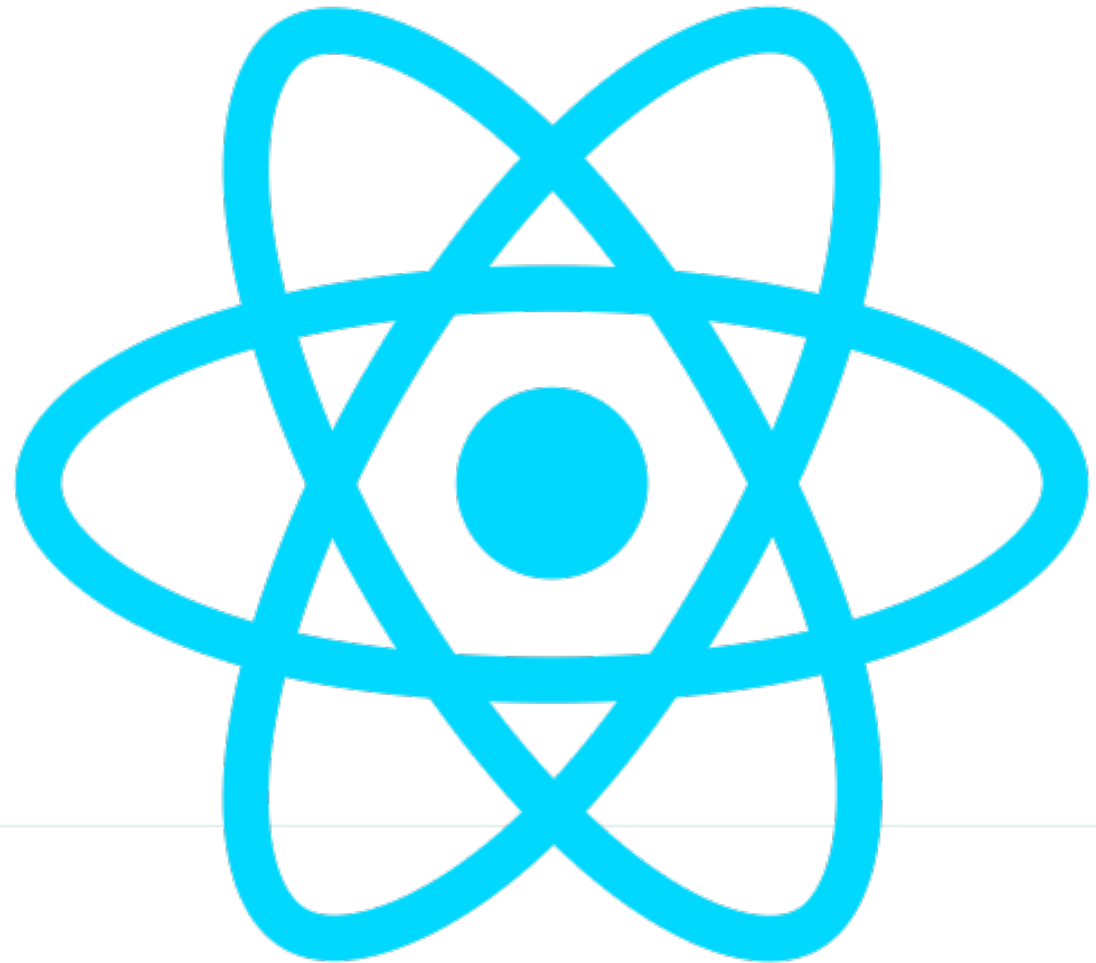
TodoList

Todo

Komponenter?



Todo-lista med React



React

- **Ett bibliotek** (dvs. allt är inte inkluderat)
- Inkluderar inte en lösning för **routing** och **data store**
- Arbetar med **komponenter** (projektstruktur)
- Koncept: **Virtual DOM, JSX**
- Byggsystem **i terminalen** rekommenderas
 - <https://create-react-app.dev>

Hur kommer man igång med Angular?



Installera nödvändiga komponenter

1. Installera Node.js
 - <https://nodejs.org/en/download/>
2. Installera NPM
 - <https://www.npmjs.com/get-npm>
3. Installera Angular CLI
 - <https://cli.angular.io/>



Lär känna terminalen, nyckeln till framgång

- Skapa ett nytt Angular-projekt
`ng new projektnamn`
- Navigera till det nyskapade Angular-projektet
`cd projektnamn`
- Starta Angular-projektet
`ng serve`
- Skapa en klass i Angular-projektet
`ng generate class Todo`
- Skapa en komponent-klass i Angular-projektet
`ng generate component TodoList`

Welcome to * Todo List!

Todos

First item



1 todo left to do.

Komponenter

Welcome to Angular Todo List!

Todos

What needs to be done?

First item

x

1 todo left to do.

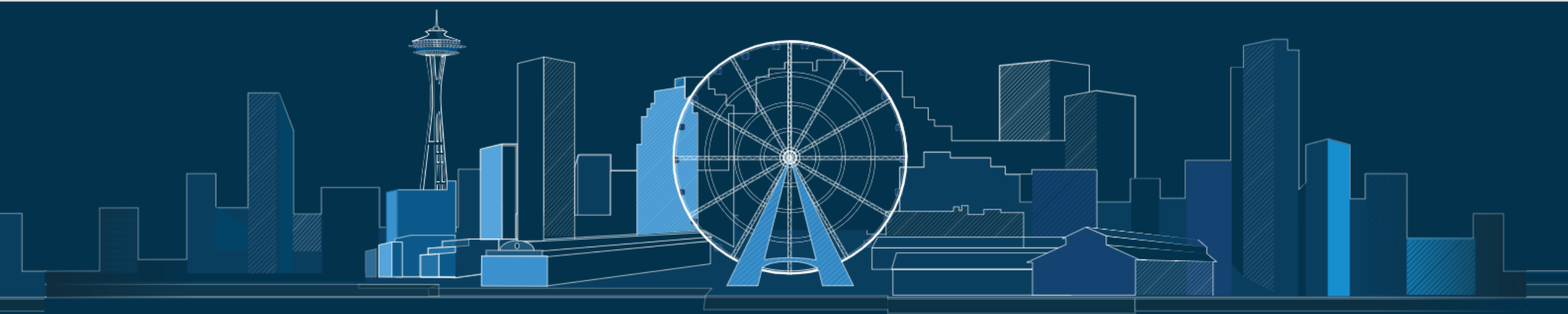
app.component

todo-list.component

todo.ts

```
1  export class Todo {  
2      id: number;  
3      title: string = '';  
4      complete: boolean = false;  
5  
6      constructor(values: Object = {}) {  
7          Object.assign(this, values);  
8      }  
9  }  
10
```

TypeScript 3.4 is now available. [Download](#) our latest version today!



TypeScript

JavaScript that scales.

TypeScript is a typed superset of JavaScript that compiles to plain JavaScript.

Any browser. Any host. Any OS. Open source.

[Download](#)[Documentation](#)

#iHeartTypeScript

Exempel på TypeScript

```
interface Person {  
    firstName: string;  
    lastName: string;  
}  
  
function greeter(person: Person) {  
    return "Hello, " + person.firstName + " " + person.lastName;  
}  
  
let user = { firstName: "Jane", lastName: "User" };  
  
document.body.innerHTML = greeter(user);
```



Todo-lista med Angular

Welcome to * Todo List!

Todos

What needs to be done?

First todo

X



Angular
Hur kan det se ut?

todo.ts

```
1  export class Todo {  
2      id: number;  
3      title: string = '';  
4      complete: boolean = false;  
5  
6      constructor(values: Object = {}) {  
7          Object.assign(this, values);  
8      }  
9  }  
10
```

TS todo-list.component.ts

todo-list.component.html

```
1 <h2>Todos</h2>
2 <input placeholder="What needs to be done?"
3   class="form-control" [(ngModel)]="newTodo.title"
4   (keyup.enter)="addTodo()">
5 <ul class="list-group" *ngFor="let todo of todos">
6   <li class="list-group-item">
7     <label>{{ todo.title }}</label>
8     <button class="btn btn-sm btn-danger"
9       (click)="removeTodo(todo)"></button>
10  </li>
11 </ul>
12 <footer>
13   <strong>{{ todos.length }}</strong>
14   {{ todos.length == 1 ? "todo" : "todos" }} left to do.
15 </footer>
```

TS todo-list.component.ts

```
1 import { Component, OnInit } from '@angular/core';
2 import { Todo } from '../todo';
3
4 @Component({
5   selector: 'app-todo-list',
6   templateUrl: './todo-list.component.html',
7   styleUrls: ['./todo-list.component.css']
8 })
9 export class TodoListComponent implements OnInit {
10
11   lastId: number = 1;
12   todos: Todo[] = [];
13   newTodo: Todo = new Todo();
14
15   constructor() {
16     this.todos.push(new Todo({
17       id: 1,
18       title: 'First item',
19       completed: false
20     }));
21   }
22
23   ngOnInit() {
24   }
25
26   addTodo() {
27     let todo = this.newTodo;
28     if (!todo.id) {
29       todo.id = ++this.lastId;
30     }
31     this.todos.push(todo);
32     this.newTodo = new Todo();
33   }
34
35   removeTodo(todo) {
36     this.todos = this.todos.filter(t => t.id !== todo.id);
37   }
38
39 }
40
```


Angular

- **Ett ramverk**
 - Inkluderar funktionalitet för: routing, data store, m.m.
- **Kräver ett byggsystem** för terminalen
- Avancerat/svårt att komma igång med
- Stort **community** (funnits länge, sedan 2010)
- **TypeScript** (Typer till JavaScript)
 - <https://www.typescriptlang.org>



Hur kommer man igång med vue.js



Simply download and include with a script tag. `Vue` will be registered as a global variable.



Don't use the minified version during development. You will miss out on all the nice warnings for common mistakes!

Development Version

With full warnings and debug mode

Production Version

Warnings stripped, 33.30KB min+gzip

CDN

For prototyping or learning purposes, you can use the latest version with:

```
<script src="https://cdn.jsdelivr.net/npm/vue"></script>
```

HTML

For production, we recommend linking to a specific version number and build to avoid unexpected breakage from newer versions:

```
<script src="https://cdn.jsdelivr.net/npm/vue@2.6.10/dist/vue.js"></script>
```

HTML

Komponenten

Welcome to Angular Todo List!

Todos

What needs to be done?

First item

X

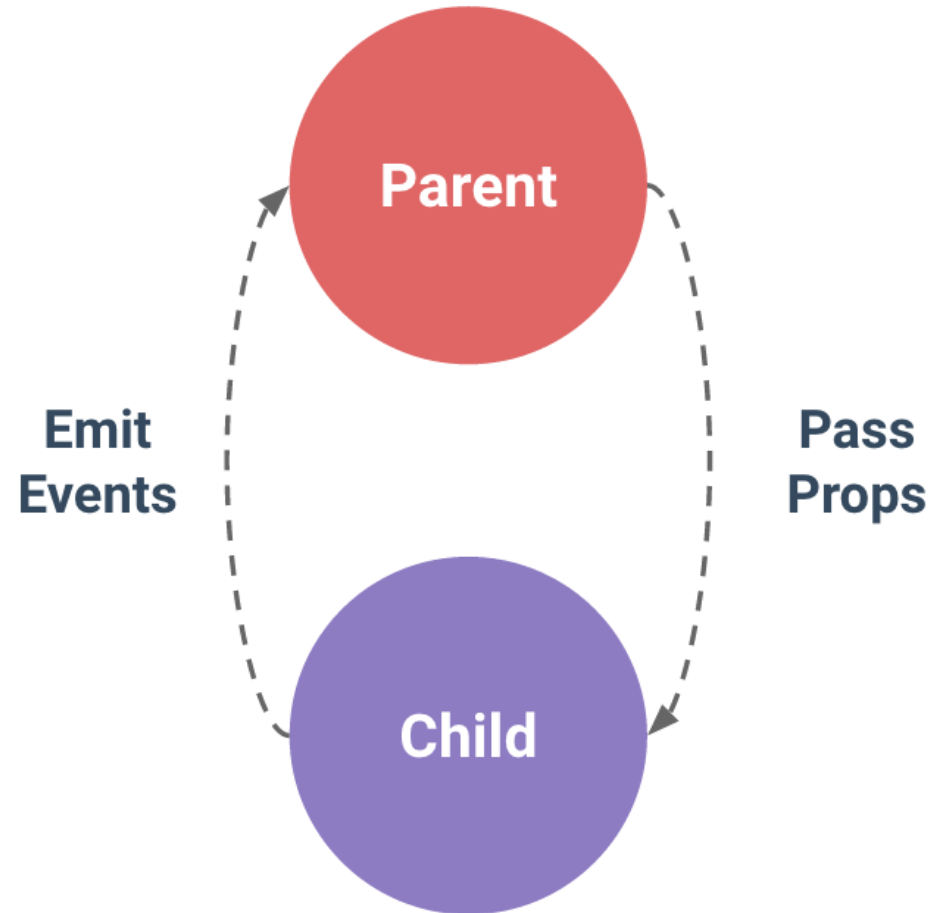
1 todo left to do.

```
<app>
```

```
<todolist>
```

```
<todo>
```

Komponenter?





Todo-lista med Vue



☰ index.html ×

JS script.js

style.css

```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Todos</title>
6     <link rel="stylesheet" href="https://stackpath.
7       integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1f
8     <link href="style.css" rel="stylesheet">
9   </head>
10  <body>
11    <div class="container" id="app">
12      <h1>Welcome to {{ title }} Todo List!</h1>
13      <h2>Todos</h2>
14      <todolist></todolist>
15    </div>
16    <script src="https://cdn.jsdelivr.net/npm/vue/d
17    <script src="script.js"></script>
18  </body>
19 </html>
20
```

☰ index.html × JS script.js × # style.css

```
1 Vue.component('todo', {
2   props: {
3     item: {
4       type: Object,
5       required: true
6     },
7     deleteItem: {
8       type: Function,
9       required: true
10    }
11  },
12  template: `<li class="list-group-item">
13    {{ item.title }}
14    <button class="btn btn-sm btn-danger" v-on:click="deleteItem(item.id)">X</button>
15  </li>`,
16  data() {
17    return {}
18  },
19  });
20
21 Vue.component('todolist', {
22   template: `<div>
23     <input type="text" placeholder="What needs to be done?" class="form-control" v-on:keyup.enter="addItem" :value="newItem">
24     <ul class="list-group">
25       <todo v-for="todo in todos" :key="todo.id" :item="todo" :deleteItem="deleteItem"></todo>
26     </ul>
27     <span>
28       <strong>{{ todos.length }}</strong> {{ nrOfItems }} left to do.
29     </span>
30   </div>`,
31   computed: {
32     nrOfItems() {
33       return this.todos.length == 1 ? "todo" : "todos";
34     }
35   },
36   methods: {
37     addItem(event) {
38       let todo = {
39         title: event.target.value,
40         id: ++this.highestId
41       };
42       this.todos.push(todo);
43       this.newItem = "";
44     },
45     deleteItem(id) {
46       this.todos = this.todos.filter(todo => todo.id != id);
47     }
48   },
49   data() {
50     return {
51       highestId: 0,
52       newItem: "",
53       todos: [{
54         id: 0,
55         title: "First item"
56       }]
57     }
58   },
59   });
60
61 var app = new Vue({
62   el: "#app",
63   data: {
64     title: "Vue",
65   }
66 });
67
```

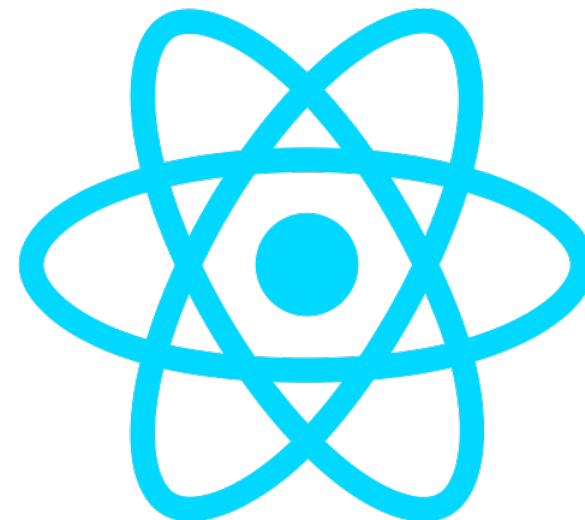
Vue

- **Ett ramverk**
 - Inkluderar funktionalitet för: **routing**, **data store**, m.m.
- Kan snabbt komma igång (med hjälp av CDN)
- **Eget byggsystem** för terminalen
- Stort **community**
- **Liten** filstorlek

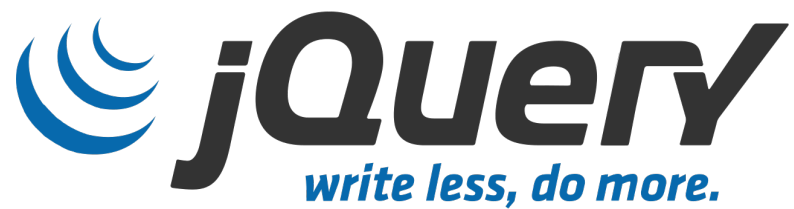


SVELTE

- Ett (**nytt**) ramverk
- **Kompilerar din kod** till färdig HTML/CSS/JS
- Arbetar med **komponenter**
 - Varje komponent består av HTML/CSS/JS



Frågor?



Kursens projekt

TEAMWORK



**THAT'S HOW
WE DO IT**

Skapa en webbapplikation genom ett JS-ramverk

T.ex. React, Angular, Vue, Ember, etc.



Projekt

Projektet ska göras i grupper om **tre (3) personer**. Ni anmäler er projektgrupp via Canvas på den här länken: <https://mau.instructure.com/courses/8054/groups#tab-8222>. Anmäl er grupp **senast den 12:e Maj (23:59)** - det är **superviktigt** att ni gör detta i tid, så att ni får redovisa projektet vid redovisningstillfällena i slutet av kursen

Hittar ni ingen projektgrupp så skriv (i god tid) i kanalen **#hitta-projektgrupp** på Discord för att hitta andra studenter som också vill hitat en projektgrupp.

Skulle ni inte lyckas att skapa en grupp innan deadline så **skapa en grupp med bara en person i (du)** - så kommer vi att slå ihop alla “en”-grupper till “tre”-grupper senast den 14:e maj.

Kravspecifikation

- Ni ska bygga en webbapplikation baserat på ett ramverk/bibliotek (t.ex. React, Vue, Angular, m.m.).
- Ni ska motivera ert val av detta ramverk/bibliotek genom en jämförelse med två andra (t.ex. om ni valt React, varför valde ni inte Vue eller Angular?).
- Er webbapplikation ska använda sig av **minst** ett externt API (som tidigare **inte** använts i laborationerna eller uppgifterna).
- Er webbapplikation ska antingen göra det möjligt för en användare att kunna spara data, eller att webbapplikationen sparar data på egen hand, i relation till det valda API:et.
 - Till exempel ett filmbibliotek genom ett API för filmer.
 - Till exempel en lista över de senaste vädersökningarna genom ett API för väderdata.
- Er webbapplikation ska ha ett grafiskt gränssnitt med fokus på användbarhet på olika enheter (dvs. vara responsiv).
- Ni ska använda er av Git i kombination med GitHub för att versionshantera ert projekt.
- Ni ska skapa en sk. **README** på GitHub där ni kort beskriver ert projekt **och** inkluderar jämförelsen mot de andra ramverken/biblioteken.



Är ni osäkra på er projektidé kan ni lämpligen stämma av med oss på ett tillfälle för projekthandledning eller en laboration.

Projekthandledning

Projekthandledning kommer ske på ett snarlikt vis som våra laborationer (samma Zoom-rum) med ett viktigt undantag:

- Ni ska komma dit med hela er grupp och vara förberedda med frågor. Därefter kommer vi använda oss av break-out-rooms för att hjälpa respektive grupp.

Redovisning

Varje projektgrupp kommer få cirka 10 minuter (plus 5 minuter för frågor) för att presentera sitt projekt. Försök att utnyttja tiden på bästa sätt. Ni kan utgå från följande punkter till presentationen:

- Introduktion till vad ni valt att bygga och varför.
- Vilket ramverk/bibliotek ni valt och varför.
- Vilka API:er ni använder av.
- En kort demonstration av er webbapplikation.
- En kort reflektion över hur arbetet har gått och vilka lärdomar ni tar med er från detta.

På det tillfälle (onsdag eller torsdag, v.22) ni redovisar så förväntas det att ni stannar hela tillfället.



Veckan innan presentationerna kommer vi att publicera en Doodle där varje projektgrupp kan anmäla sig till ett redovisningstillfälle (**glöm inte** att ange nummer för er projektgrupp).

V.	Måndag	Tisdag	Onsdag	Torsdag	F	L	Söndag
13	F: Kursintroduktion	F: Repetition HTML/CSS (Sys) F: Repetition JS (IA)		L: Repetition HTML/CSS (Sys) L: Repetition JS (IA)			
14		F: JavaScript & DOM		L: JavaScript & DOM			D: Inl. 1
15	F: RWD & CSS-ramverk	D: Peer-review (Inl. 1.b)		L: RWD & CSS-ramverk			D: Inl. 2
16	F: jQuery & Ajax	D: Peer-review (Inl. 2.b)		L: jQuery & Ajax			D: Inl. 3
17	F: Store & Locate	D: Peer-review (Inl. 3.b)		L: Store & Locate			D: Inl. 4
18	F: Utvecklingsmetodik	D: Peer-review (Inl. 4.b)		L: Utvecklingsmetodik			D: Inl. 5
19	F: JS-Ramverk		L: JS-Ramverk				D: Inl. 6
20	Projekthandledning		Projekthandledning				
21	Projekthandledning		Projekthandledning				
22			Projektredovisning	Projektredovisning			D: Projekt



Frågor kring projektet?



SVELTE

