



Orientera er i olika ramverk, olika sätta att skapa webbsidor

Syfte med dagens föreläsning

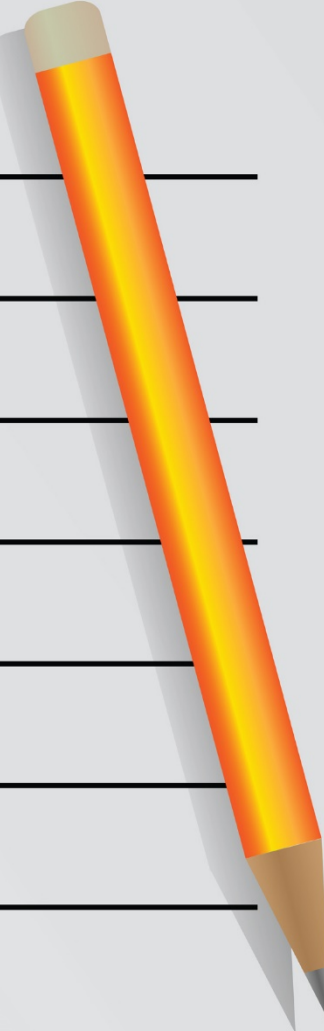
Dagens föreläsning

- Vanilla JS
- jQuery <http://jquery.com>
- Angular <https://angular.io/>
- Vue.js <https://vuejs.org/>
- React <https://reactjs.org/>

A large yellow square containing the letters 'JS' in a bold, black, sans-serif font, representing JavaScript.

To Do List:

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____



Welcome to * Todo List!

Todos

What needs to be done?

First todo

X

Second Todo

X

Third todo

X

3 todos left to do.

Funktioner i vår Todo-lista

- Visa alla todos
- Lägg till en todo
- Ta bort en todo
- Vissa antal todos som är kvar att göra



Welcome to * Todo List!

Todos

What needs to be done?	
First todo	<input type="checkbox"/>
Second Todo	<input type="checkbox"/>
Third todo	<input type="checkbox"/>

3 todos left to do.

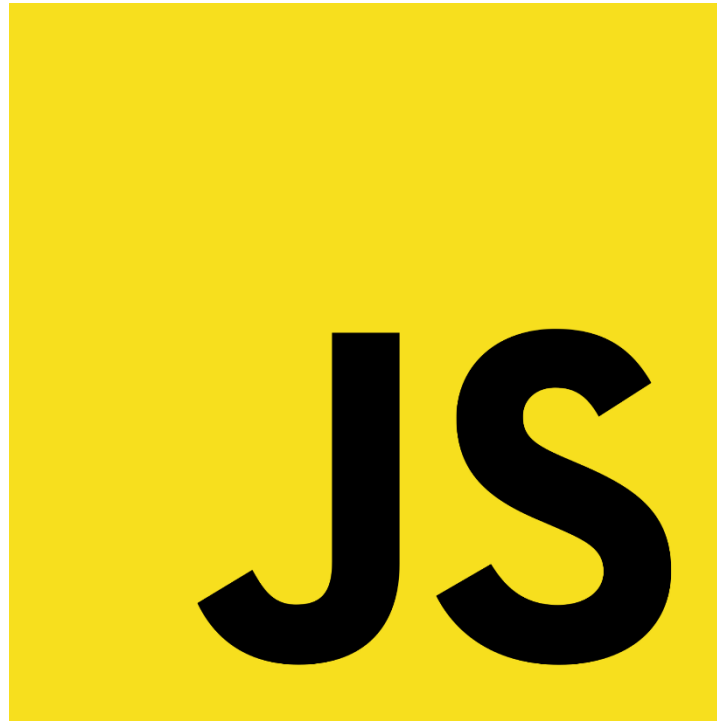
Welcome to * Todo List!

Todos

First item




1 todo left to do.



Vanilla 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>
```



```
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"));
```

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 App</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>
17
18    <script src="script.js"></script>
19  </body>
20 </html>
```

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

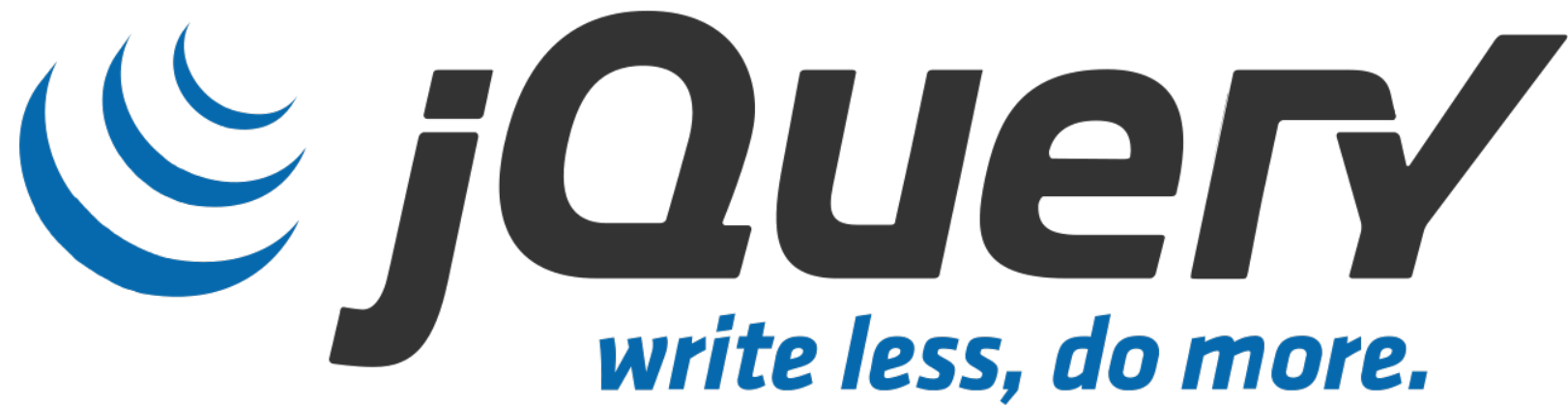
First item




1 todo left to do.

<http://webshare.mah.se/tsanti/VT2019/DA355A/todo/js/index.html>

Todo list – Vanilla JS style



```
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>
```




```
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>
17
18     <script src="script.js"></script>
19   </body>
20 </html>
```

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/index.html>

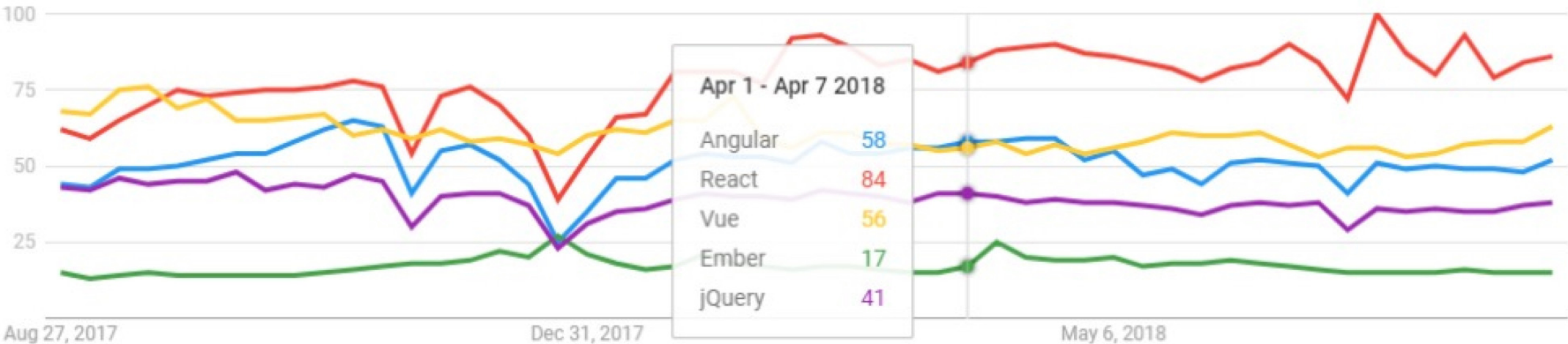
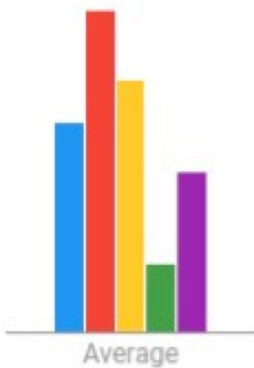
Todo list – jQuery style

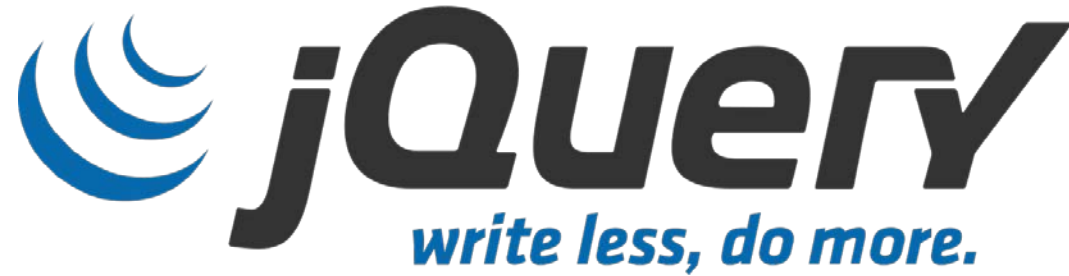
PIECE OF CAKE



MMMMM CAKE

Interest over time 





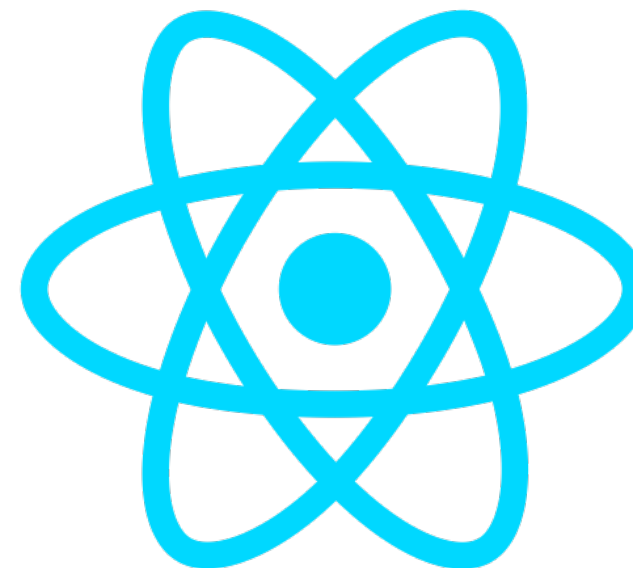
Fördelar

- Enkelt att arbeta med Ajax
- Väldigt snabbt att utveckla med
- X-browser
- Väldigt enkel DOM-manipulation
- Väldigt många plugins
- Snabbt att hitta buggar

Nackdelar

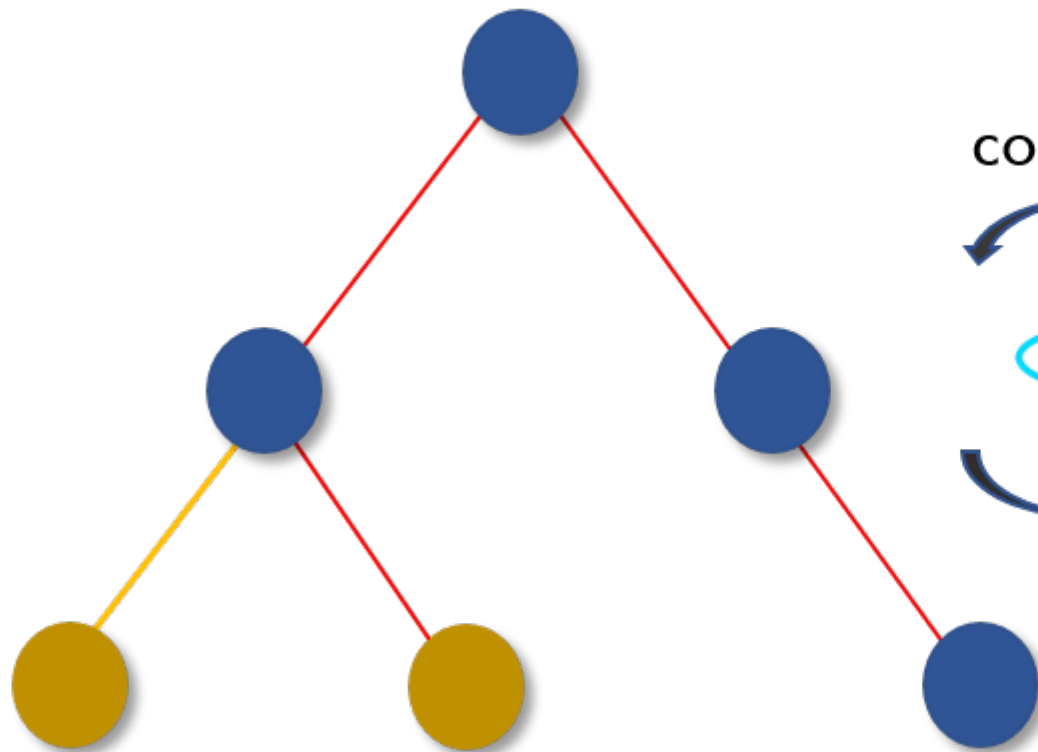
- Skapar overhead till webbsidan

Varför använder vi JS-ramverk?

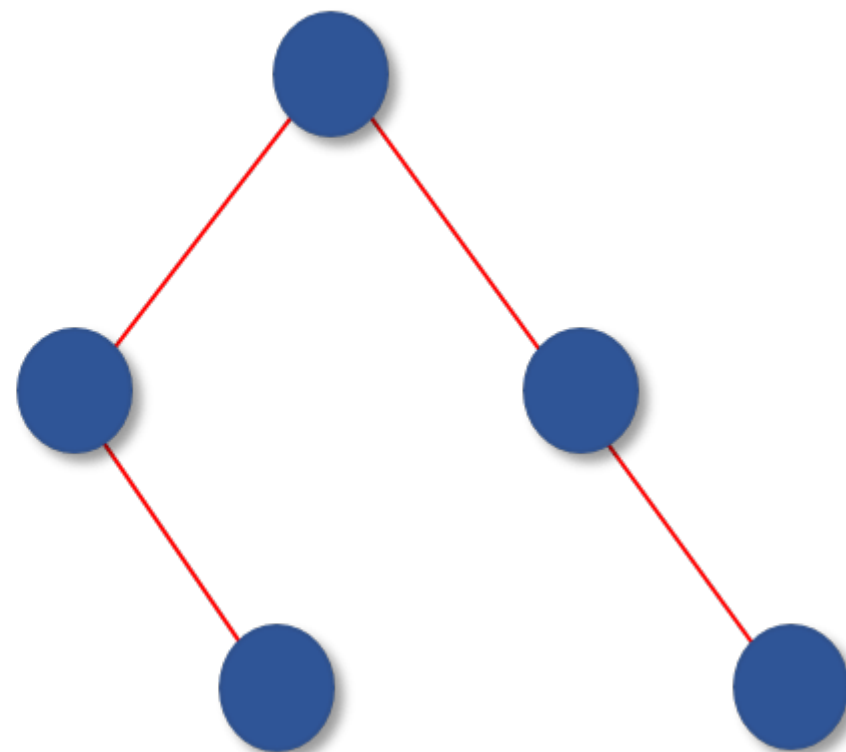


**KEEPING THE UI
IN SYNC
WITH THE STATE
IS HARD**

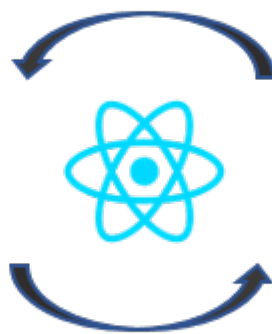
Virtual DOM



Real DOM

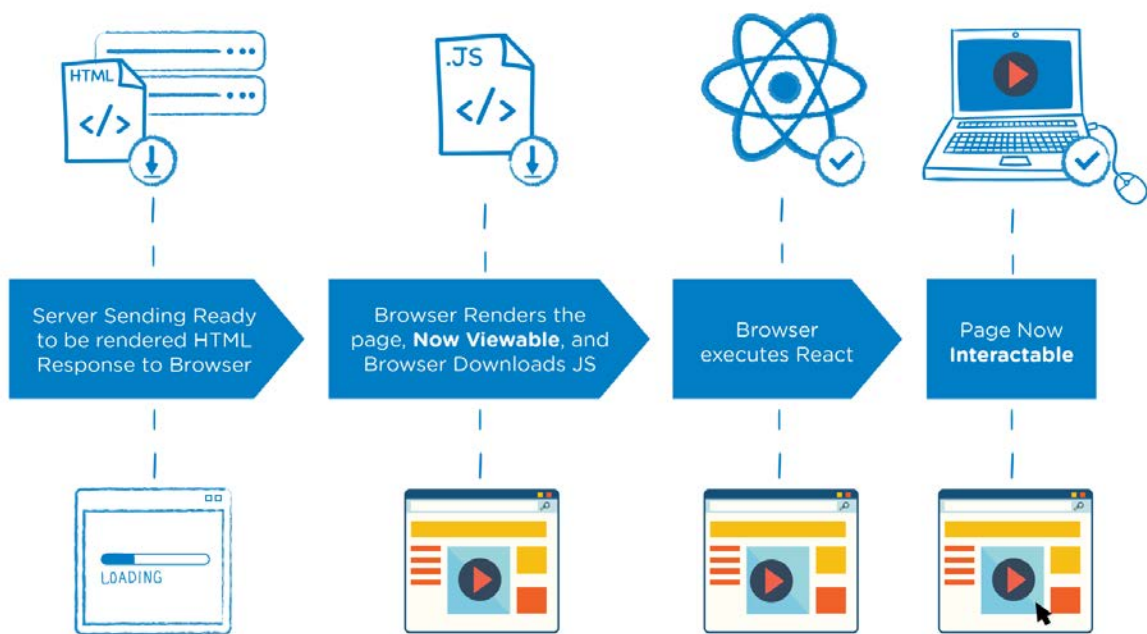


compare

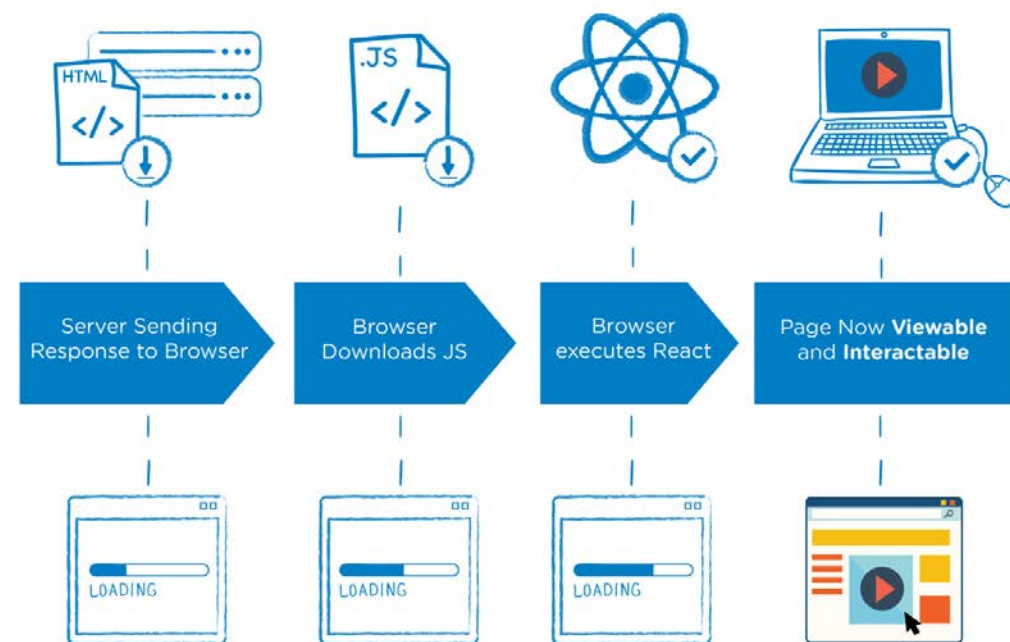




SSR



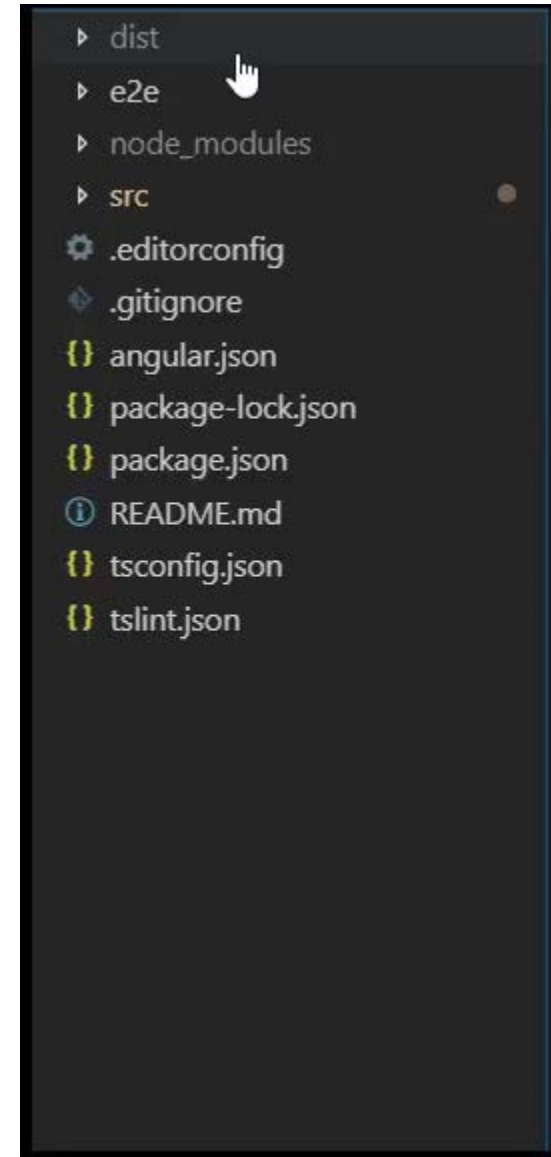
CSR



Att använda ett ramverk vs. ett bibliotek

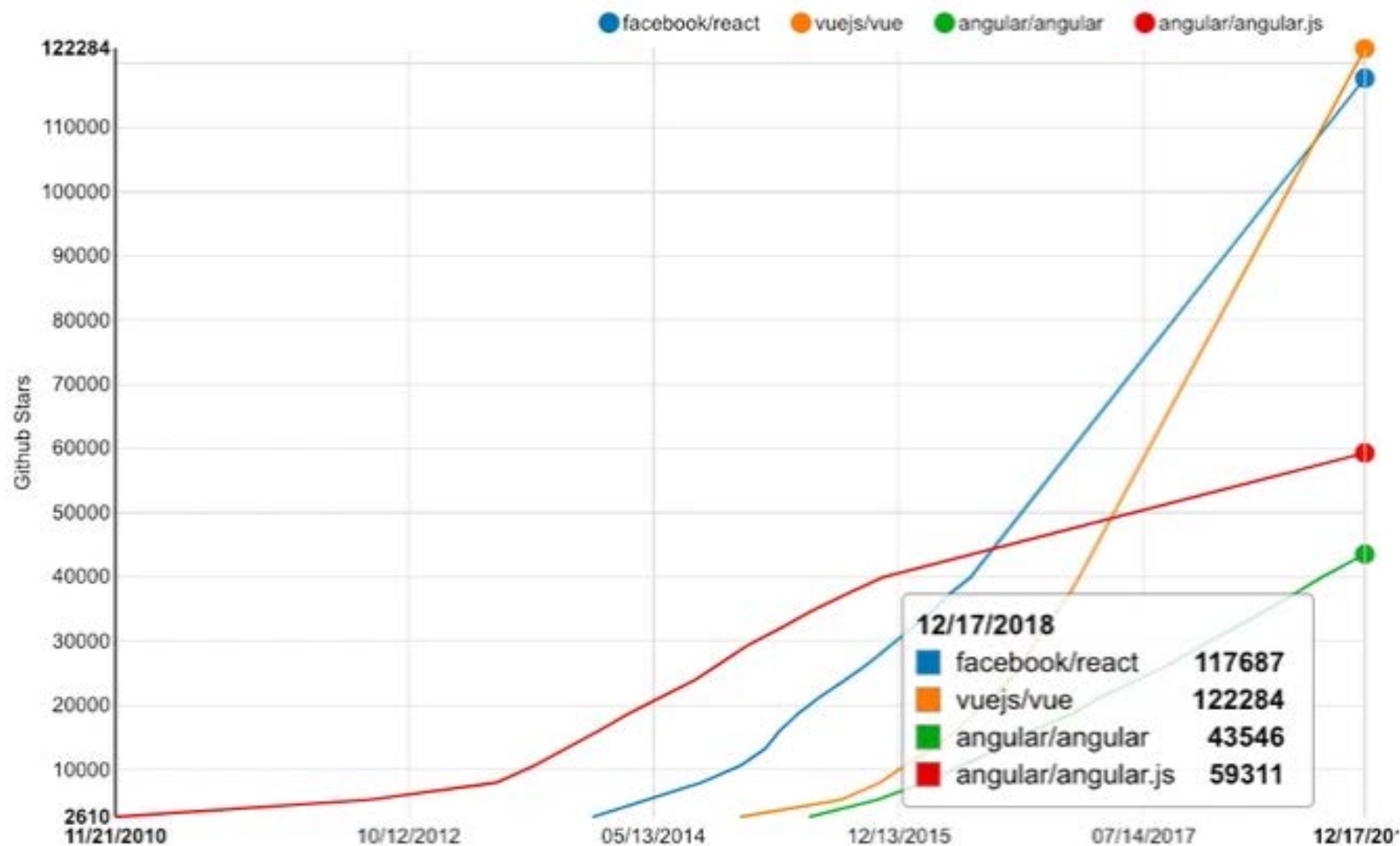


jQuery.js



The history of Angular vs React vs Vue

	Angular	React	Vue
Initial release	2010	2013	2014
Official site	angular.io	reactjs.org	vuejs.org
Approx. size (KB)	500	100	80
Current version	7	16.6.3	2.17
Used by	Google, Wix	Facebook, Uber	Alibaba, GitLab



Number of stars on GitHub projects for Angular, React, and Vue

För och nackdelar med de olika ramverken

- <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>
- <https://medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d>
- <https://kruschecompany.com/blog/post/ember-jQuery-angular-react-vue-what-to-choose>



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.

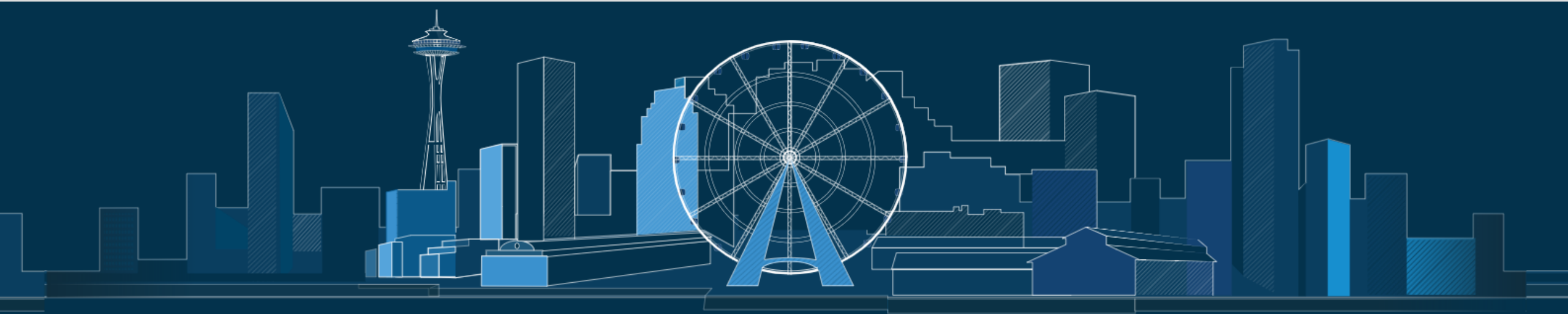
app.component

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);
```


Exempel på TypeScript

```
class Student {
    fullName: string;
    constructor(public firstName: string, public middleInitial: string, public lastName: string) {
        this.fullName = firstName + " " + middleInitial + " " + lastName;
    }
}

interface Person {
    firstName: string;
    lastName: string;
}

function greeter(person: Person) {
    return "Hello, " + person.firstName + " " + person.lastName;
}

let user = new Student("Jane", "M.", "User");

document.body.innerHTML = greeter(user);
```



Todo-lista med Angular

Welcome to * Todo List!

Todos

What needs to be done?

First todo

X



Sammanfattning - Angular

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
```



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

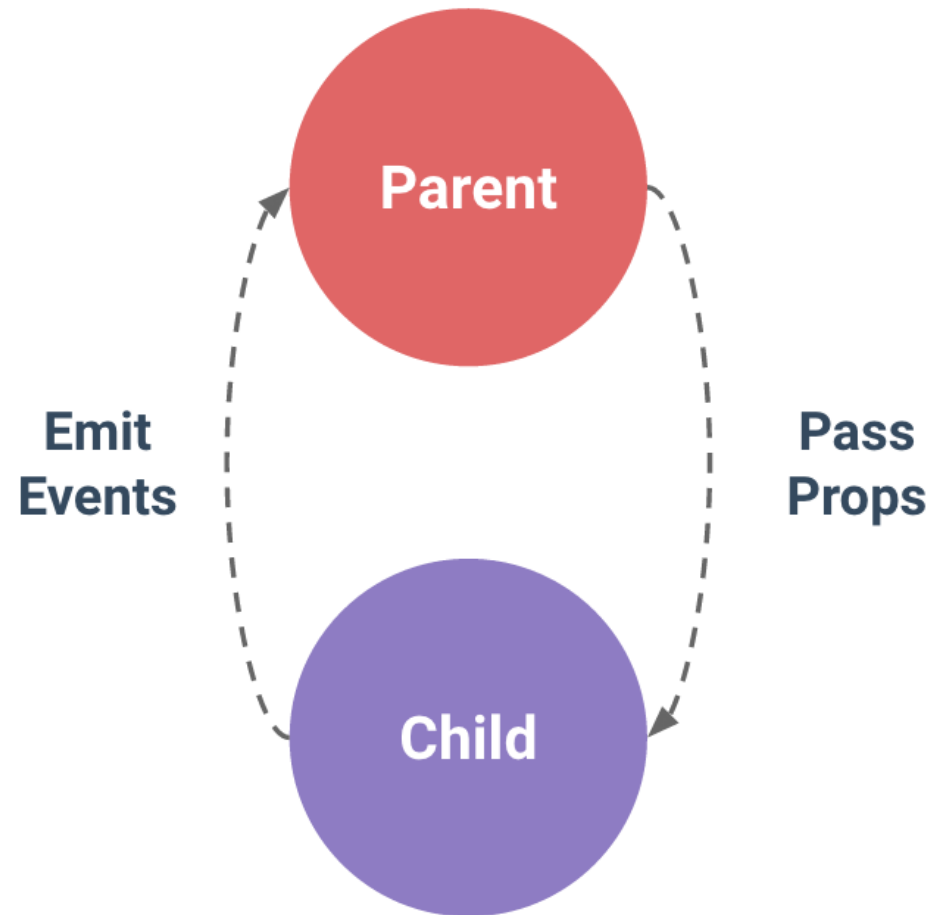
1 todo left to do.

`<app>`

`<todolist>`

`<todo>`

Komponenter?





Todo-lista med Vue



Sammanfattning med Vue

☰ index.html ×

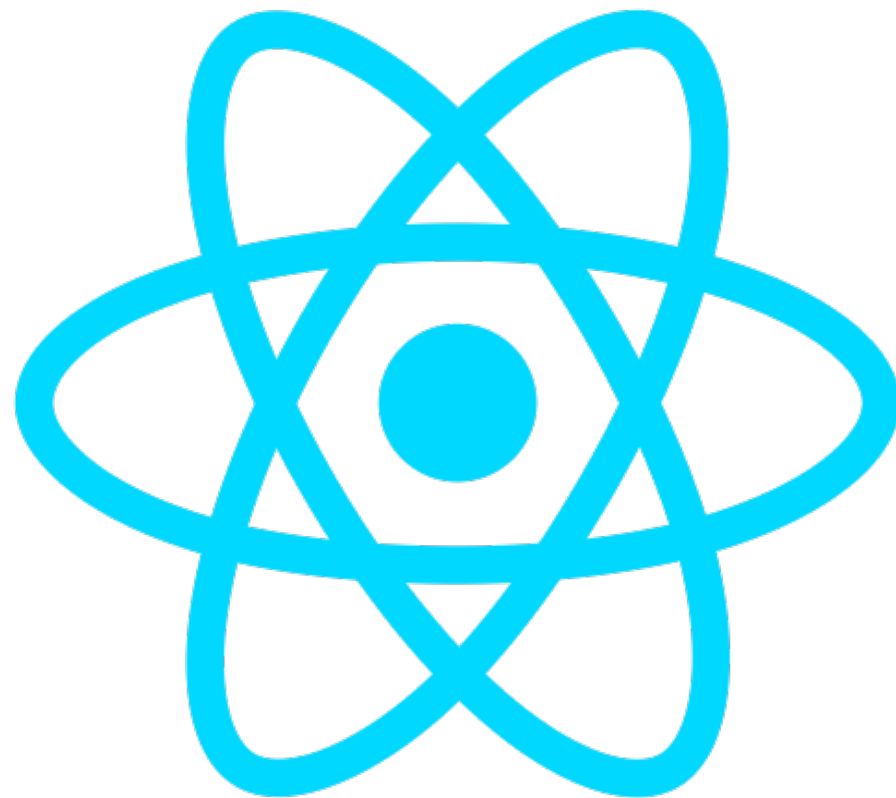
JS script.js

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         {
55           id: 0,
56           title: "First item"
57         }
58       ]
59     };
60   },
61   var app = new Vue({
62     el: "#app",
63     data: {
64       title: "Vue",
65     }
66   });
67
```



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

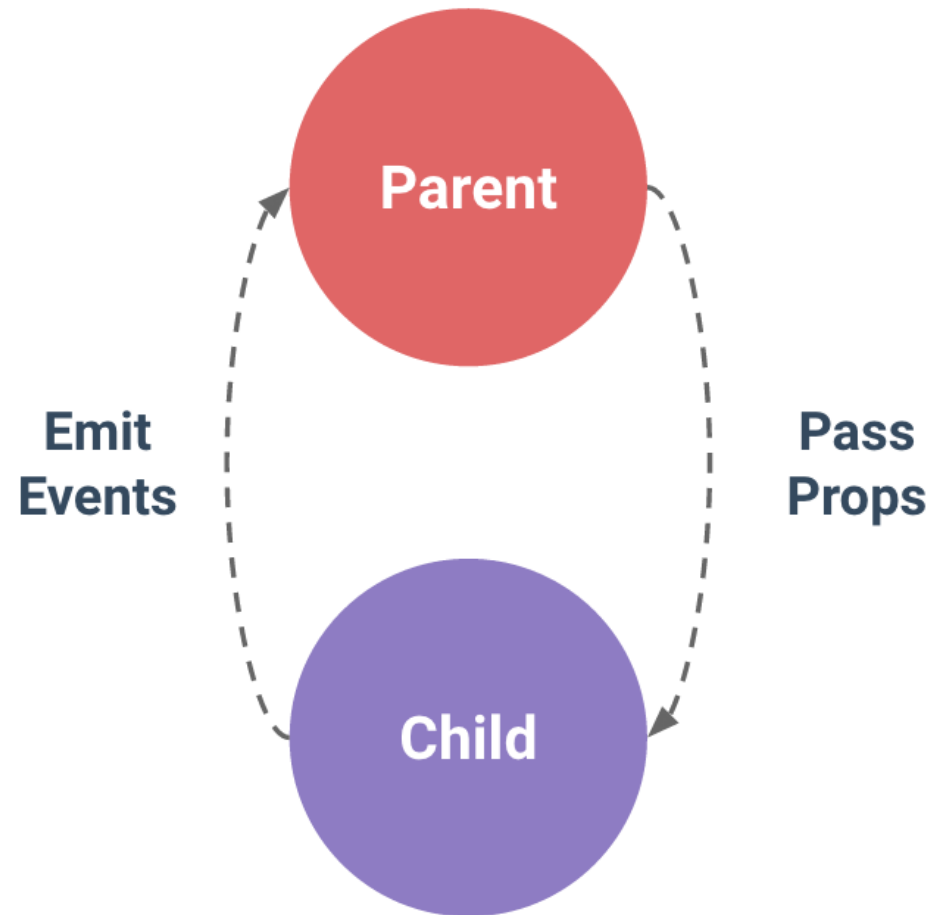
1 todo left to do.

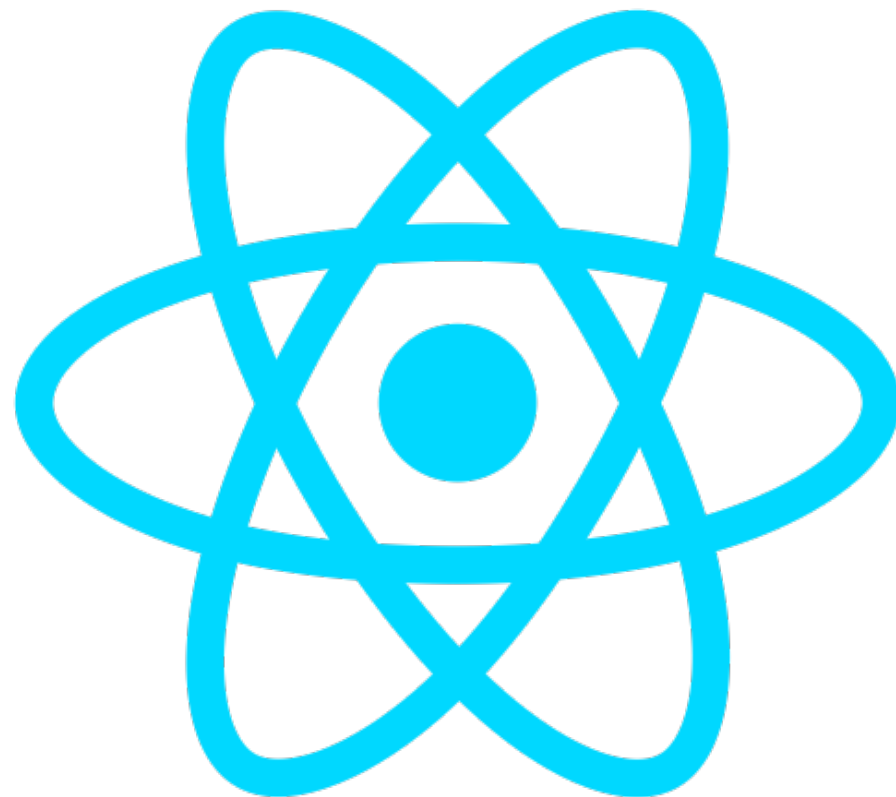
App

TodoList

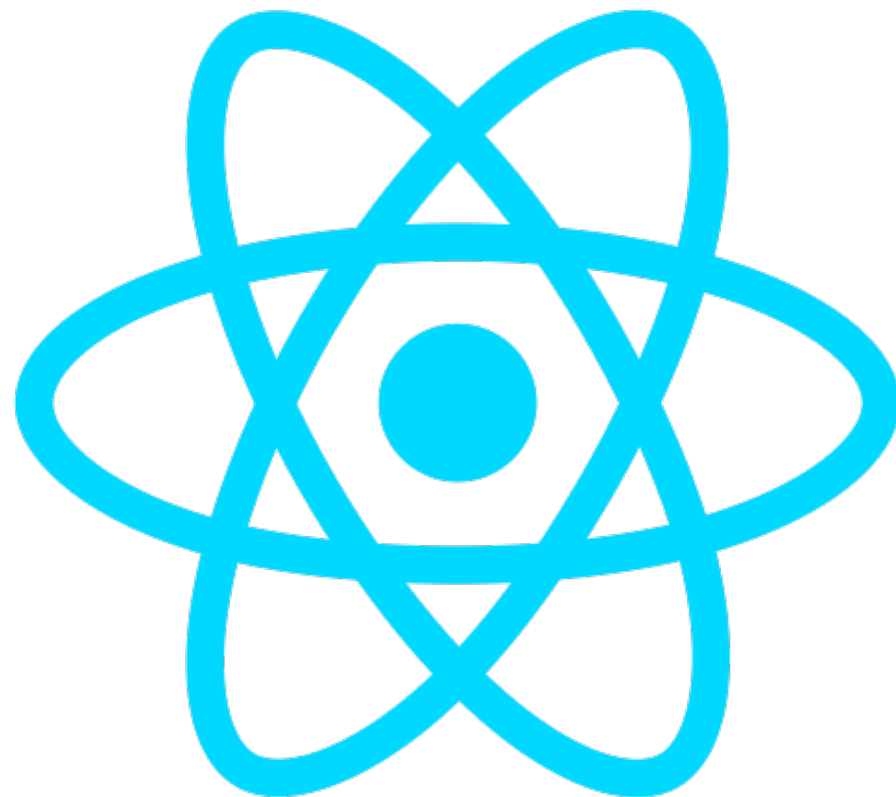
Todo

Komponenter?





Todo-lista med React



Sammanfattning med React

```

index.html x JS App.js # App.css JS TodoList.js JS Todo.js # main.763c0672.chunk.css
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <meta name="theme-color" content="#000000" />
8     <!--
9     manifest.json provides metadata used when your web app is installed on a
10    user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-
11    -->
12    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
13    <!--
14    Notice the use of %PUBLIC_URL% in the tags above.
15    It will be replaced with the URL of the `public` folder during the build.
16    Only files inside the `public` folder can be referenced from the HTML.
17
18    Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
19    work correctly both with client-side routing and a non-root public URL.
20    Learn how to configure a non-root public URL by running `npm run build`.
21    -->
22    <title>React App</title>
23    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap
24  </head>
25  <body>
26    <noscript>You need to enable JavaScript to run this app.</noscript>
27    <div id="root"></div>
28    <!--
29    This HTML file is a template.
30    If you open it directly in the browser, you will see an empty page.
31
32    You can add webfonts, meta tags, or analytics to this file.
33    The build step will place the bundled scripts into the <body> tag.
34
35    To begin the development, run `npm start` or `yarn start`.
36    To create a production bundle, use `npm run build` or `yarn build`.
37    -->
38  </body>
39 </html>

```

```

index.html JS App.js JS index.js x # App.css JS TodoList.js JS
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5 import * as serviceWorker from './serviceWorker';
6
7 ReactDOM.render(<App />, document.getElementById('root'));
8
9 // If you want your app to work offline and load faster, you can change
10 // unregister() to register() below. Note this comes with some pitfalls.
11 // Learn more about service workers: https://bit.ly/CRA-PWA
12 serviceWorker.unregister();

```

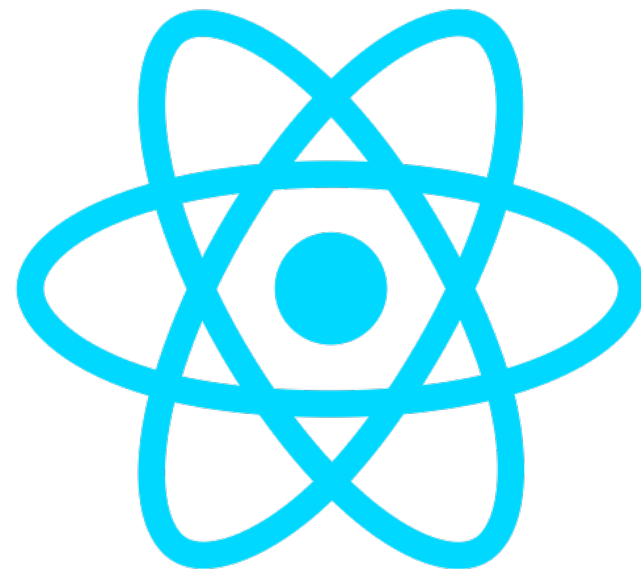
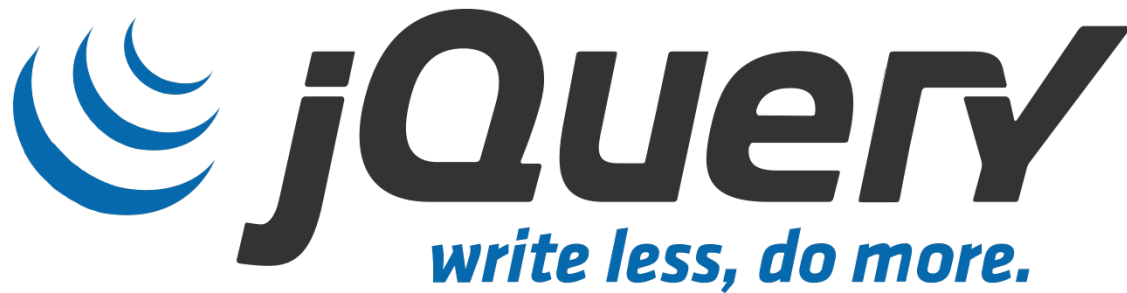
```

index.html JS App.js x # App.css JS TodoList.js JS
1 import React from 'react';
2 import TodoList from './components/TodoList';
3 import './App.css';
4
5 function App() {
6   return (
7     <div className="container">
8       <h1>Welcome to React Todo List!</h1>
9       <h2>Todos</h2>
10      <TodoList />
11    </div>
12  );
13 }
14
15 export default App;

```

```
index.html JS App.js JS index.js # App.css JS TodoList.js JS Todo.js
1 import React from 'react';
2 import Todo from './Todo';
3
4 class TodoList extends React.Component {
5
6   state = {
7     todos: [{
8       title: "First Item",
9       id: 0
10     }],
11     newTodo: "",
12     lastId: 1
13   };
14
15   addTodo = (event) => {
16     if (event.keyCode === 13) {
17       let todo = {
18         title: this.state.newTodo,
19         id: this.state.lastId
20       };
21       this.setState({
22         todos: [...this.state.todos, todo],
23         newTodo: "",
24         lastId: this.state.lastId + 1
25       });
26     }
27   }
28
29   updateNewTodo = (event) => {
30     this.setState({
31       newTodo: event.target.value
32     });
33   }
34
35   deleteItem = (id) => {
36     let todos = this.state.todos.filter(todo => todo.id !== id);
37     this.setState({
38       todos: todos
39     });
40   }
41
42   render(){
43     return (
44       <div>
45         <input type="text" value={this.state.newTodo} onKeyDown={this.addTodo} onChange={this.updateNewTodo} placeholder="What needs to be done?" className="form-control" />
46         <ul id="todos" className="list-group">
47           { this.state.todos.map(todo => <Todo item={todo} deleteItem={this.deleteItem} key={todo.id} /> ) }
48         </ul>
49         <span id="nr-of-todos"><strong>{this.state.todos.length}</strong> {this.state.todos.length === 1 ? "todo" : "todos"} left to do.</span>
50       </div>
51     );
52   }
53 }
54
55 export default TodoList;
```

```
index.html JS App.js JS index.js # App.css JS TodoList.js JS Todo.js x
1 import React from 'react';
2
3 class Todo extends React.Component {
4   render(){
5     return (
6       <li className="list-group-item">
7         <label>{this.props.item.title}</label>
8         <button className="btn btn-sm btn-danger" onClick={() => this.props.deleteItem(this.props.item.id)}>X</button>
9       </li>
10     );
11   }
12 }
13
14 export default Todo;
```



Frågor?

Kursens projekt

TEAMWORK



**THAT'S HOW
WE DO IT**

Skapa en webbapplikation genom ett JS-ramverk

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

Krav på projektet

- Ni ska motivera ert val av JS-ramverk, genom en jämförelse med minst två andra JS-ramverk. Inkludera källor.
- Ni ska bygga en webbapplikation med valt ramverk, men minimumkrav:
 - Ni ska använda er att minst ett externa webb-API (som inte är något av de API som använts vid någon laboration)
 - Ni ska tillåta användaren att spara data i relation till API:t
 - T.ex. ett filmbibliotek om man använder ett film-API
 - T.ex. statistik / diagram om vädertemperaturer de senaste vädersökningarna om man använder ett väder-API
 - Ni ska använda er utav Github för versionshantering av ert projekt
 - Ni ska ha ett grafiskt gränssnitt på er webbapplikation med fokus på användbarhet för olika enheter. Webbsidorna ska se fina ut i mobil/surfplatta/dator.

Skapande av team

- Max 3 studenter per team
 - Rekommenderat, 3 studenter
- Gärna över programgränser
- Teamen definieras av er på Canvas
- No team – no problem!
 - Måndagen v.20 (nästa vecka) kl. 10.00 i sal NI:A0407 så finns möjlighet att hitta team-medlemmar. Så har du inget team till dess – kom förbi!



[Konto](#)[Översikt](#)[Kurser](#)[Kalender](#)[Inkorg](#)[Commons](#)[Hjälp](#)

[VT2019-DA355A-TS996](#) > [Personer](#) > [Grupper](#)

VT2019

[Hem](#)[Meddelanden](#)[Uppgifter](#)[Omdömen](#)[Personer](#)[Moduler](#)[Konferenser](#)[Kursöversikt](#)[Samarbeten](#)[Diskussioner](#)[Sidor](#)[Filer](#)[Lärandemål](#)[Quiz](#)[Inställningar](#)[Alla](#)[Projekt - JS-ramverk](#)[+ Gruppsammansättning](#)Egen-anmälan har aktiverats för de här grupperna. [?](#)

Grupper är begränsade till 3 medlemmar.

[+ Grupp](#)

Icke tilldelade studenter (59)

Sök efter användare

Ted Ahlberg +

Christopher Andersson ... +

Oscar Andersson +

Aida Arvidsson +

Christian Barlöf +

Viktor Berger Grönros +

Johannes Blüml +

Oscar Carlsson +

Victor Dahl +

Sara Kristina Dalvig +

Oskar Engström Magnu... +

Zacharias Eryd Berlin +

Hanna Fellving +

Elin Forsnor Nilsson +

Grupper (20)

[▶ Projekt - JS-ramverk 1](#)

0 / 3 studenter

[▶ Projekt - JS-ramverk 2](#)

0 / 3 studenter

[▶ Projekt - JS-ramverk 3](#)

0 / 3 studenter

[▶ Projekt - JS-ramverk 4](#)

0 / 3 studenter

[▶ Projekt - JS-ramverk 5](#)

0 / 3 studenter

[▶ Projekt - JS-ramverk 6](#)

0 / 3 studenter

[▶ Projekt - JS-ramverk 7](#)

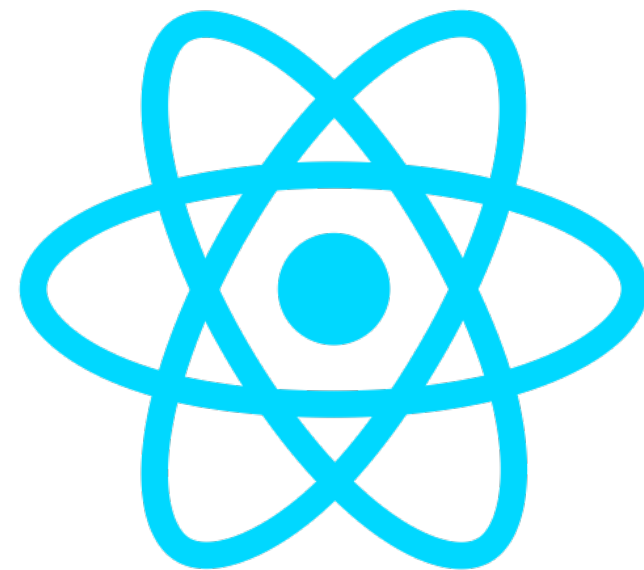
0 / 3 studenter

[▶ Projekt - JS-ramverk 8](#)

0 / 3 studenter



Vecka 19				
Måndag	09:15 - 12:00	Vue.js / React / Angular	NI:A0606	AT
Tisdag	13:15 - 15:00	Utvecklingsmetodik, pakethanterare	NI:A0307	JH
Torsdag	08:15 - 12:00	Labb: Utvecklingsmetodik	G8:405, G8:407	JH, CN, EI
Söndag	Deadline: Inlämningsuppgift 1			
Vecka 20				
Måndag	10:15 - 12:00	Föreläsning: Extra	NI:A0407	AT
Tisdag	13:15 - 17:00	Labb: Vue.js / React / Angular	OR:E222, OR:E223	AT, CN, EI
Torsdag	08:15 - 12:00	Labb: Extra	G8:405, G8:424	AT
Vecka 21				
Måndag	10:15 - 12:00	Projekthandledning	NI:A0407	AT
Tisdag	13:15 - 15:00	Projekthandledning	NI:A0606	AT
Torsdag	08:15 - 12:00	Labb: Extra	G8:405	AT
Vecka 22				
Måndag	08:15 - 12:00	Projekthandledning	NI:A0606	AT
Måndag	13:15 - 17:00	Labb: Extra	NI:A0606	AT
Vecka 23				
Måndag	08:15 - 12:00	Presentation av projekt	NI:B0E07	AT, JH
Tisdag	13:15 - 17:00	Presentation av projekt	NI:B0E07	AT, JH



Frågor kring projektet?