



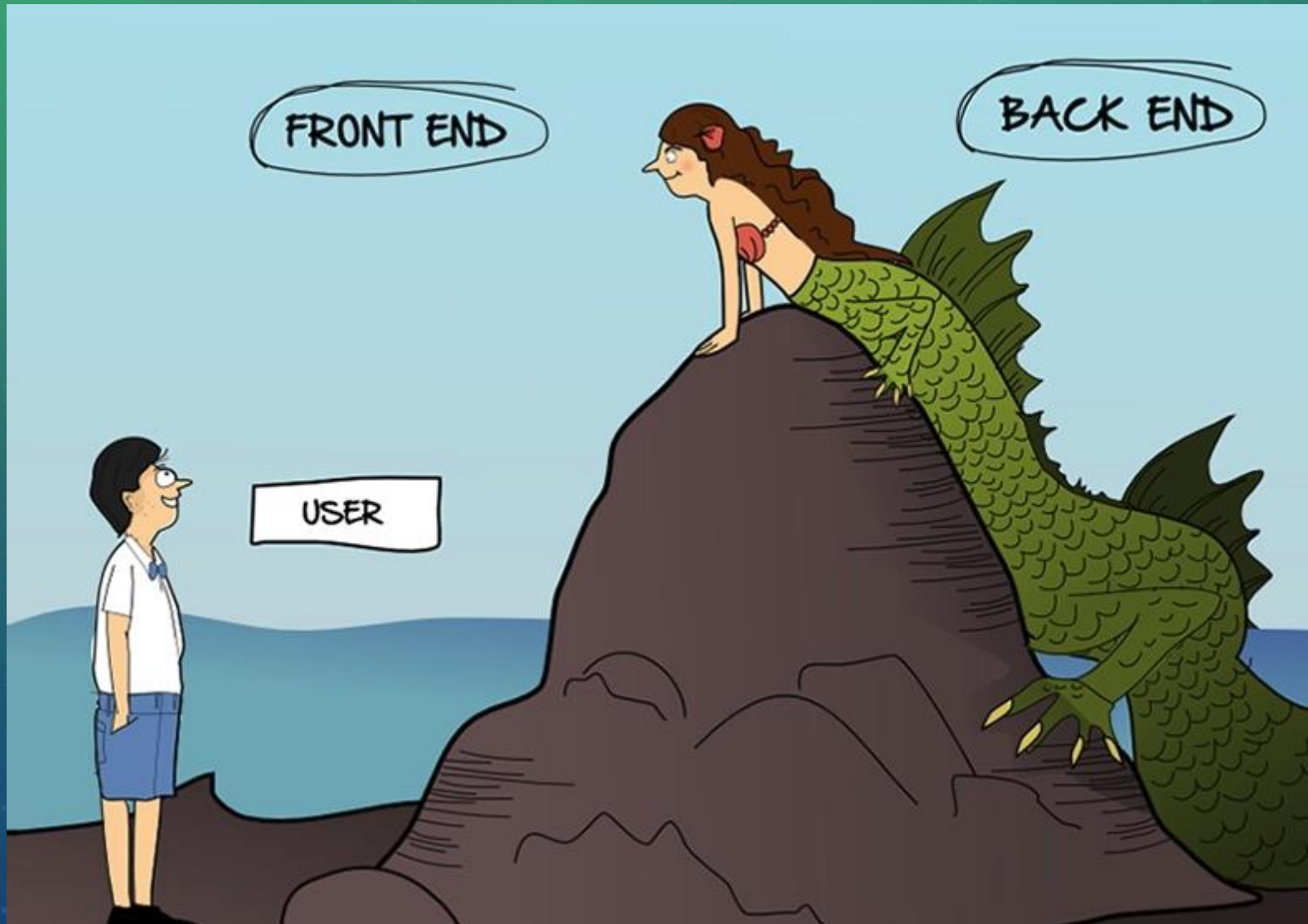
# FULLSTACK WEBFEJLESZTÉS, AVAGY REACT ÉS NODE.JS HASZNÁLATA A GYAKORLATBAN

---

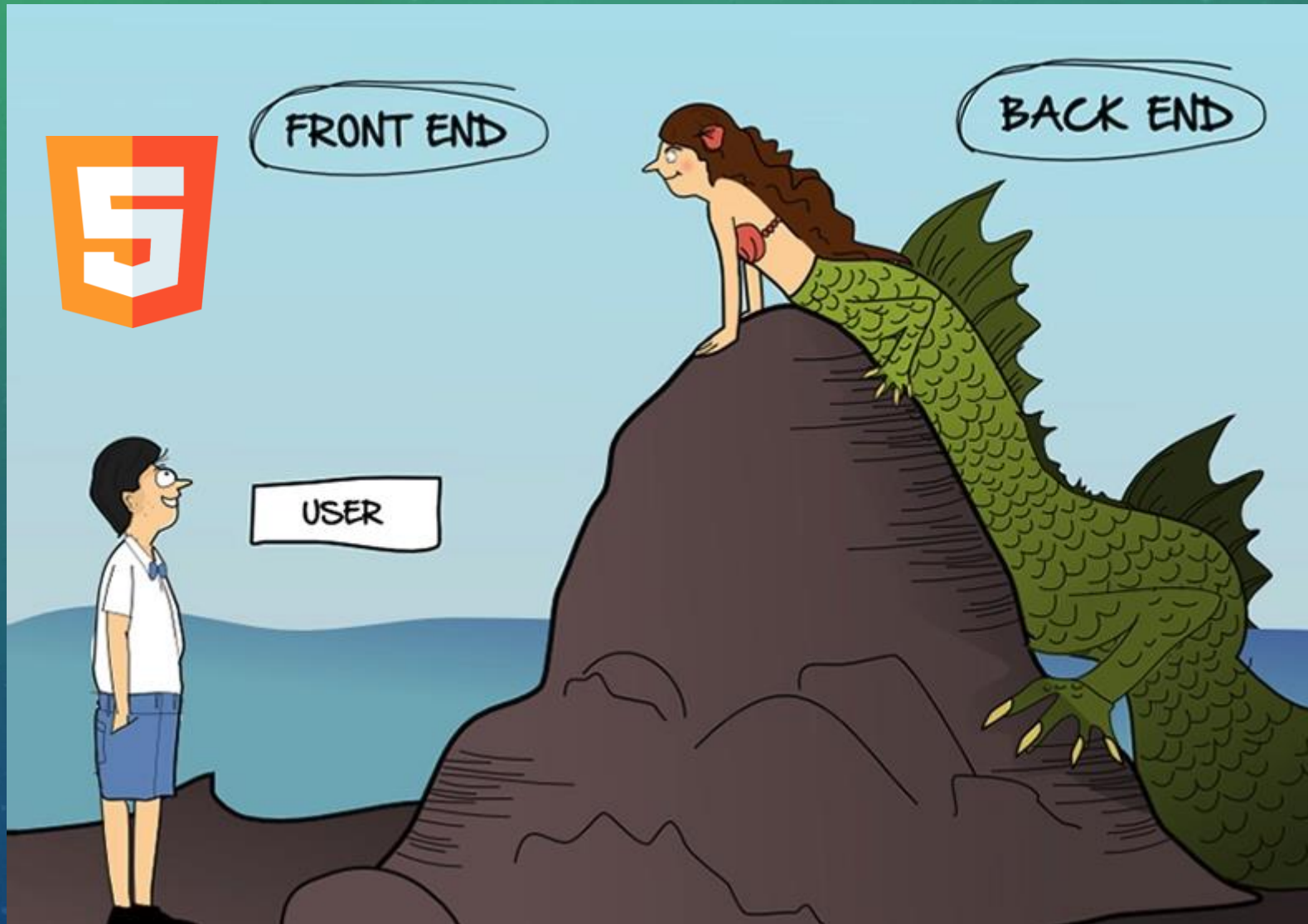
**CSUVIK VIKTOR**

SZEGEDI TUDOMÁNYEGYETEM  
MTA-SZTE MESTERSÉGES INTELLIGENCIA KUTATÓCSOPORT

# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?

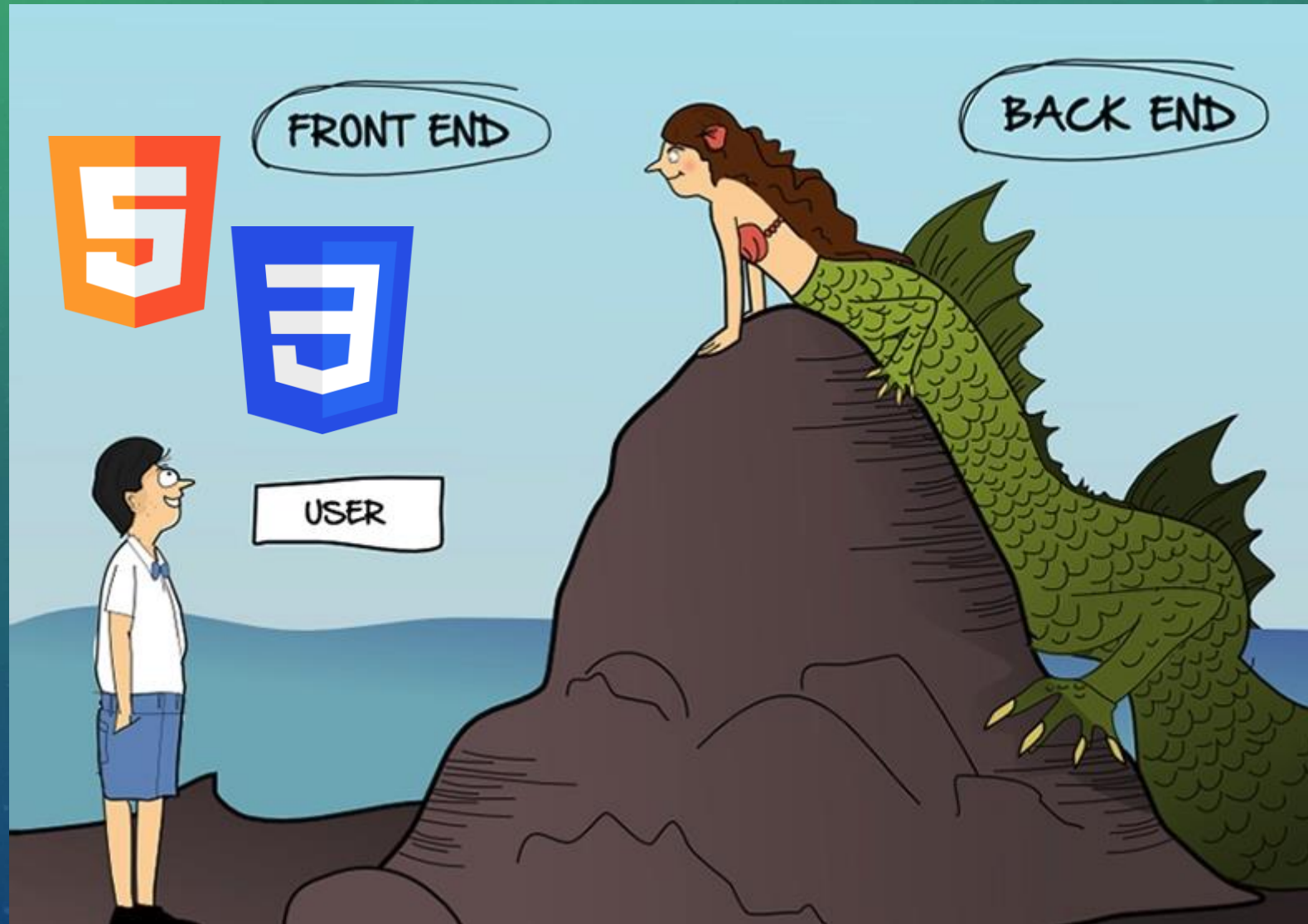


# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?





# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?



# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?



# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?

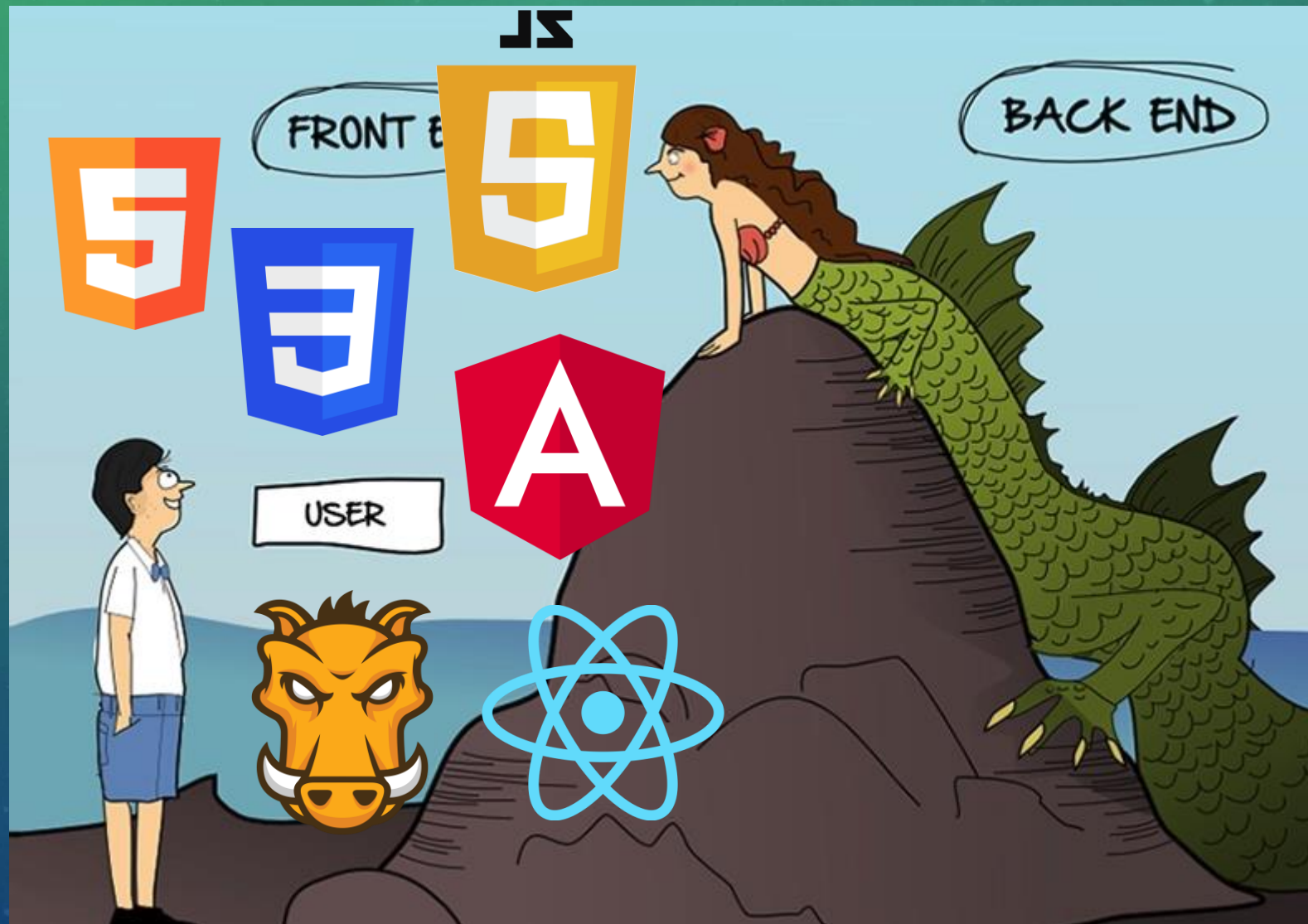




# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?

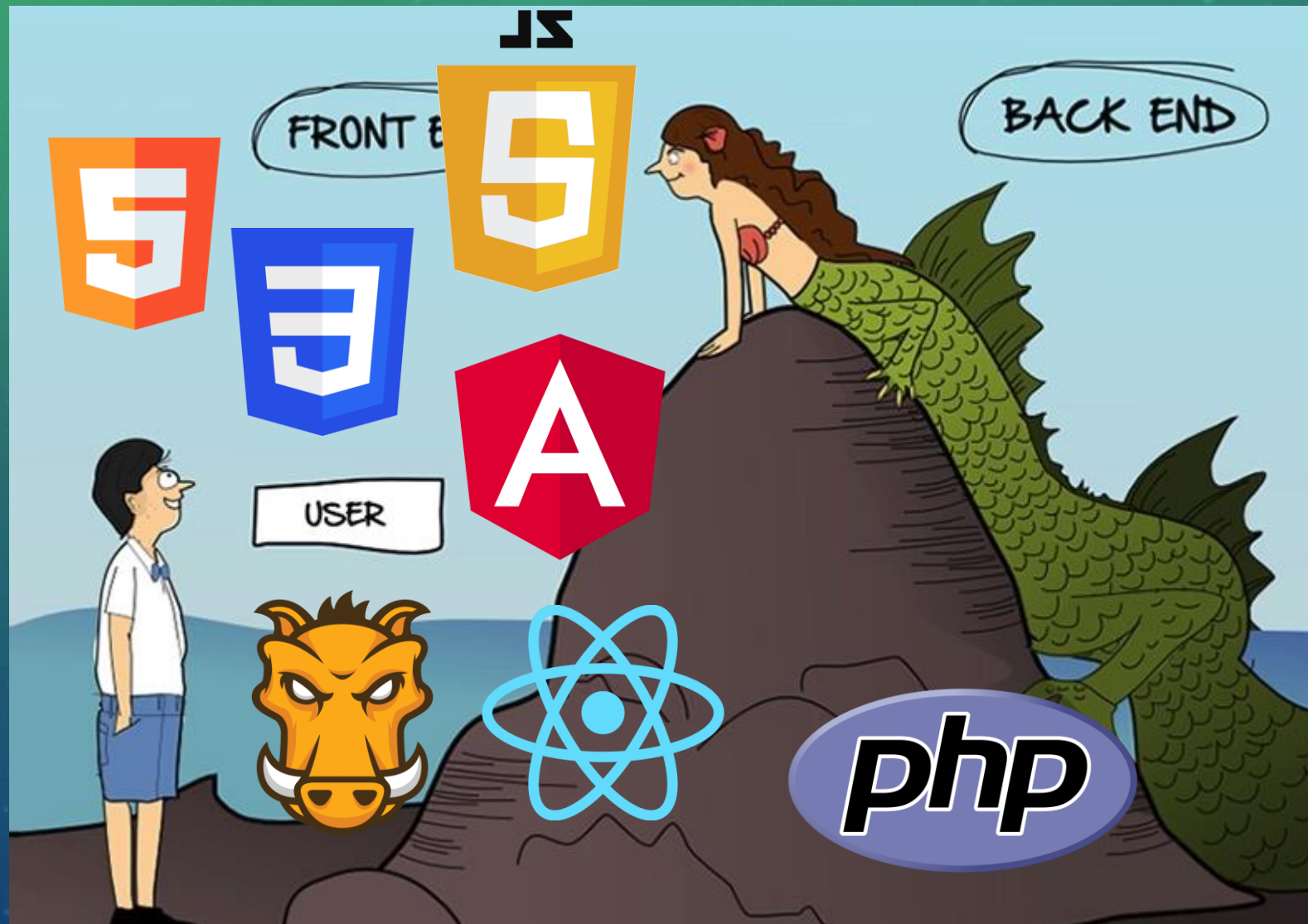


# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?

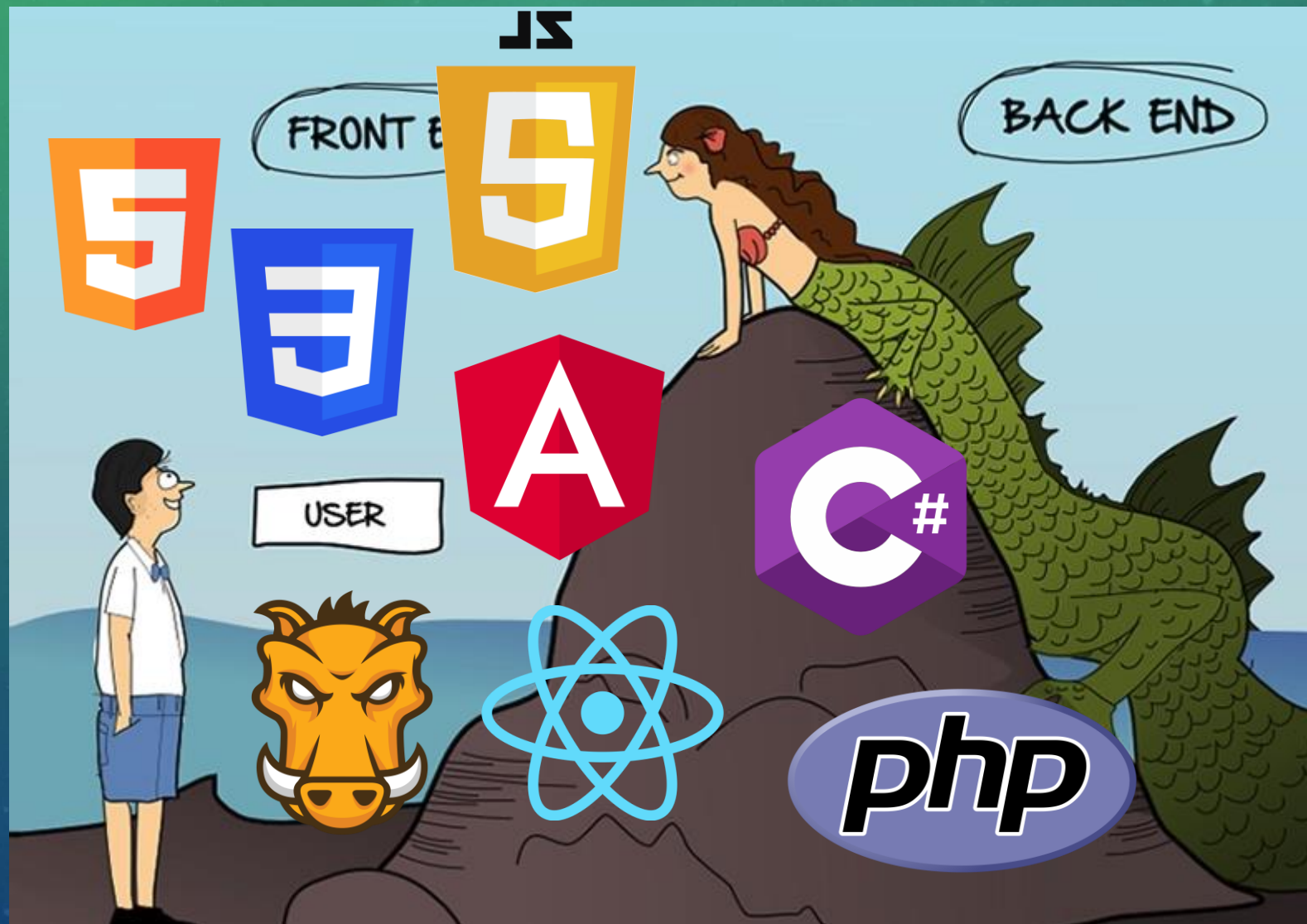




# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?

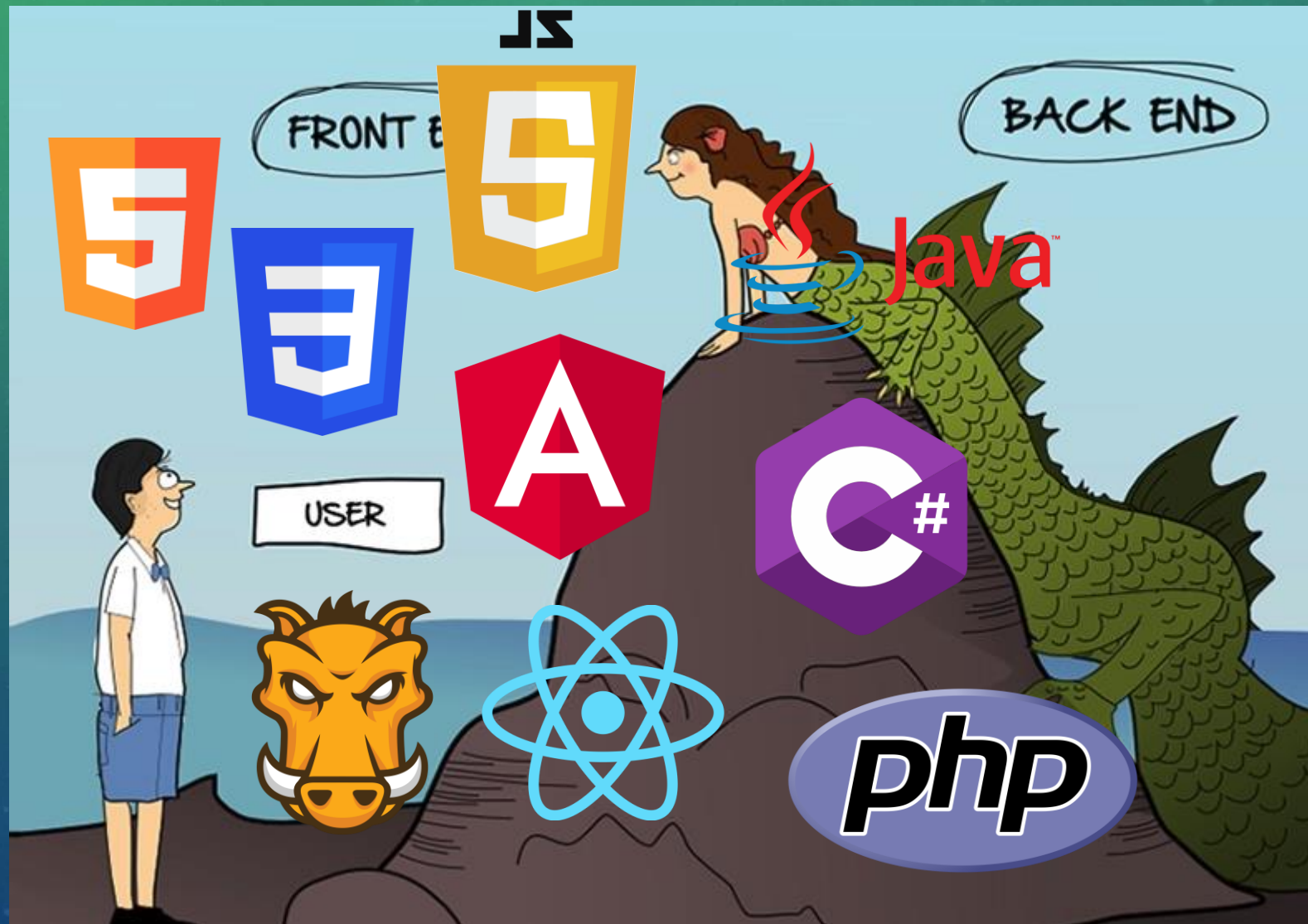


# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?



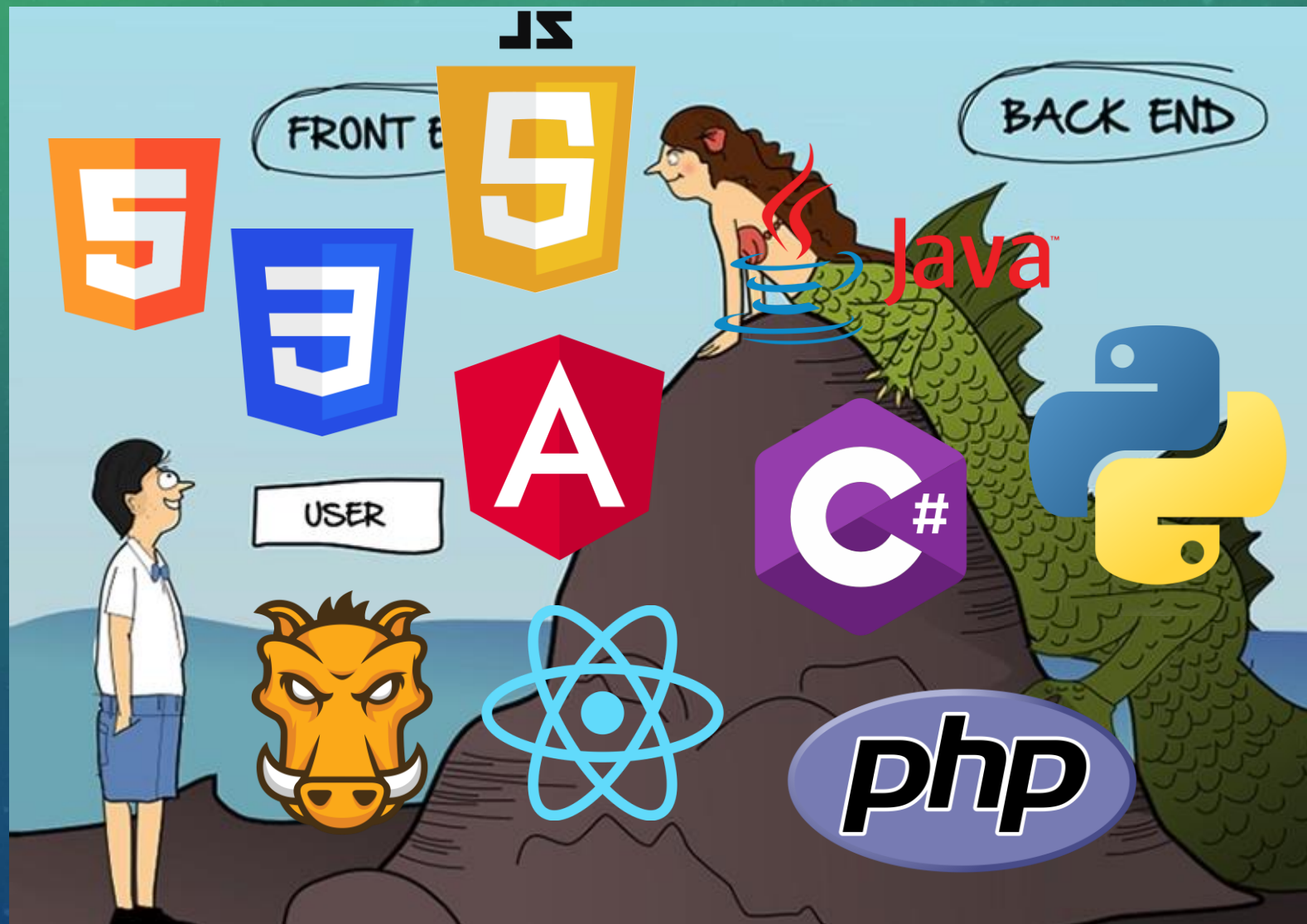


# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?

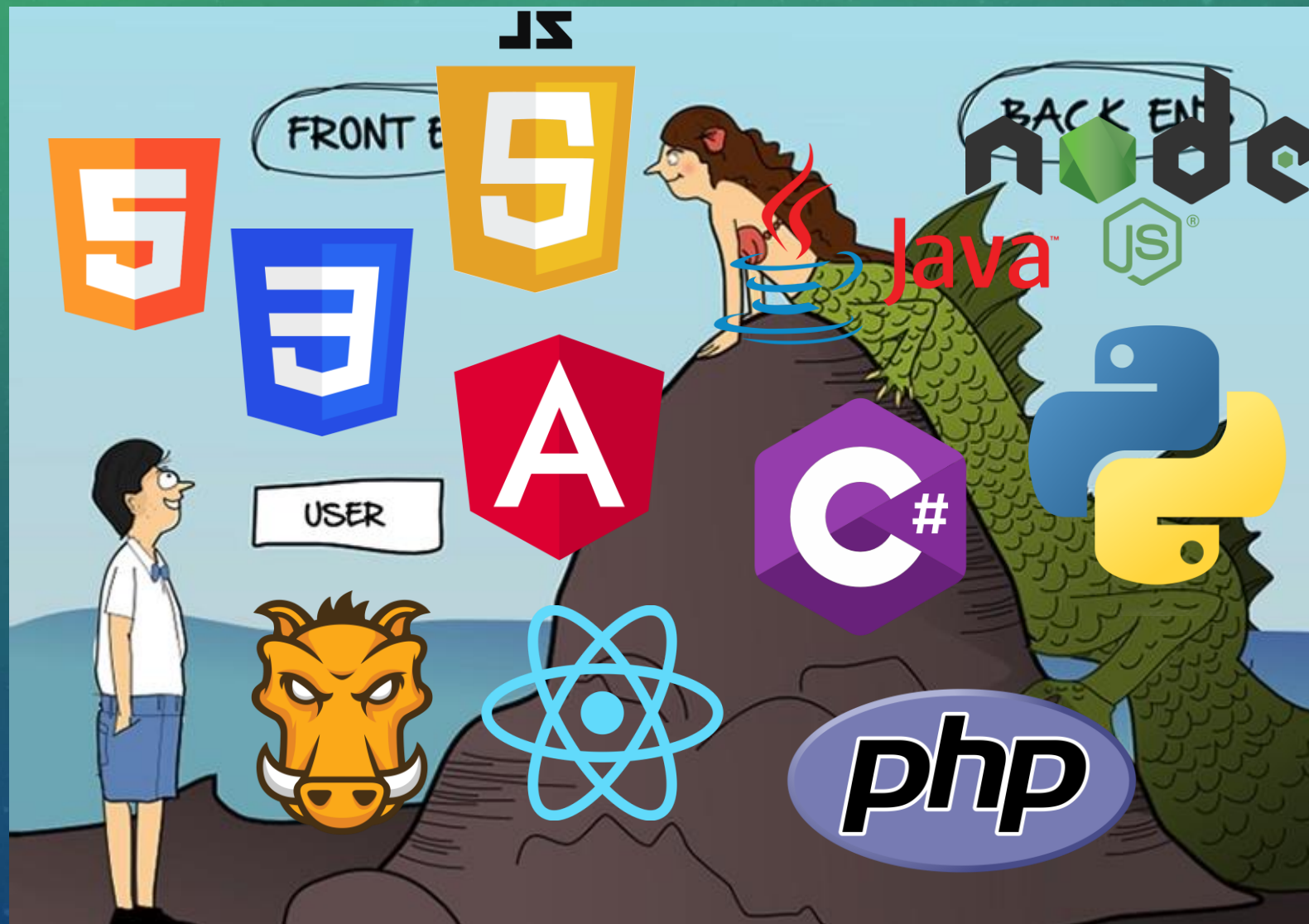




# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?



# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?



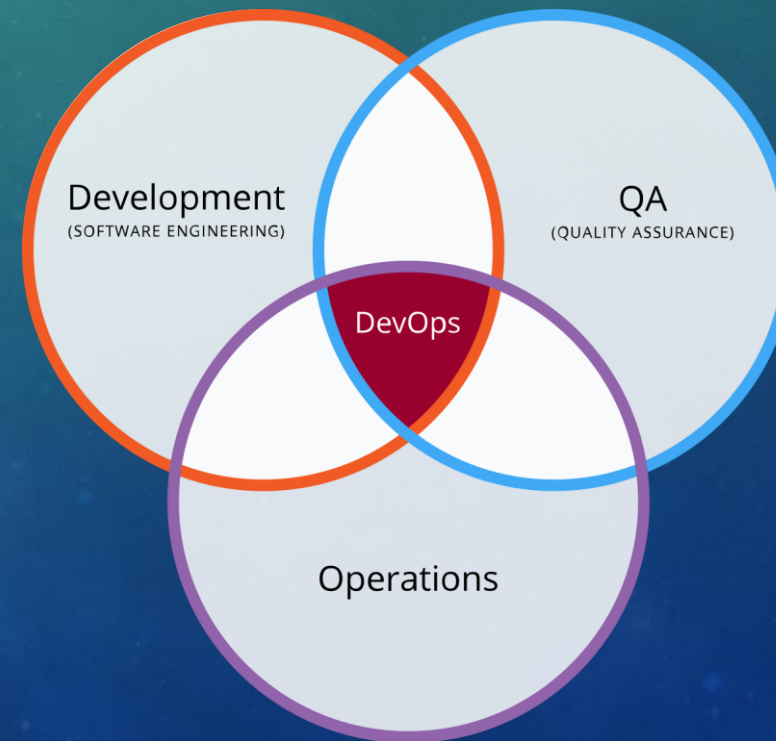
# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?

- Szóval a Full-Stack webfejlesztő:
  - A böngészőben megjelenő dolgokért felelős
  - A szerveren futó dolgokért felelős
  - + adatbázist programoz



# MI AZ A FULLSTACK / FULL STACK / FULL-STACK?

- Szóval a Full-Stack webfejlesztő:
  - A böngészőben megjelenő dolgokért felelős
  - A szerveren futó dolgokért felelős
  - + adatbázist programoz



# NÉPSZERŰ STACK-EK

- LAMP stack: JavaScript - Linux - Apache - MySQL - PHP
- LEMP stack: JavaScript - Linux - Nginx - MySQL - PHP
- MEAN stack: JavaScript - MongoDB - Express - AngularJS - Node.js
- MERN stack: JavaScript - MongoDB - Express - React - Node.js
- Django stack: JavaScript - Python - Django - MySQL
- Ruby on Rails: JavaScript - Ruby - SQLite - Rails

## PRO

- Gyors prototípus
- Csökkentett projekt költség (?)
- Mindenki ért mindent (?)
- Kevesebb kommunikáció csapattagok között
- Mindenki fejleszthet mindent

## KONTRA

- Adott stack nem biztos, hogy optimális a projekthez
- Adott megoldás kulcsszemélyt generálhat
- A megoldás a fejlesztő készségeitől függ
- Mindenki fejleszthet mindent



# A MERN STACK



**MongoDB**

# A MERN STACK



**MongoDB**



**Express**

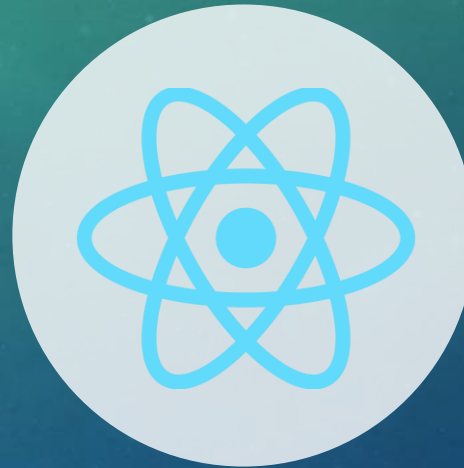
# A MERN STACK



**MongoDB**



**Express**



**React**



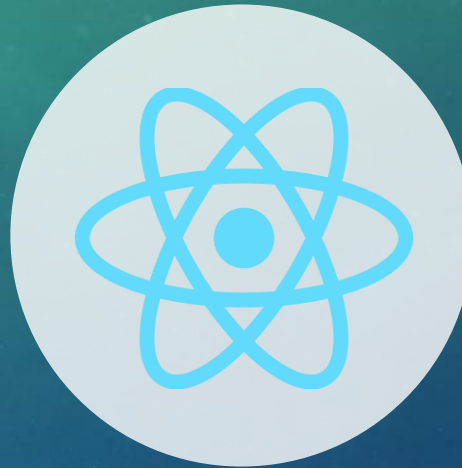
# A MERN STACK



**MongoDB**



**Express**



**React**



**Node.js**

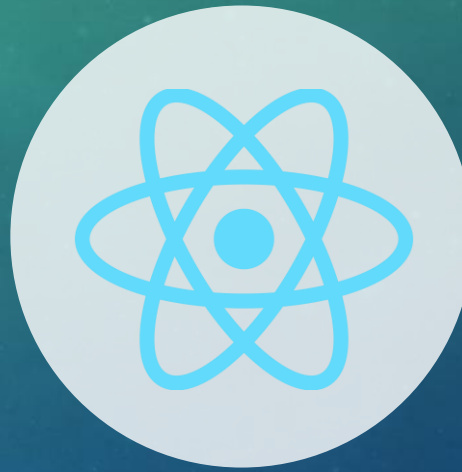
# A MERN STACK



**MongoDB**



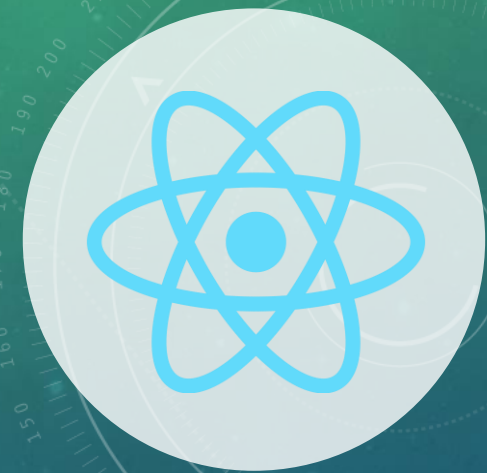
**Express**



**React**



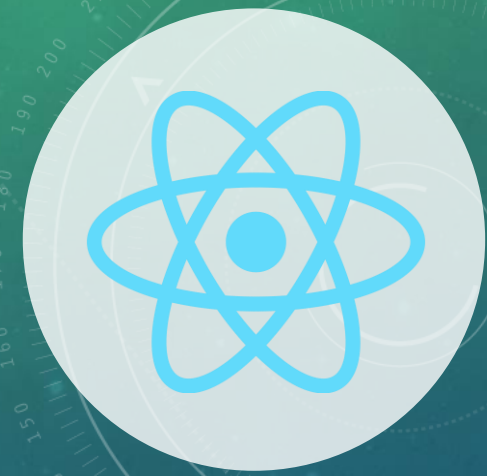
**Node.js**



- Ingyenes, nyílt forráskódú, JavaScript könyvtár
- **Komponensekre** alapul a benne való fejlesztés
  - Komplex UI interfészeket különálló részekre lehet bontani
- A Meta (Facebook) és más kisebb független fejlesztők tartják karban
- A React csak a UI elkészítésében és ezek renderelésében segít, egy teljes alkalmazás elkészítéséhez más eszközöket is használni kell



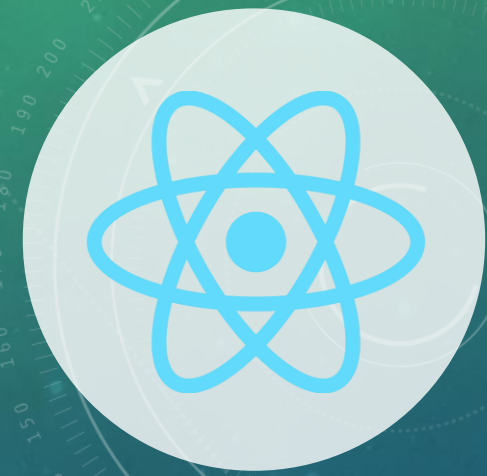
# REACT — LIBRARY VAGY FRAMEWORK?



- Library: én döntöm el mikor és hol használom
- Framework: ez határozza meg az alkalmazás folyamatait
- React nem nyújt teljes fejlesztési „ökoszisztémát”, nekünk kell kiválasztani az eszközöket, pl.:
  - Routing-hoz react-router
  - Form kezeléshez Redux-Forms, Formiq, vagy Final-Form
  - Stb.
- **A React ezért library (könyvtár)**
- De van framework ami a React-ra alapul, pl: Next.js

# REACT - KOMPONENSEK

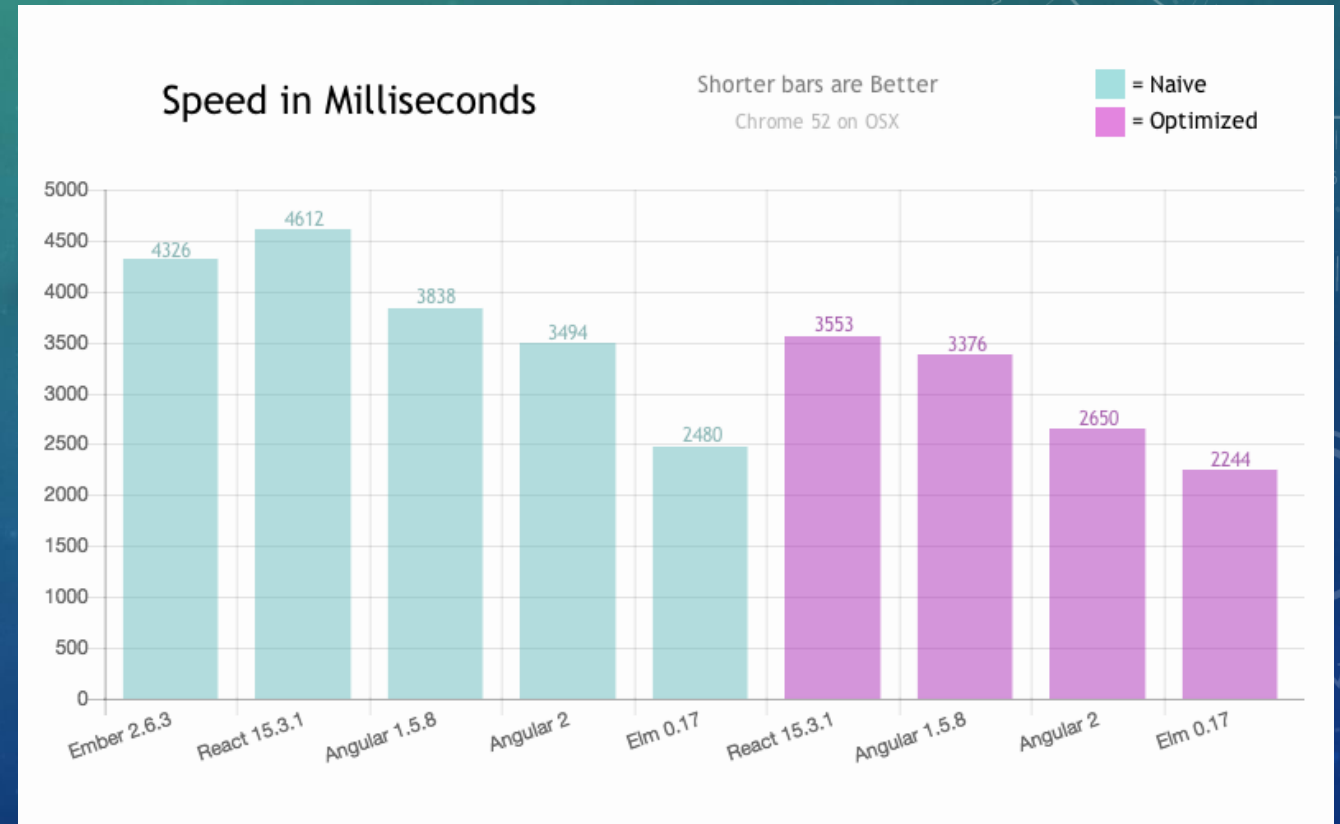
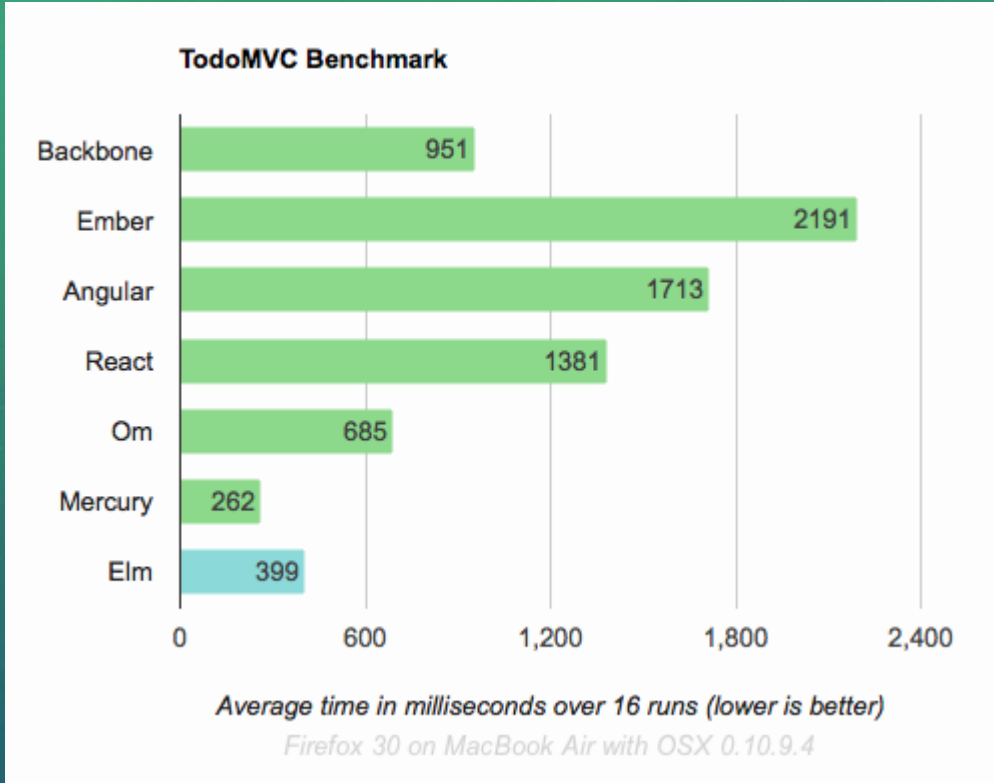
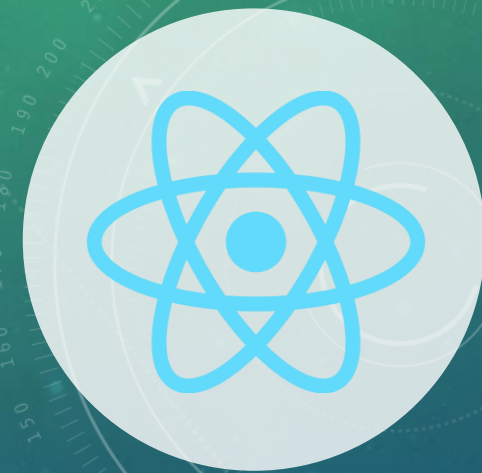
- A komponensek független és újrafelhasználható kód részletek. Ugyanazt a célt szolgálják, mint a JavaScript függvények, de elszigetelten működnek, és HTML kódot adnak vissza.



VS

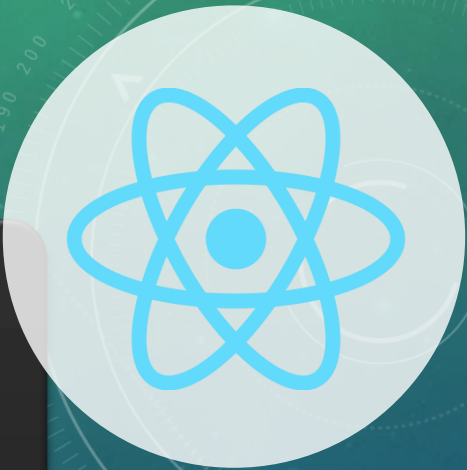


# REACT - SEBESSÉG





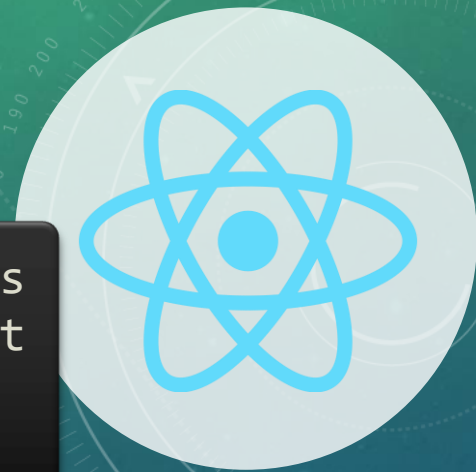
# REACT - JSX



```
class ShoppingList extends React.Component {  
  render() {  
    return (  
      <div className="shopping-list">  
        <h1>Shopping List</h1>  
        <ul>  
          <li>Instagram</li>  
          <li>WhatsApp</li>  
          <li>Oculus</li>  
        </ul>  
      </div>  
    );  
  }  
}
```

```
return React.createElement('div', {className: 'shopping-list'},  
  React.createElement('h1', /* ... h1 children ... */),  
  React.createElement('ul', /* ... ul children ... */) );
```

# REACT — CLASS VS FUNCTION



```
class Tool extends React.Component {  
  render() {  
    return (  
      <h1> {this.props.name} </h1>  
    );  
  }  
}
```

Class  
component

```
function Tool(props) {  
  return (  
    <h1> {props.name} </h1>  
  );  
};
```

Function  
component

```
const Tool = (props) => {  
  return (  
    <h1> {props.name} </h1>  
  );  
};
```

Function  
component

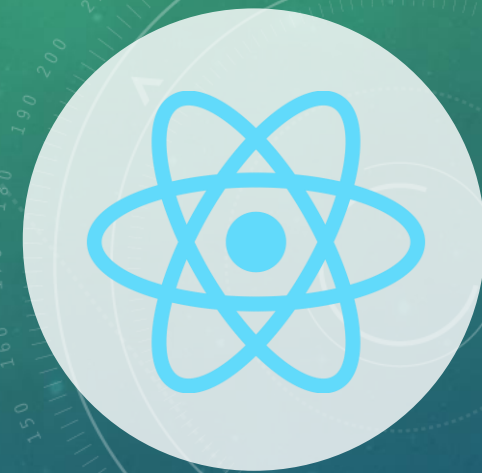
# REACT - PROPS

```
import Tool from "../Tool";  
const Example = () => {  
  return (  
    <>  
      <div className="app">  
        <Tool name="Mndwrk" />  
      </div>  
    </>  
  );  
};
```

Example.jsx

```
const Tool = (props) => {  
  return (  
    <h1>  
      {props.name}  
    </h1>  
  );  
};
```

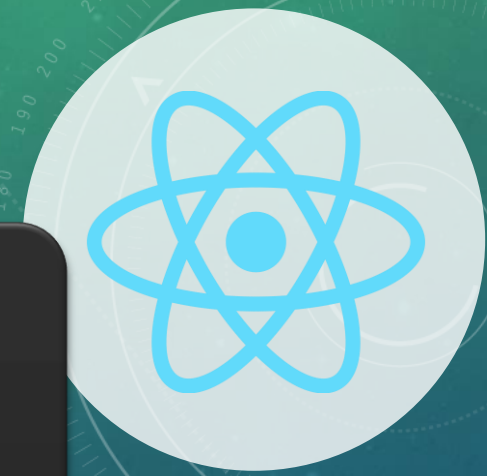
Tool.jsx



**Mndwrk**



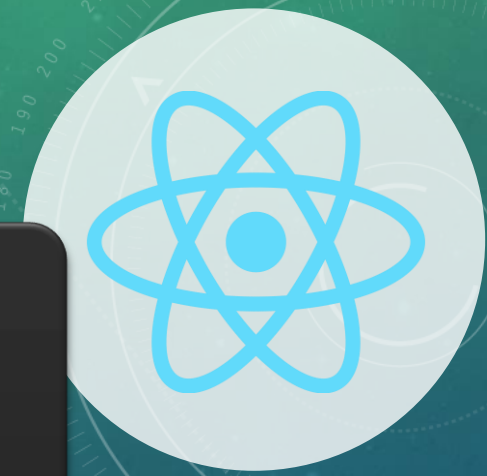
# REACT STATE



```
class Tool extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = { value: null, };  
  }  
  
  render() {  
    return (  
      <button  
        onClick={() => this.setState({value: 'Mndwrk'})} >  
        {this.state.value}  
      </button>  
    );  
  }  
}
```

Mndwrk

# REACT HOOKS



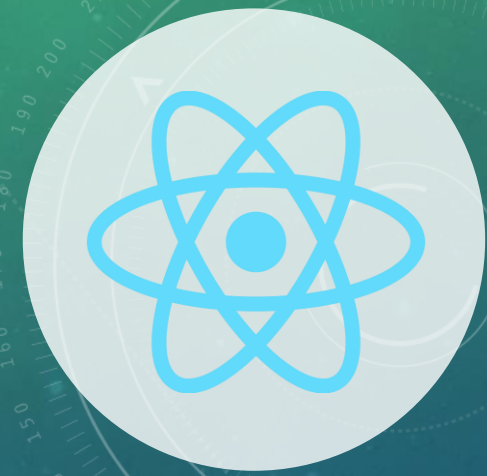
```
import React, { useState } from 'react';

const Tool = () => {
  const [value, setValue] = useState(null);

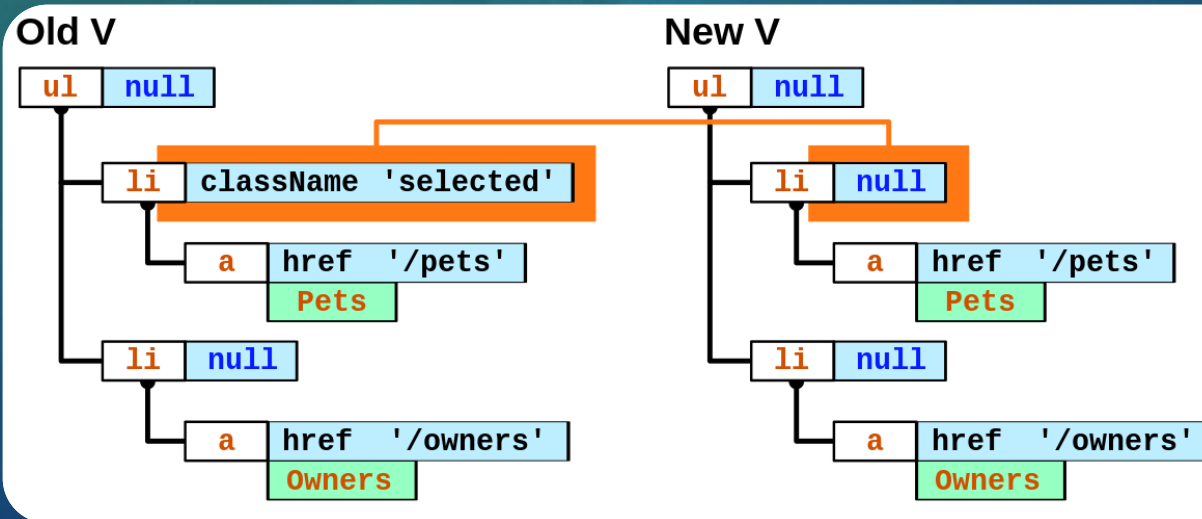
  return (
    <button onClick={() => setValue('Mndwrk')} >
      {value}
    </button>
  );
}
```

Mndwrk

# REACT ÉLETCIKLUS



- Egy komponens élettartamában lévő „eseményekre” iratkozhatunk fel
  - `ShouldComponentUpdate`
  - `componentDidMount`
  - `componentWillUnmount`
  - `render`
  - stb.

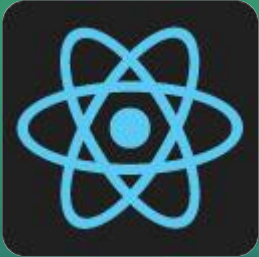


## Real DOM

```
<ul>
  <li> class="selected">
    <a href="/pets">Pets</a>
  </li>
  <li>
    <a href="/owners">Owners</a>
  </li>
</ul>
```



# REACT ÖKOSZISZTÉMA



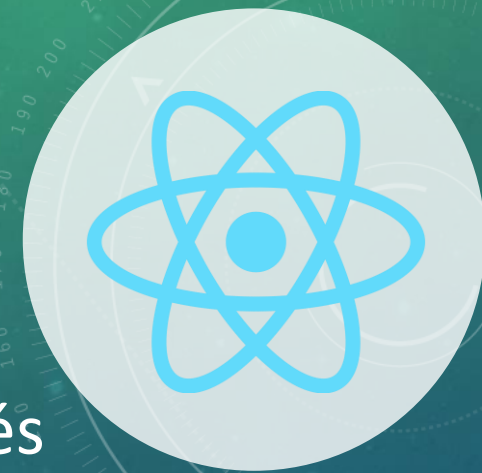
React DevTools: böngésző kiegészítő React state és props megtekintéséhez



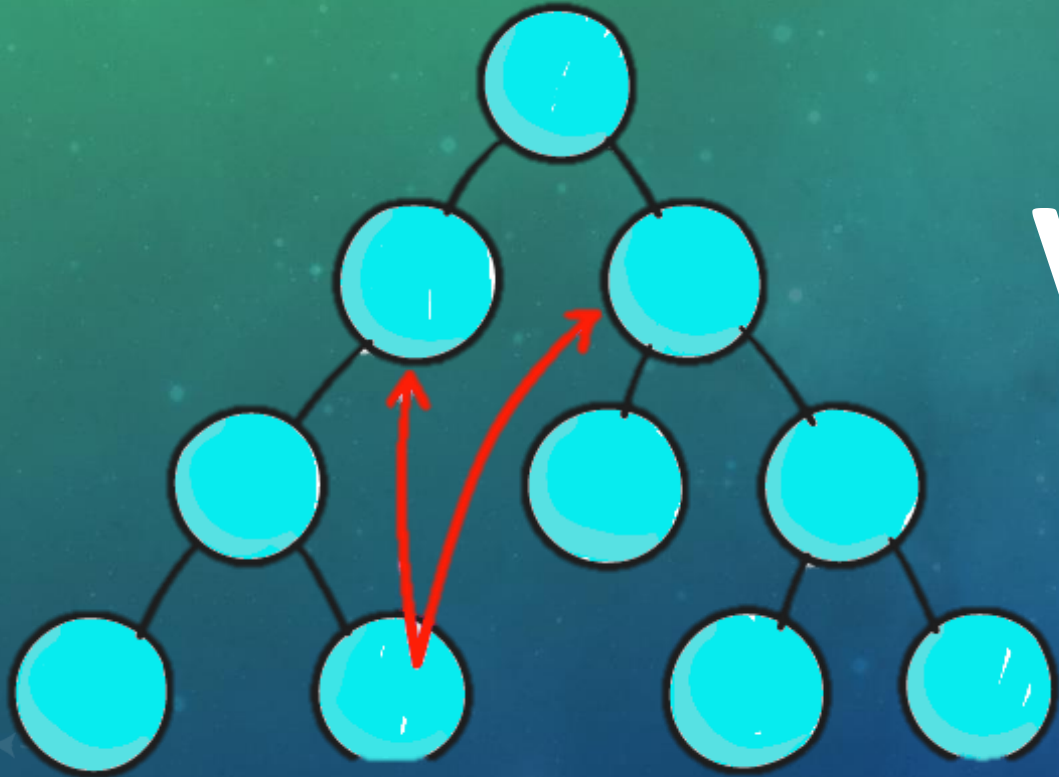
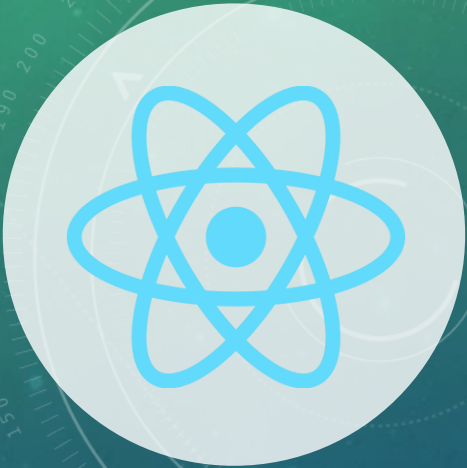
Redux: state kezelést egységesíti, struktúrálja és menedzseli



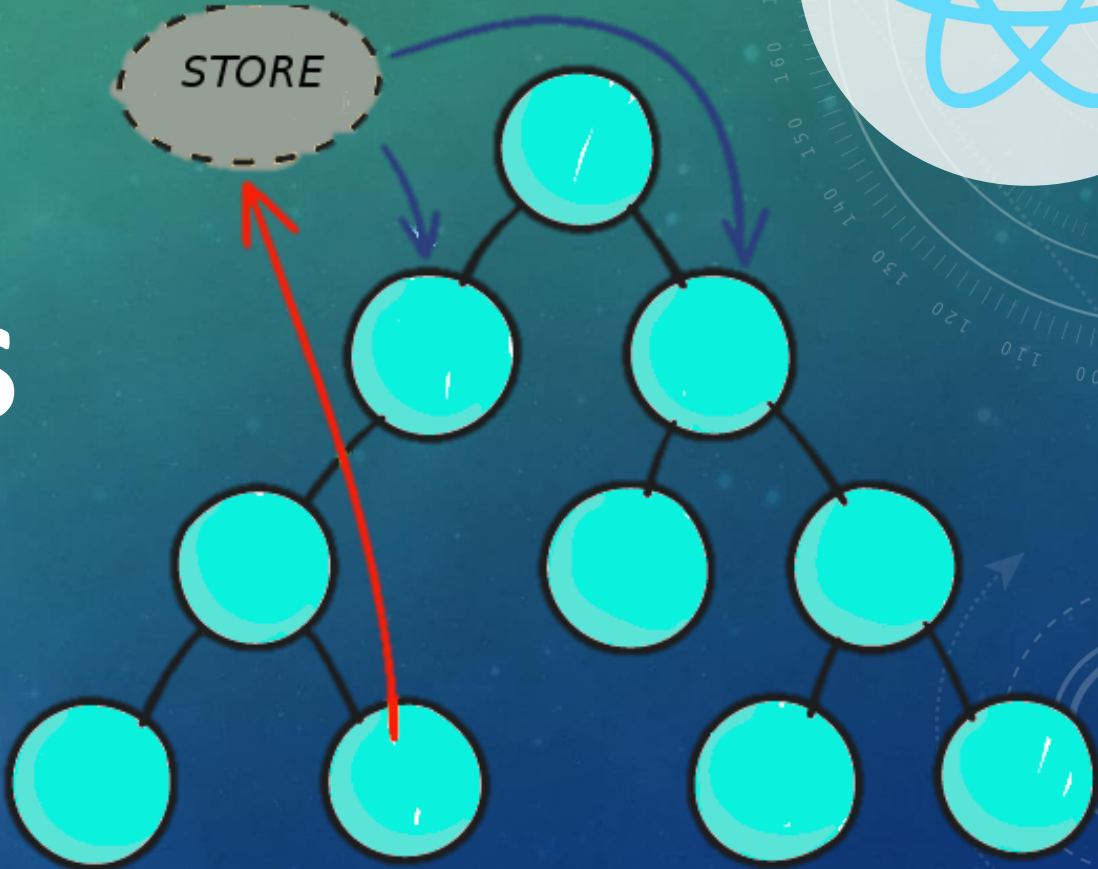
Create React App: hivatalosan támogatott React alkalmazás „indító” konfiguráció nélkül



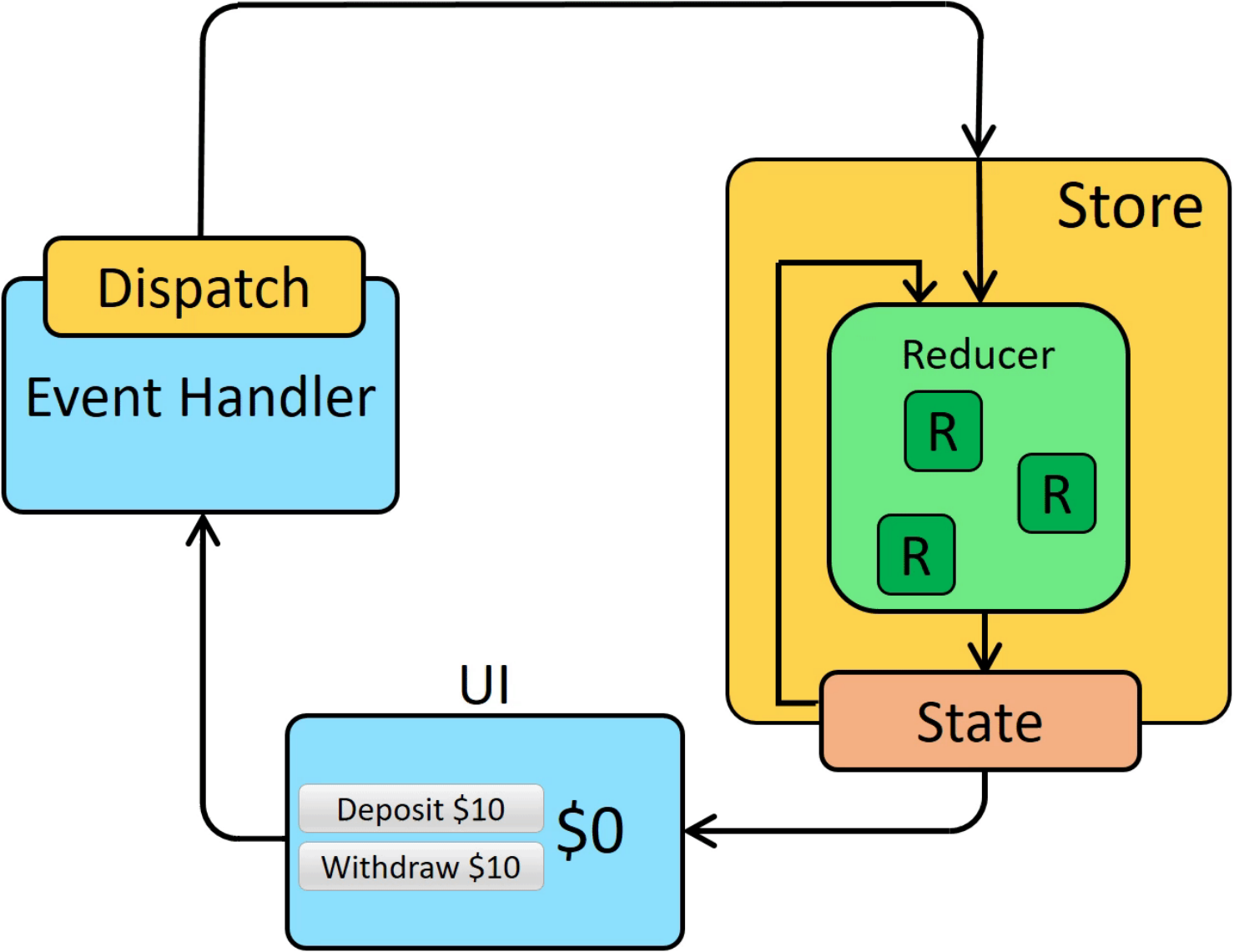
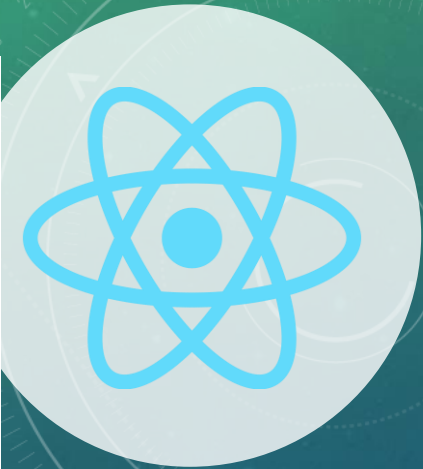
# REDUX



VS



# REDUX





# REDUX

```
const INITIAL_STATE = { name: '', ... }

export default (state = INITIAL_STATE,
action={}) => {
  switch(action.type) {
    case "SET_DATA":
      return { ...state, ...action.content };
    default: return state;
  }
};
```

reducer.js

```
import { createStore, applyMiddleware } from "redux";
import thunk from 'redux-thunk';
import reducer from "./reducer";

const store = createStore(
  reducer,
  applyMiddleware(thunk)
);
export default store;
```

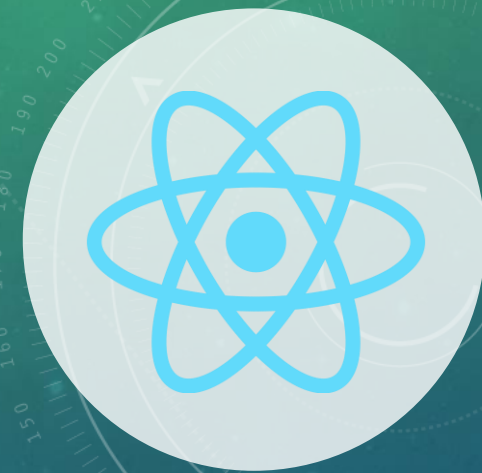
store.js

```
const setData = (content) => {
  return { type: "SET_DATA", content }
}

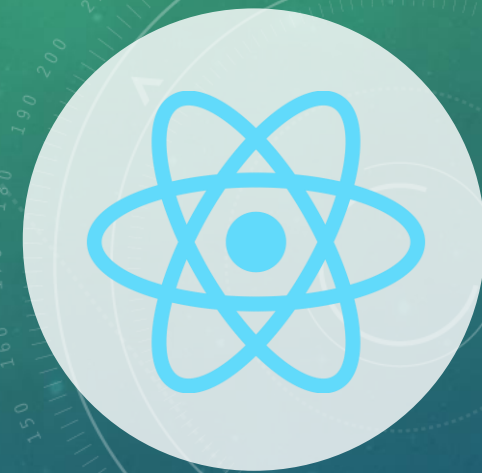
const appendData = (obj) => {
  return (dispatch) => {
    dispatch(setData(obj));
  }
}

export { appendData }
```

action.js



# REDUX



```
import React from "react";
import { connect } from "react-redux";
import { appendData } from "../action";

class App extends React.Component {
  componentDidMount() {
    this.props.appendData({ name: name, ... });
  }
  render() { return (<> ... </>); }
}

const mapDispatchToProps = { appendData }
const mapStateToProps = state => ( {
  name: state.name, ...
});
export default connect(mapStateToProps,
mapDispatchToProps)(Application);
Application.jsx
```

```
import ReactDOM from "react-dom";
import { Provider } from "react-redux";

const rootElement =
  document.getElementById("root");

ReactDOM.render(
  <Provider store={store}>
    <App/>
  </Provider>,
  rootElement
);
index.jsx
```

# CREAT REACT APP

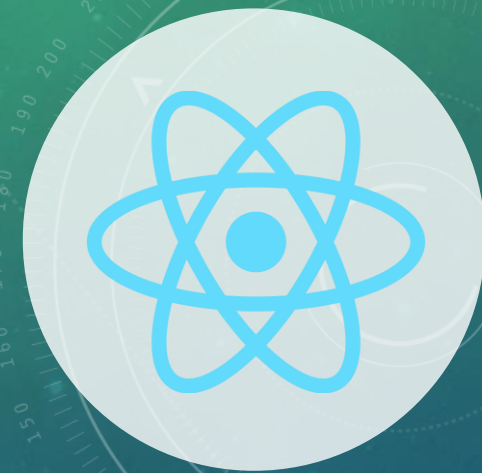
```
npm init react-app my-app
```

```
npm test
```

```
npm run build
```

```
npm start
```

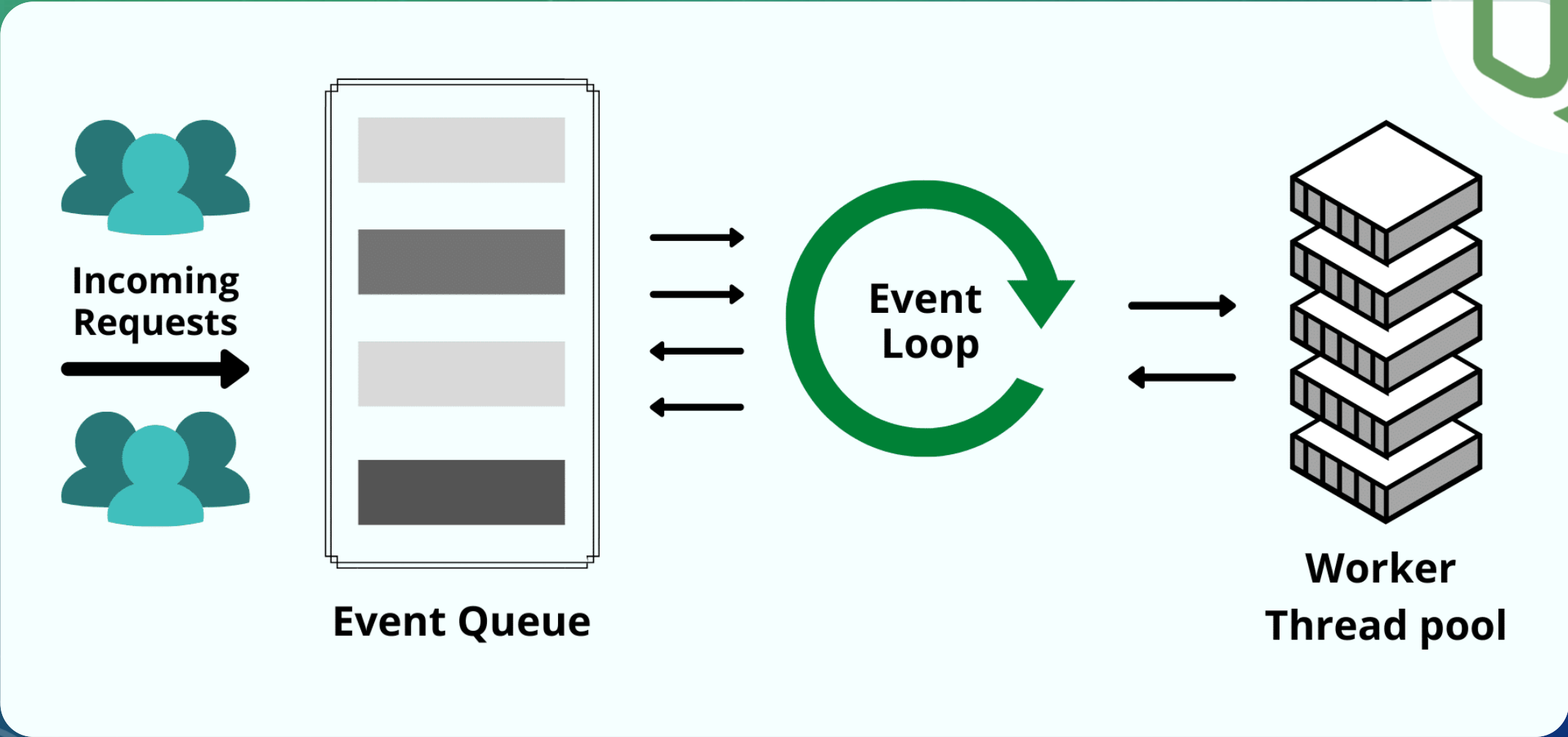
```
my-app
├── README.md
├── node_modules
├── package.json
├── .gitignore
├── public
│   ├── favicon.ico
│   ├── index.html
│   ├── logo192.png
│   ├── logo512.png
│   ├── manifest.json
│   └── robots.txt
└── src
    ├── App.css
    ├── App.js
    ├── App.test.js
    ├── index.css
    ├── index.js
    ├── logo.svg
    ├── serviceWorker.js
    └── setupTests.js
```







- Platformfüggetlen, nyílt forráskódú JavaScript futtatókörnyezet
- A V8 JavaScript Engine-t használja
- Egységesítette a webalkalmazások készítésének nyelvét
- Tervezésekor a következők voltak a fő szempontok:
  - Áteresztőképesség: üzenetkézbesítés mértéke (requests/second)
  - Skálázhatóság: növekvő mennyiségű munkát is tudja kezelni
- Ezt eseményvezérelt architektúrával éri el, ami aszinkron I/O-ra képes



# NODE.JS — BLOCKING VS NON-BLOCKING



- Eseményvezérelt programozást viszi a webszerverekre
- A fejlesztők szálkezelés nélkül is létrehozhatnak skálázható alkalmazásokat callback-ek használatával a feladat befejezésének jelzésére

```
const fs = require("fs");  
const data = fs.readFileSync("/file.md");
```

szinkron / bloking

```
const fs = require("fs");  
  
fs.readFile("/file.md", (err, data) => {  
  if (err) throw err;  
});
```

aszinkron /  
non-bloking



# NODE.JS — BLOCKING VS NON-BLOCKING



```
const fs = require("fs");

fs.readFile("/file.md", (err, data) => {
  if (err) throw err;
  console.log(data);
});

fs.unlinkSync("/file.md");
```



- npm – a Node.js alap csomagkezelője, viszont más alkalmazásokhoz is használható
- Command line eszköz és online adatbázis
  - <https://www.npmjs.com>
- `package.json` fájlból telepíti az összes függőséget

```
npm install
```



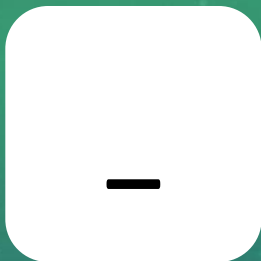
```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```





Express application generator: hivatalos Express alkalmazás „indító”



Express: RESTful API készítéséhez használható keretrendszer Node.js mellé



Passport: autentikációs middleware Node.js alkalmazásokhoz, Express mellé könnyen beilleszthető

# NODE.JS - NVM



- Linux-ra és Windows-ra is elérhető Node verziókezelő

```
nvm install latest
```

```
nvm list
```

```
nvm use 14.20.0
```

- Robusztus routing, nagy teljesítmény, HTTP helperek

```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```



# EXPRESS — KELL?

ex

```
const http = require('http');
const fs = require('fs');

function reqListener(req, res) {
  const url = req.url;
  const method = req.method;

  if (url === '/') {
    res.write('<html>');
    //...
    res.write('</html>');
    return res.end();
  }

  if (url === '/message' && method === 'POST') {
    const body = [];
    req.on('data', (chunk) => {
```

# EXPRESS — KELL?

```
    console.log(chunk);
    body.push(chunk);
  })

  req.on('end', () => {
    const parsedBody = Buffer.concat(body).toString();
    const message = parsedBody.split('=')[1];
    fs.writeFile('message.txt', message, (err) => {
      res.statusCode = 302;
      res.setHeader('Location', '/');
      return res.end();
    });
  });
}

const server = http.createServer(reqListener);
server.listen(3000);
```



ex

# EXPRESS — KELL!

ex

```
const express = require('express')
const app = express()

app.get('/', (req, res) => {
  res.send('<html>...</html>')
})

app.post('/message', (req, res) => {
  const message = req.body.message;
  fs.writeFile('message.txt', message, (err) => {
    res.redirect('/')
  });
})
```



# EXPRESS APPLICATION GENERATOR

```
npm install -g express-generator
```

```
express
```

```
npm install
```

```
npm start
```

```
├── app.js
├── bin
│   └── www
├── package.json
├── public
│   ├── images
│   ├── javascripts
│   └── stylesheets
│       └── style.css
├── routes
│   ├── index.js
│   └── users.js
└── views
    ├── error.jade
    ├── index.jade
    └── layout.jade
```



ex

# FORRÁSOK

- <https://en.wikipedia.org>
- <https://reactjs.org>
- <https://github.com/facebook/react>
- <https://elm-lang.org>
- <https://create-react-app.dev>
- <https://redux.js.org>
- <https://nodejs.org>
- <https://kinsta.com>
- <https://progressivecoder.com>
- <https://medium.com>
- <https://scriptverse.academy>