

# Anatomy of an Application

---



**Roland Guijt**

Freelance consultant and trainer

@rolandguijt roland.guijt@gmail.com



# A Component

```
const Banner = () => <h1>This is a banner</h1>;
```



# Components and JSX

**Components are JavaScript functions  
that return jsx**

**JSX looks like HTML but it isn't**

**An alternative way to write JavaScript**

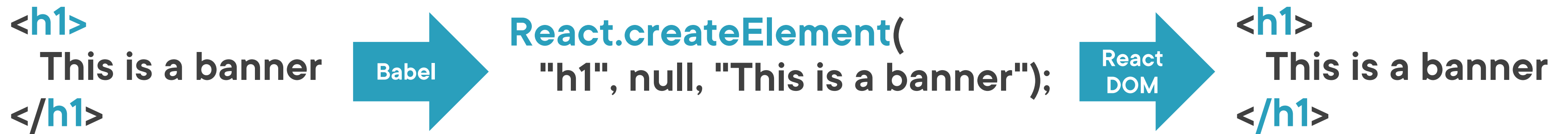
**JSX: JavaScript eXtension**

**Translated to JavaScript by a tool**

**Babel**



# JSX to Markup



# Separation of Concerns

**react**

**react-dom**



## Another Component

```
const Greeting = () => (  
  <div>  
    <Banner />  
    <h2 className="highlight">Greetings!</h2>  
  </div>  
) ;
```



# Built-in Components

**Correspond to DOM elements**

**Different output targets use different built-in components**

**Are camelCased**

**Custom components are PascalCased**



# Class Components

```
class Greeting extends React.Component {  
  render() {  
    return (  
      <div>  
        <Header />  
        <h2 className="highlight">Hi there!</h2>  
      </div>  
    )  
  }  
}
```





# Component Types

**React team recommends functions**

**New future React features only available  
in functions**

**Functions are less verbose and easier  
to manage**



# Putting It All Together

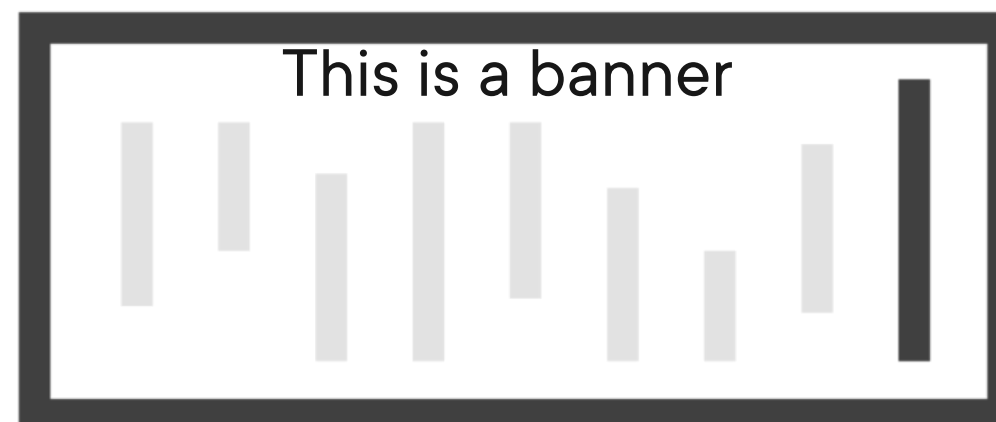
```
const Banner = () => <h1>This is a banner</h1>;
```



```
<div>  
  <Banner />  
  <Chart />  
</div>
```



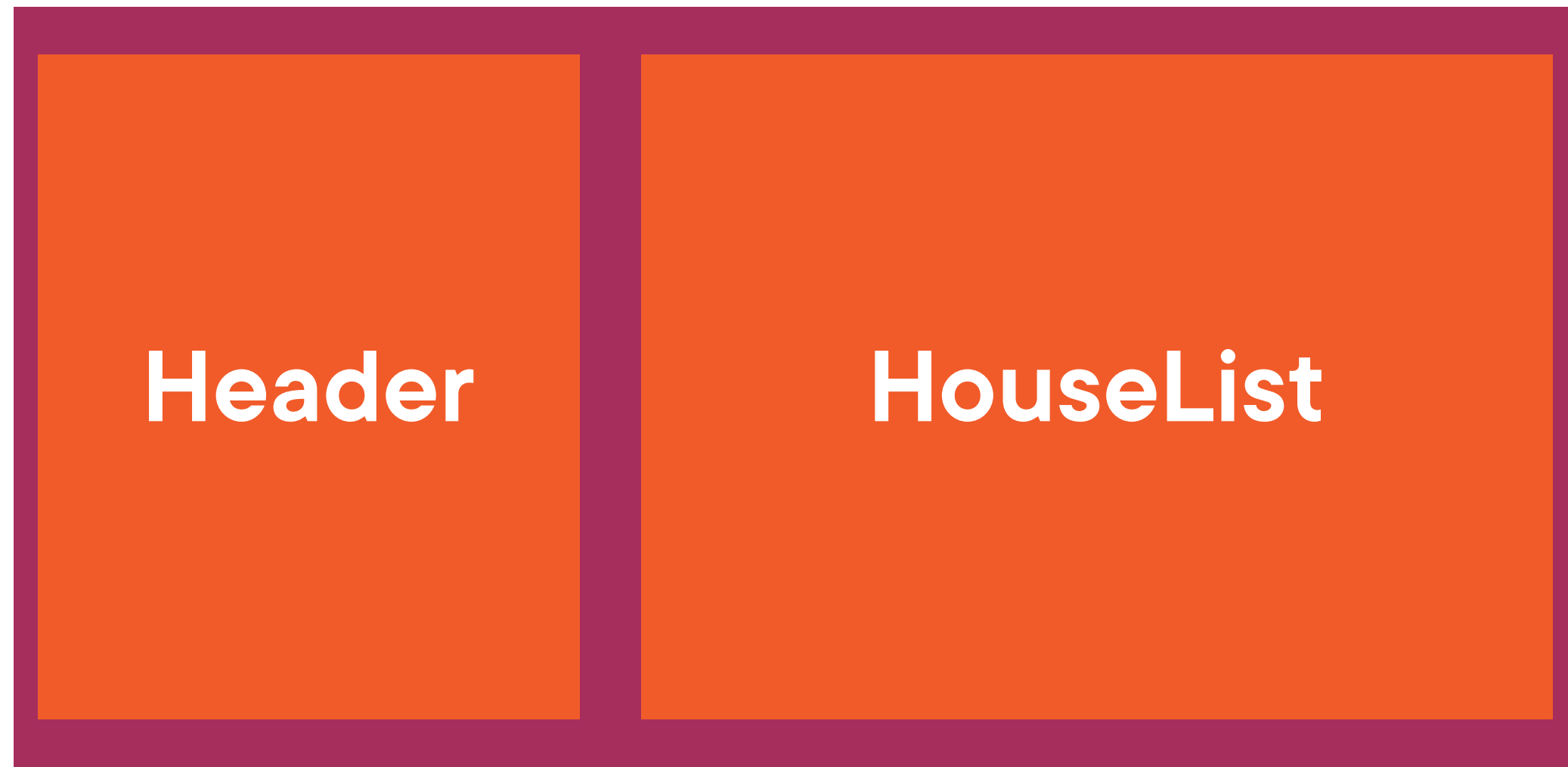
```
React.createElement( ...
```





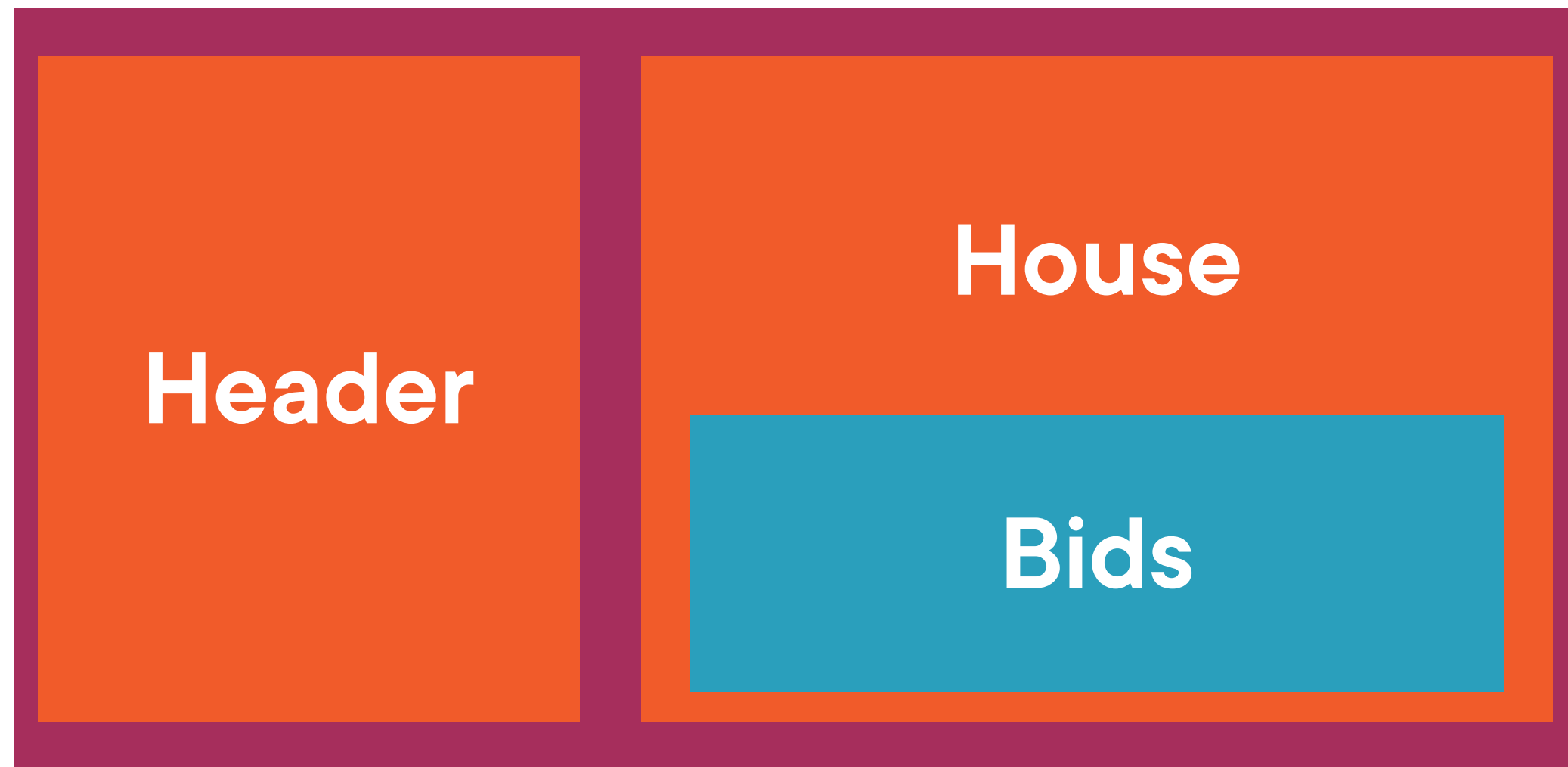
# Composition 1

App



# Composition 2

App



# Why Tooling is Needed

**Starting point**

**Transform JSX to JavaScript**

**Process JavaScript files**

**Run a development server**

**Automatically update browser when source files change**

**Create a production build**



# Options for Tooling

## Do It Yourself

### Use a ready-to-go development environment

- Create React App (CRA)
- Next.js



Next.js

**Development environment used in this course**

**Adds features**

**Goal: pure React**

**Next.js has optional server-side features**





React !== Next.js



Install Node.js  
Download from [nodejs.org](https://nodejs.org)



# Routing in Next.js

**Every file in pages directory is a "page"**

**Just a component**

**Renders when filename matches with url**

**/contact => contact.js**

**index.js renders at the root url**



# Modules



# Modules

module.js

```
const doSomething = () => {  
  ...  
};  
export { doSomething };
```

anotherModule.js

```
import { doSomething } from "./module";  
  
doSomething();
```



# Modules

module.js

```
const doSomething = () => {  
  ...  
};  
export default doSomething;
```

anotherModule.js

```
import do from "./module";  
  
do();
```



# React and Modules

banner.js

```
const Banner = () => (  
  <h1>  
    This is a banner  
  </h1>  
>;  
export default Banner;
```

greeting.js

```
import Banner from "./banner";  
  
const Greeting = () => (  
  <Banner />  
>;
```



# Reasons to Use Modules

**Code structure**

**Reusability**

**Encapsulation**

**Needed for bundling**





# Running the Finished Project

**Clone repository:**

**[https://github.com/RolandGuijt/  
ps-globomantics-reactfundamentals](https://github.com/RolandGuijt/ps-globomantics-reactfundamentals)**

**npm install**

**npm run dev**



# ESLint

**Detect problems**

**Code styling**

**Commonly used**

**Next.js: Built-in React ruleset**



Modify rules to your liking in  
the ESLint configuration file

