

Intro to Web Dev and React

3DC

Bryce Goh



Basic React Introduction

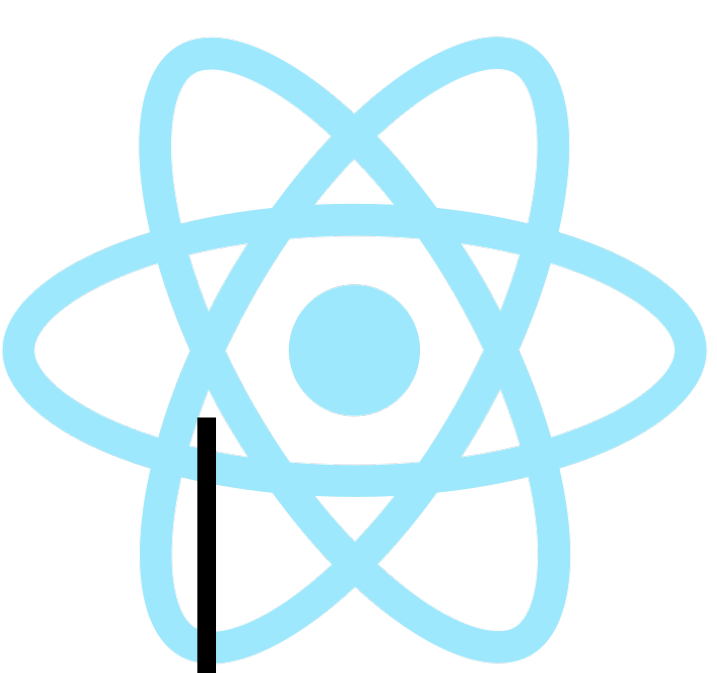
Basic overview of the web

Why react

Javascript Refreshers

Components and JSX

To-do App



Basic overview of the web



Browser

HTTP Request

GET

POST

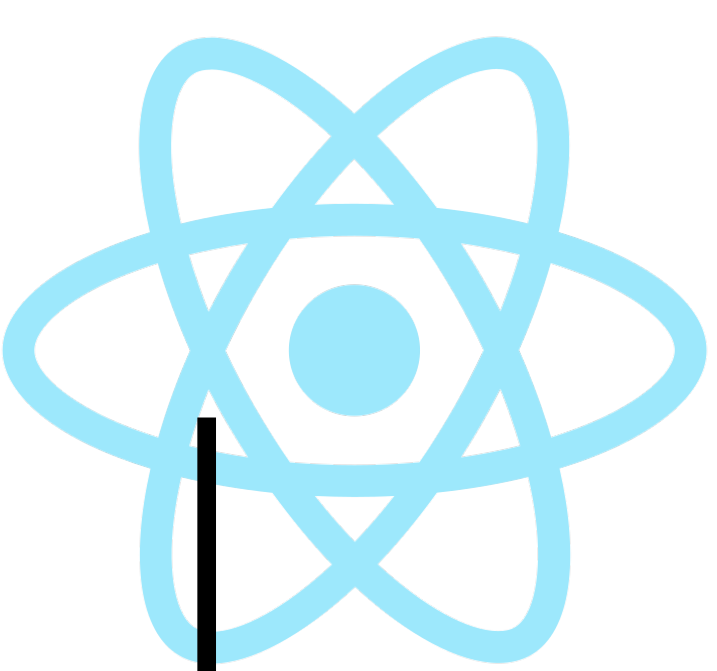
DELETE

PUT



Host Server

HTTP Response



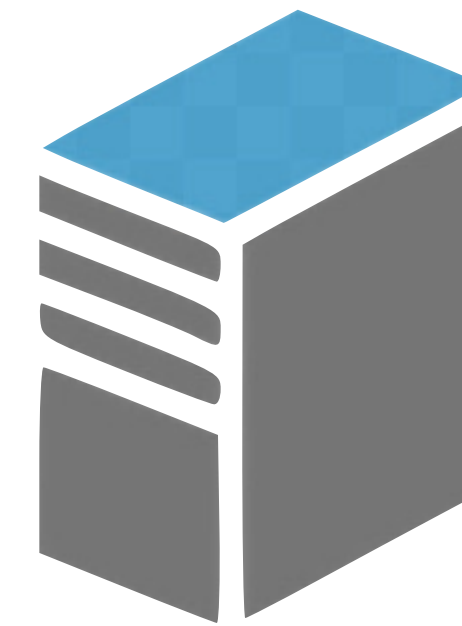
Basic overview of the web

www.sutd.edu.sg



Browser

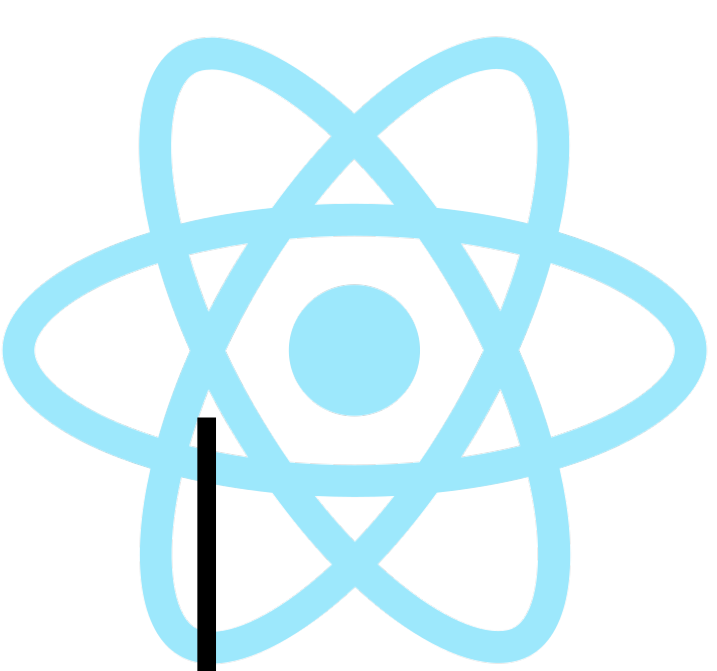
**HTTP Request
GET**



DNS Server

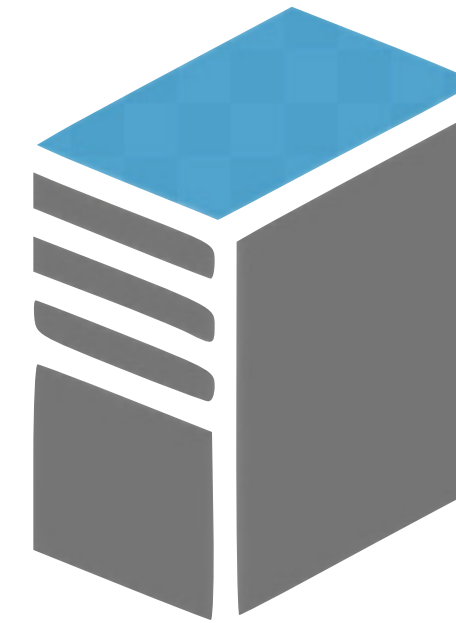


Host Server



Basic overview of the web

`www.sutd.edu.sg`



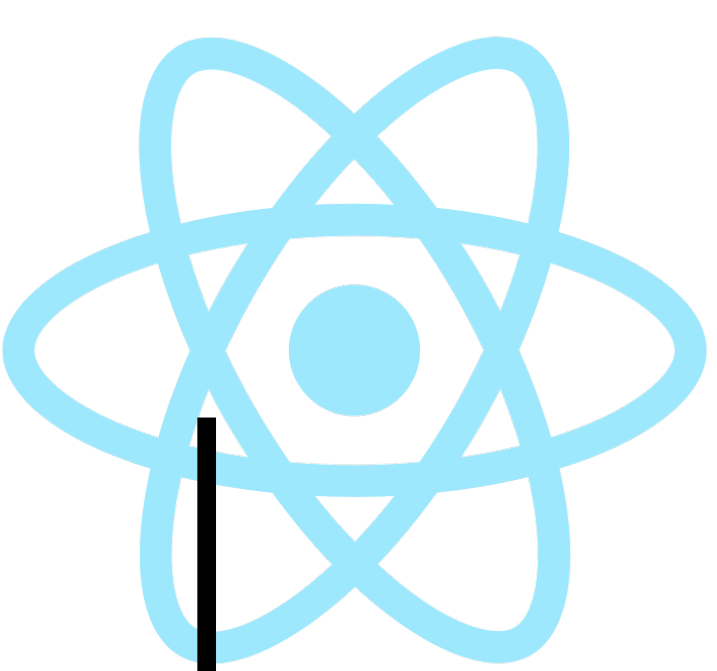
DNS Server



Browser



Host Server



Basic overview of the web

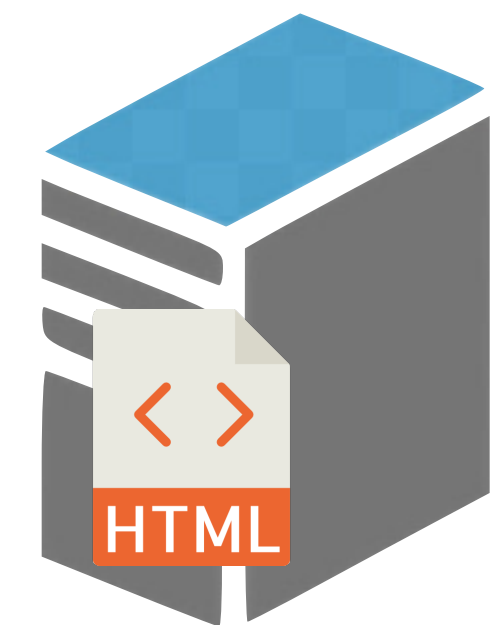


Browser

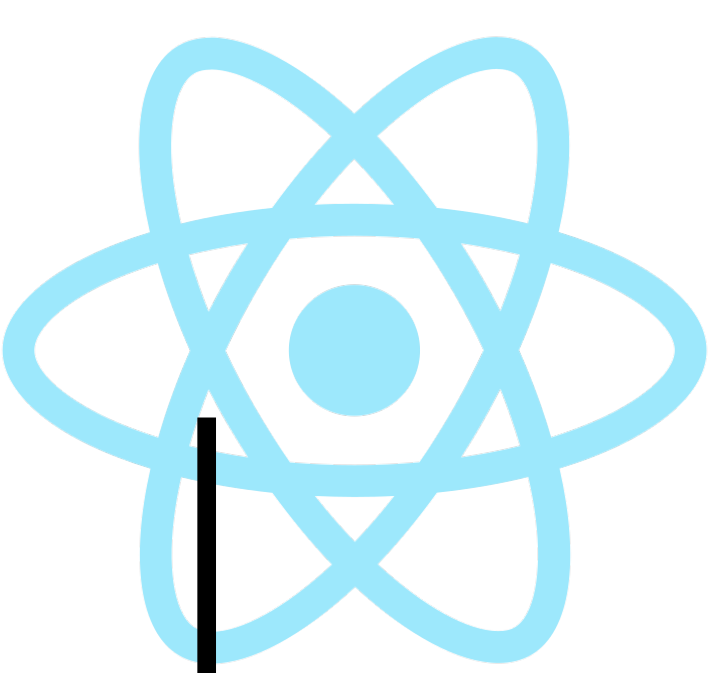


DNS Server

IP address



Host Server



Basic overview of the web

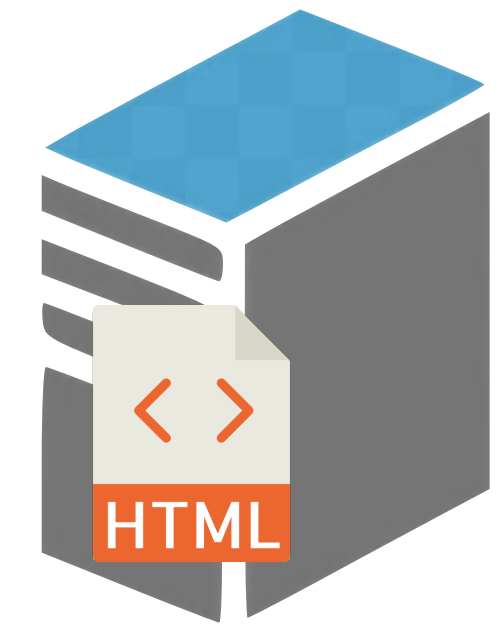


Browser

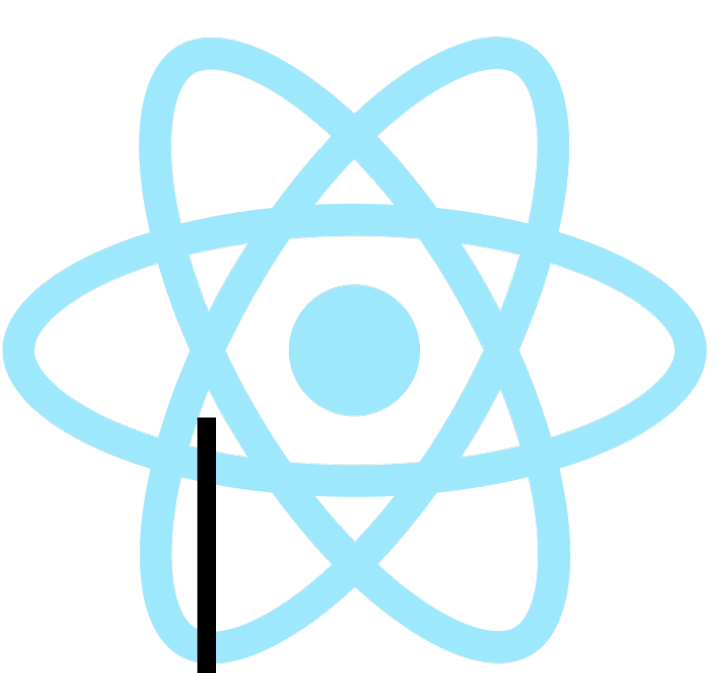


DNS Server

IP address



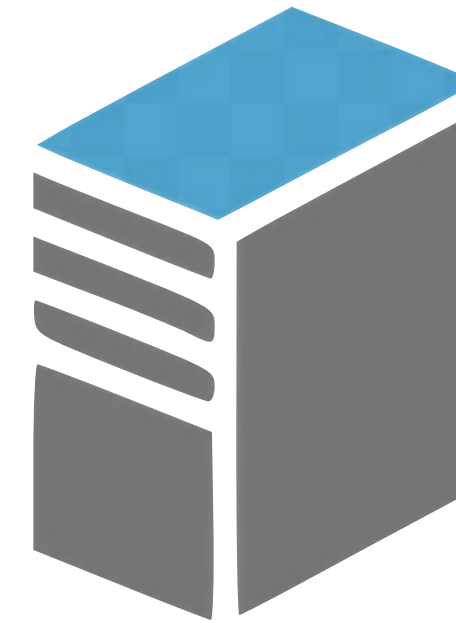
Host Server



Basic overview of the web

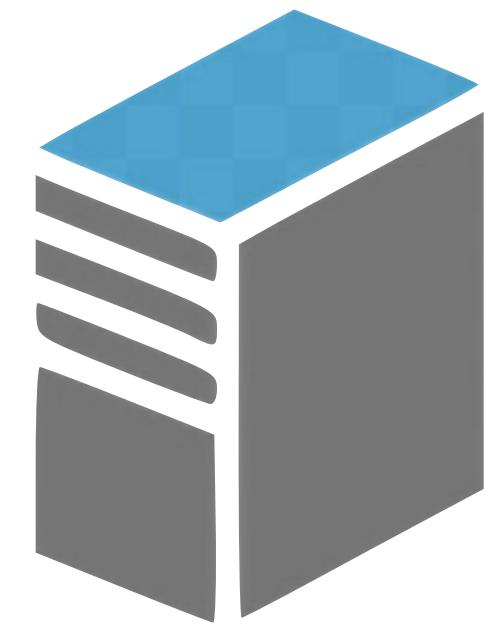


Browser

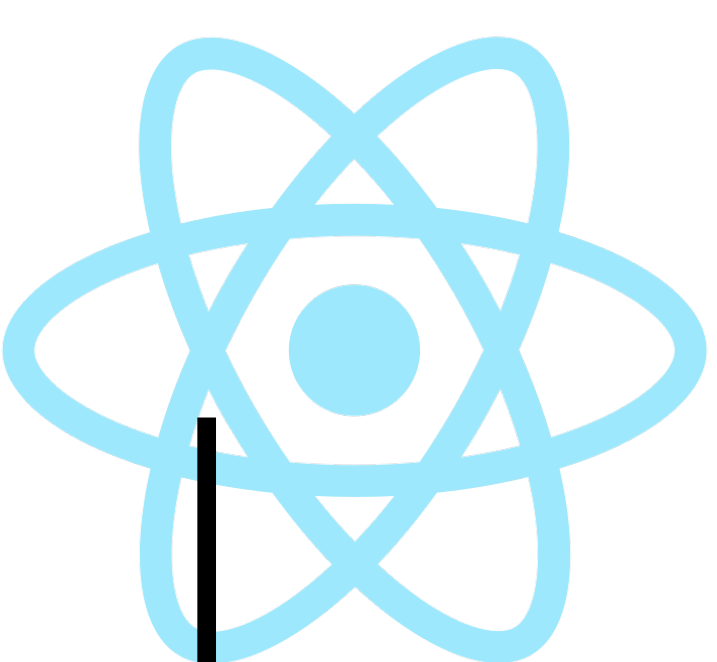


DNS Server

IP address



Host Server



Basic overview of the web



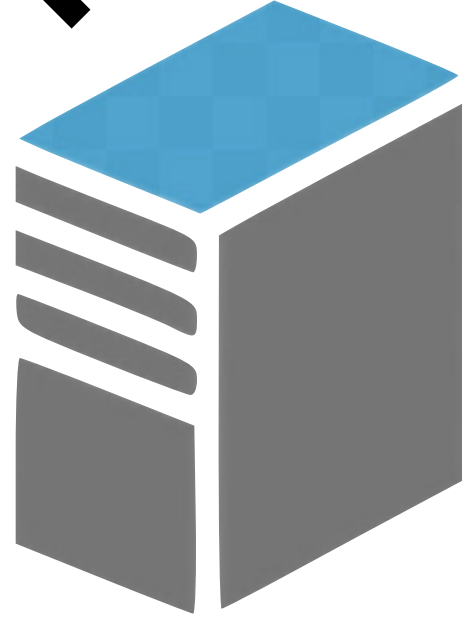
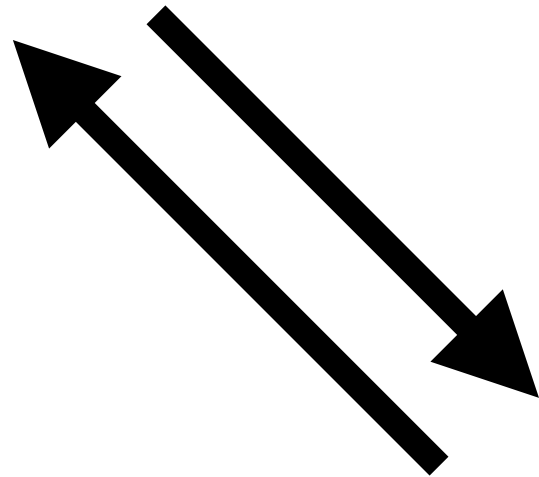
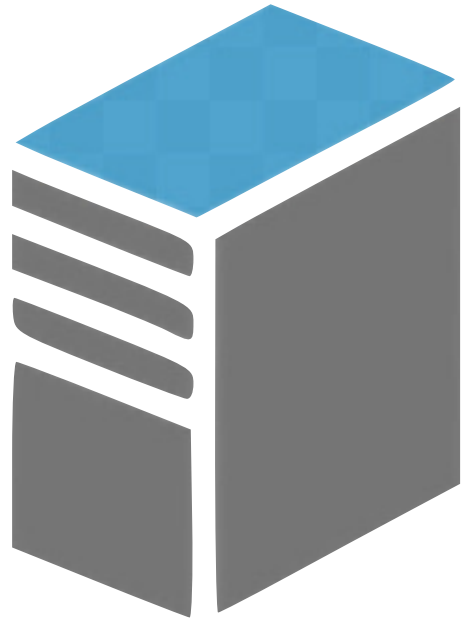
Browser

**HTTP Request
GET**



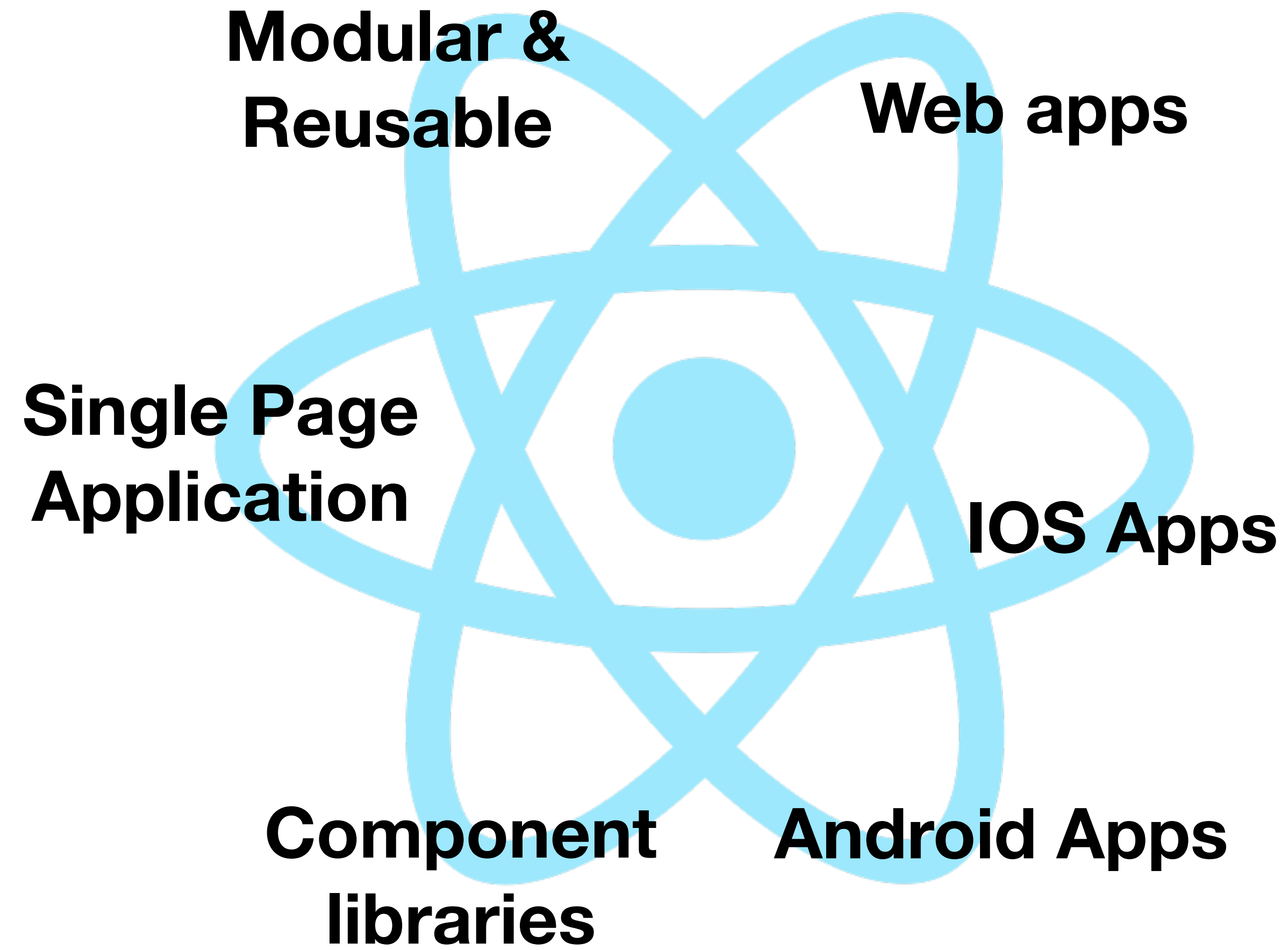
Response

API endpoint



Database

Why react



Javascript Refreshers

Defining Variables

variable_name = “hello world”

Const variableName = “hello world”
Let variableName = “hello world”

List/Array & Objects/ Dictionary

List = [....]

**Dictionary = {
Key: value
}**

Const list = [....]

**Const dictionary = {
Key: value
}**

Javascript Refreshers

Functions

Def function_name(argument):

...

...

return(...)

Const functionName = (argument) => {

...

...

return(...)

}

Loops

For x in list:

print(x)

list.map((x , index) => {

console.log(x)

}

)

HTML/CSS Refreshers

<div>

<div/>

HTML/CSS Refreshers

<div>

<input>

</input>

<button

</button>

</div>

HTML/CSS Refreshers

<div>

<input>

.className{

background-colour:red

}

</input>

<button

</button>

</div>

HTML/CSS Refreshers

<div>

<input>

```
.className{  
background-colour:red  
}
```

</input>

<button

```
.className{  
align: center  
}
```

</button>

</div>

REACT

Browser

Components

Components

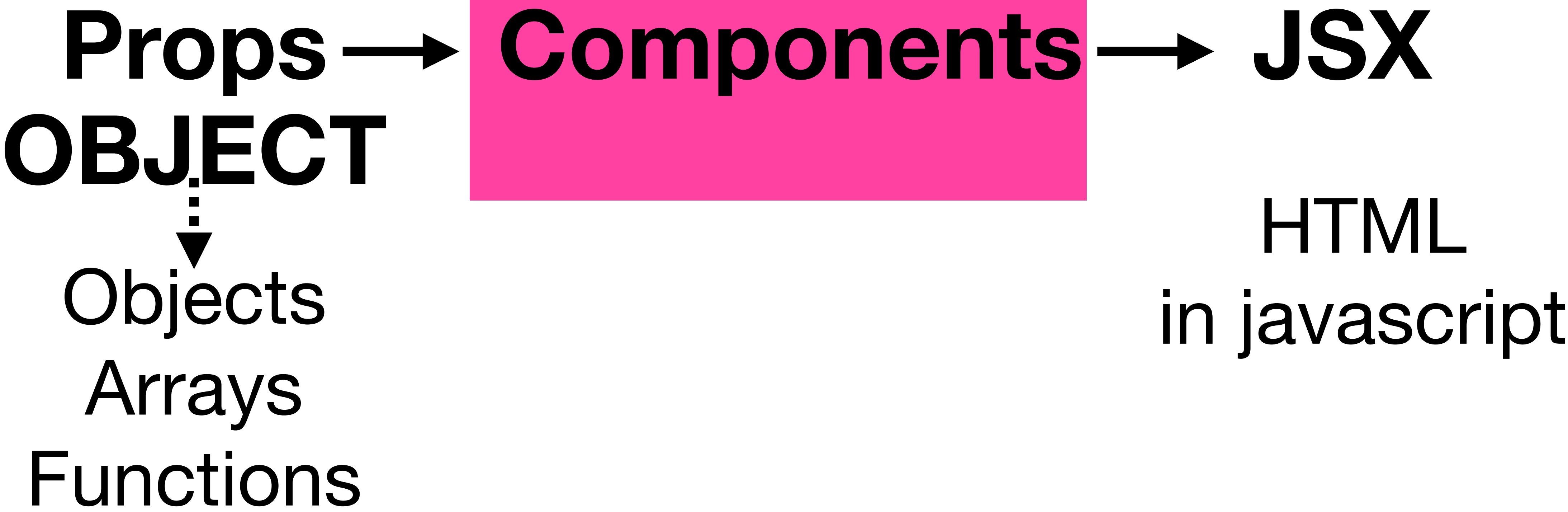
REACT

Function

Props → Components → JSX



Function



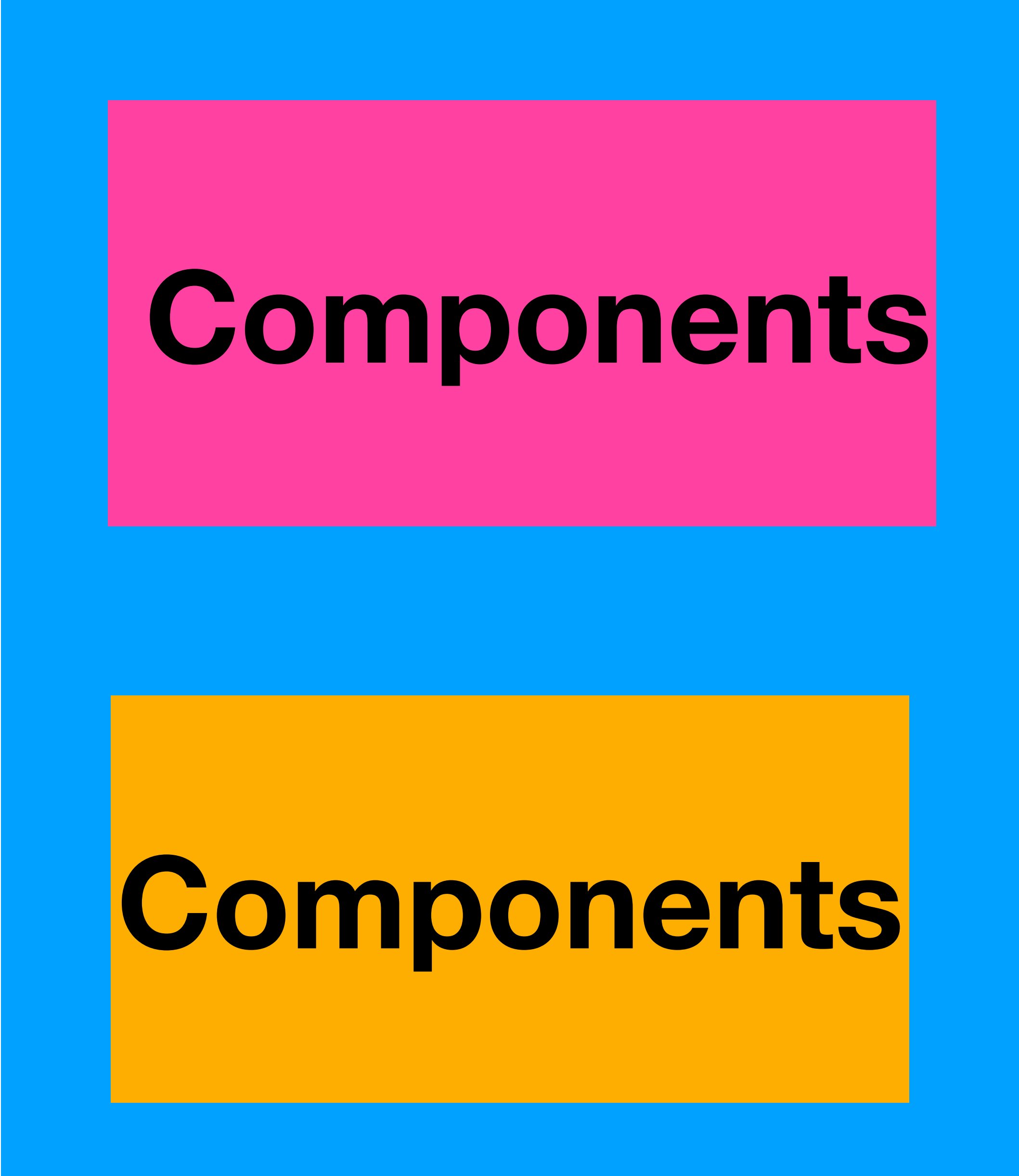
Props → **Components** → JSX

```
<div>
  Item 1
</div>
<div>
  Item 2
</div>
```

```
<div>
  {
    array.forEach( x =>
      <MySecondComponent
        firstProp = {x}
      />
    )
  }
</div>
```

```
Const MySecondComponent = (props) => {
  return(
    <div>
      props.firstProp
    </div>
  )
}
```

REACT



State



DEMO For today

State

INPUT FIELD

SAVE BUTTON



DEMO For today

State

INPUT FIELD

DATA

SAVE BUTTON

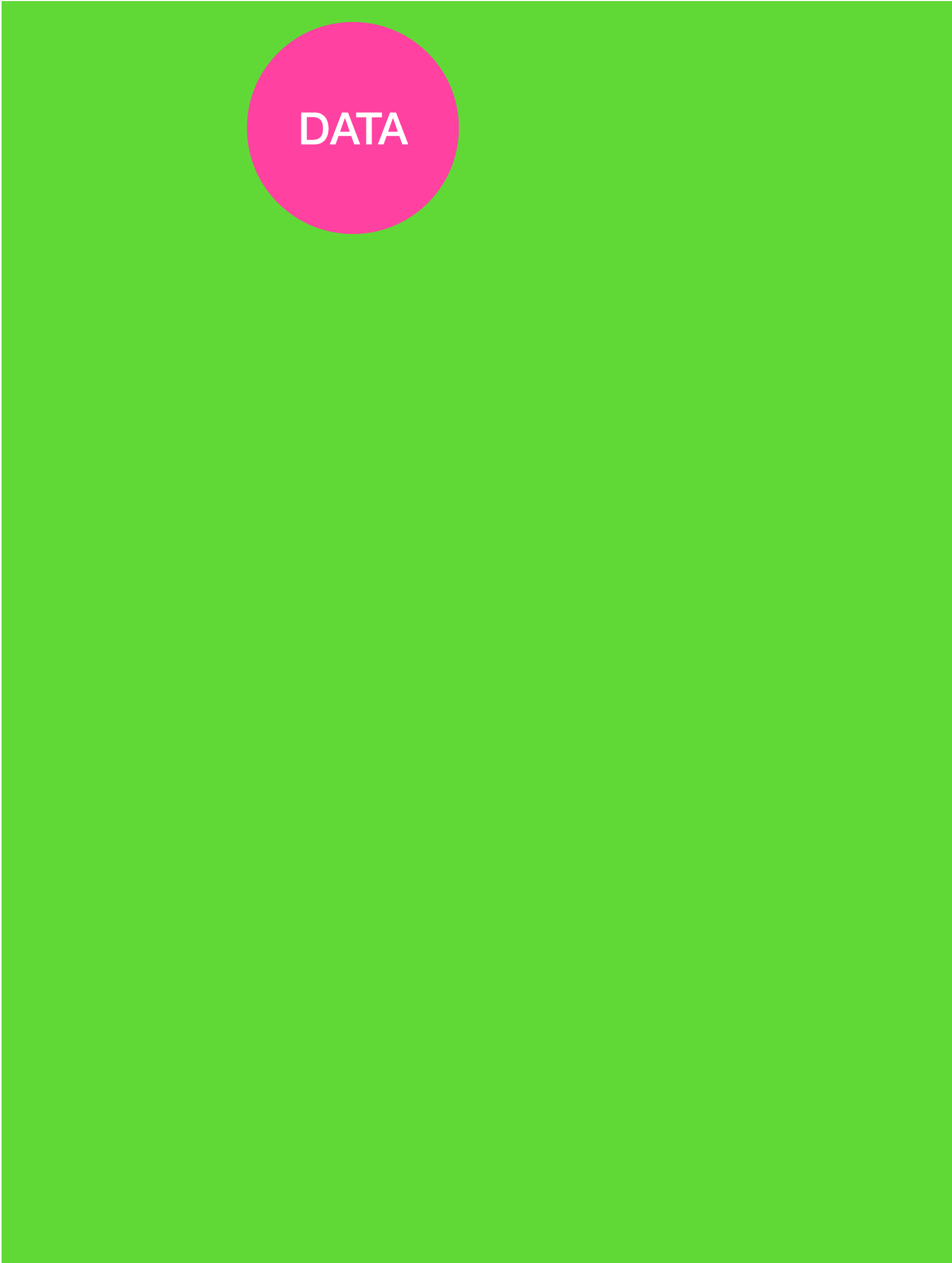


DEMO For today

INPUT FIELD

SAVE BUTTON

State

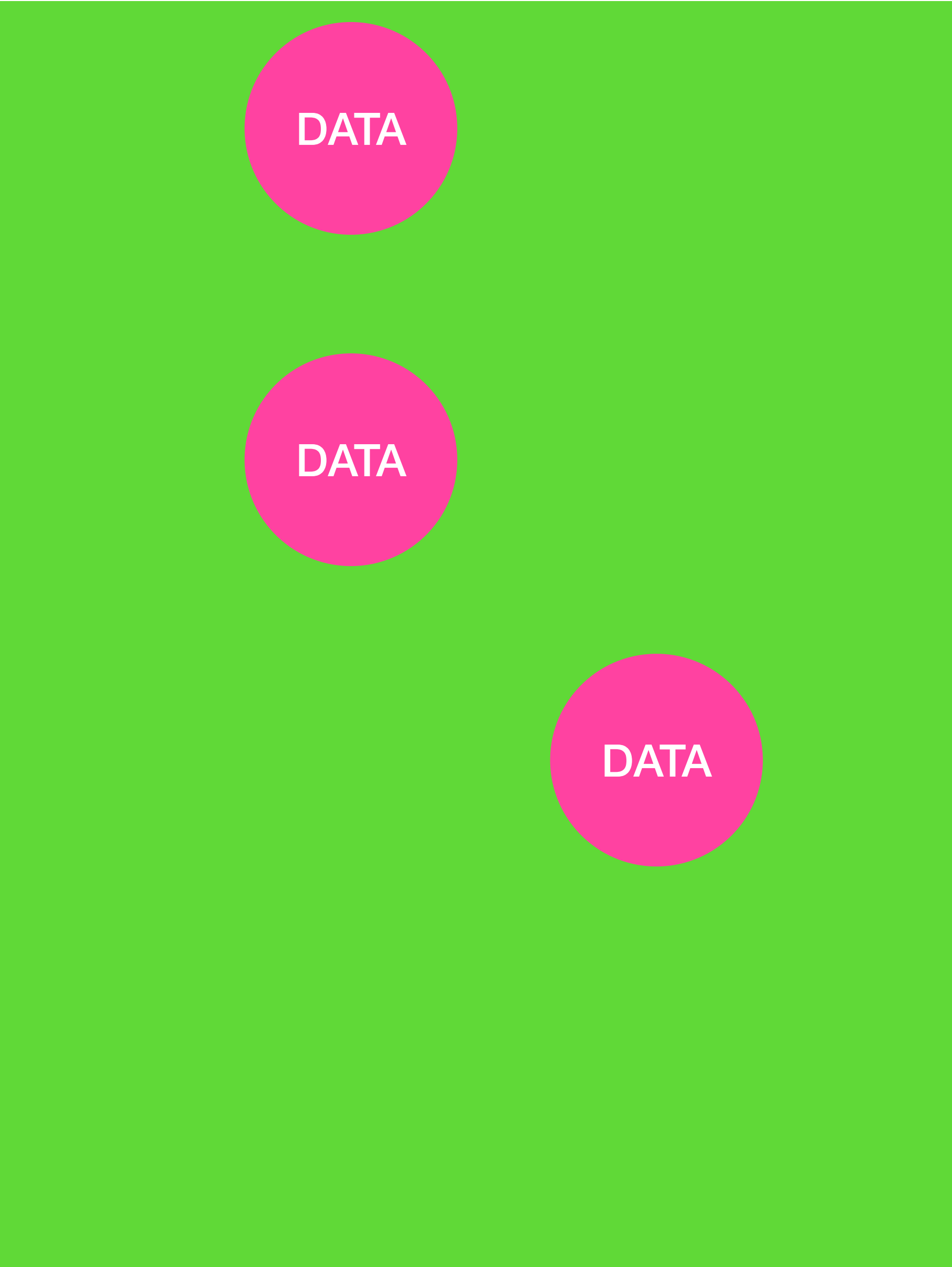


DEMO For today

INPUT FIELD

SAVE BUTTON

State



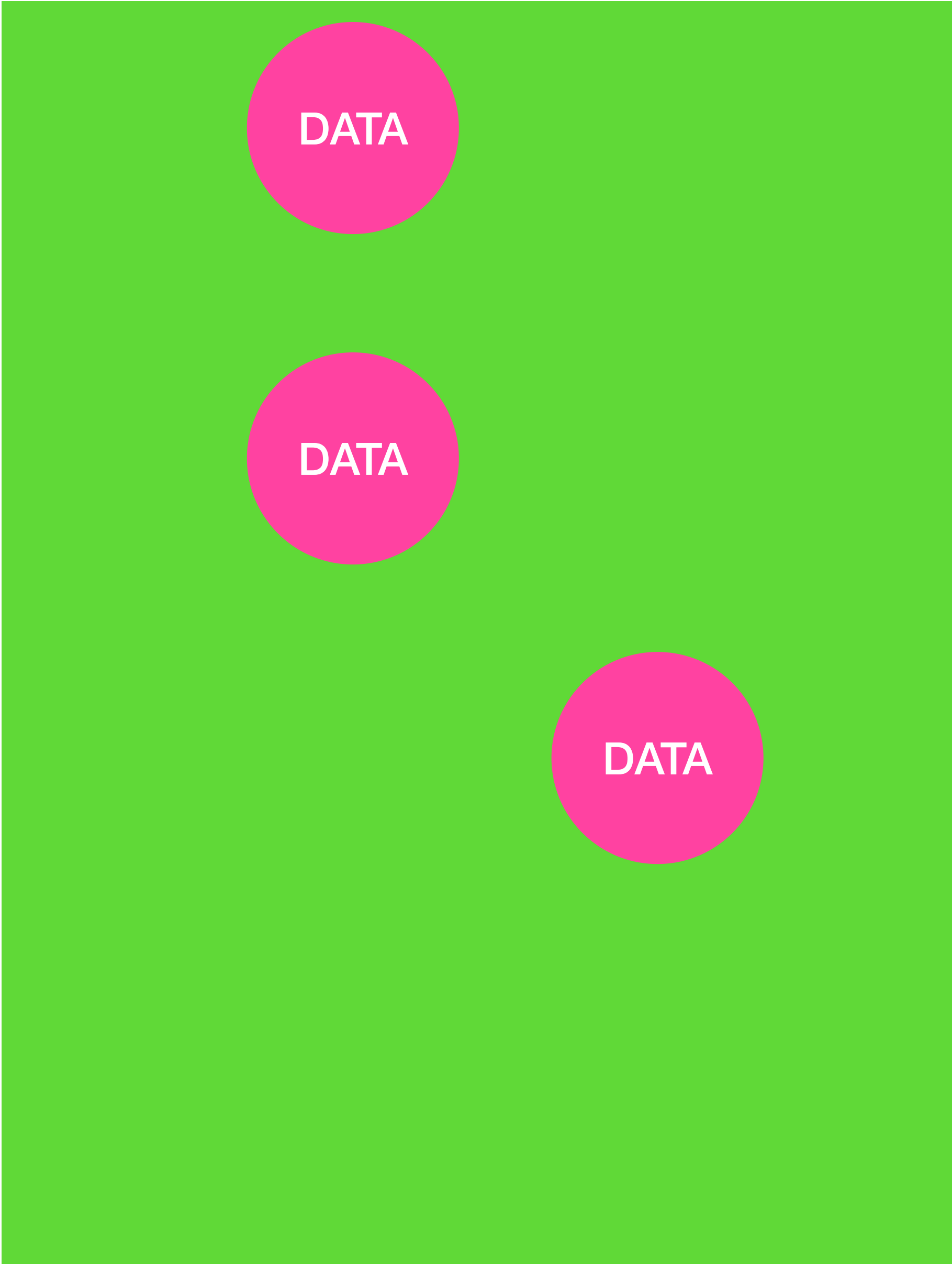
DEMO For today

INPUT FIELD

SAVE BUTTON

REMINDER ROW COMPONENT	DATA	DEL
REMINDER ROW COMPONENT	DATA	DEL
REMINDER ROW COMPONENT	DATA	DEL

State



DEMO For today

INPUT FIELD

SAVE BUTTON

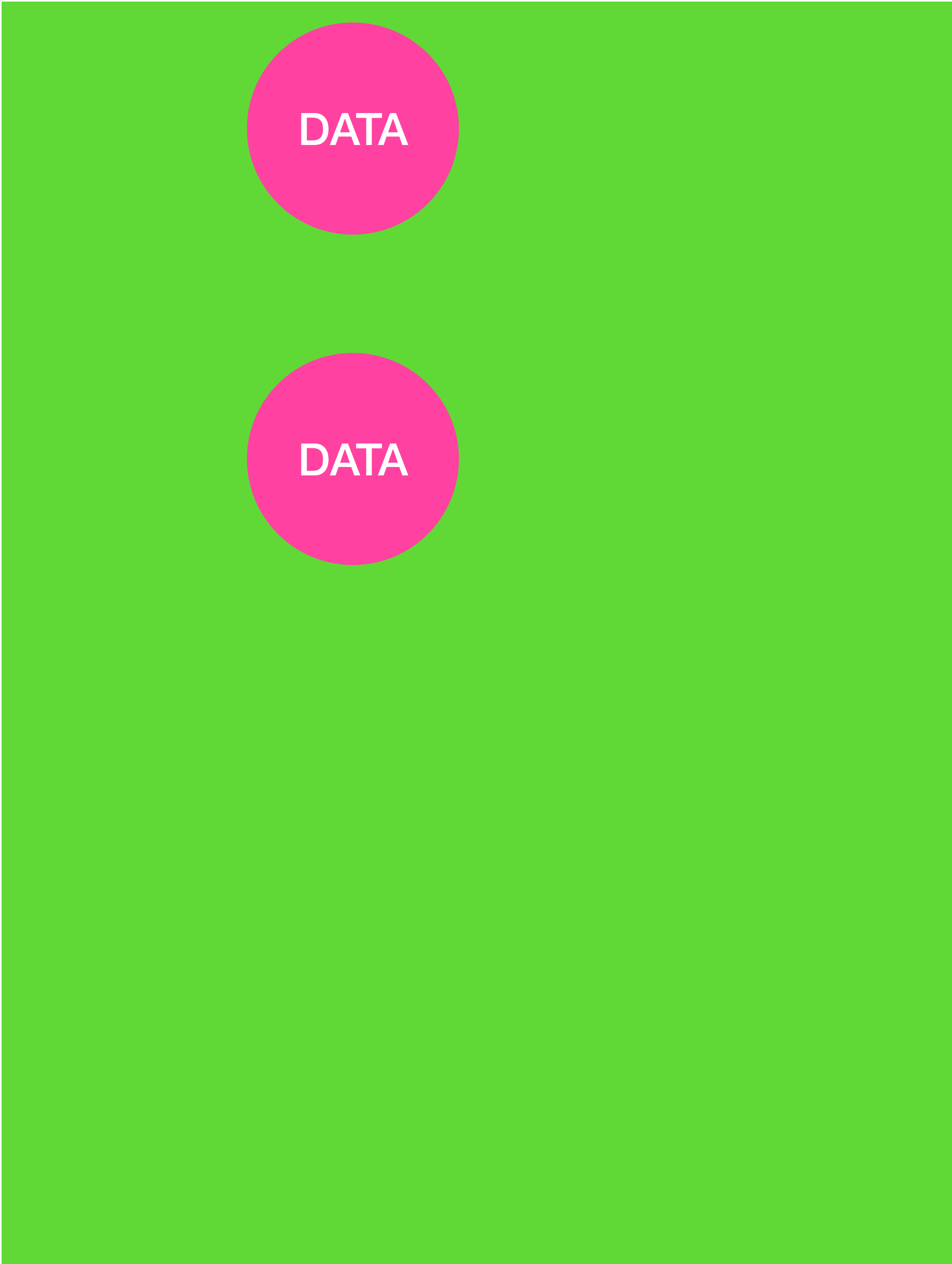
REMINDER ROW COMPONENT

DEL

REMINDER ROW COMPONENT

DEL

State



DEMO For today

INPUT FIELD

SAVE BUTTON

REMINDER ROW COMPONENT

DEL

REMINDER ROW COMPONENT

DEL

REMINDER ROW COMPONENT

DEL

State

Flexbox

State

INPUT FIELD

SAVE BUTTON

REMINDER ROW COMPONENT

DEL

REMINDER ROW COMPONENT

DEL

REMINDER ROW COMPONENT

DEL



<div>

<input>

```
.className{  
background-colour:red  
}
```

<input/>

<button

```
.className{  
.align: center  
}
```

<button/

<div/>

**className =
“mainContainer”**

DIV

DIV

DIV

```
.mainContainer{  
  display: flex;  
}
```

DIV

DIV

DIV

```
.className{  
  
  display: flex;  
  flex-direction: row;  
  
}
```




DIV

DIV

DIV

```
.className{
```

```
  display: flex;  
  flex-direction: row;
```

```
  Justify-content:  
  space-around;
```

```
}
```

DIV

DIV

DIV

```
.className{
```

```
display: flex;  
flex-direction: row;
```

```
Justify-content:  
space-around;
```

```
Align-content:  
Centre  
}
```

DEMO For today

<div> Flex-direction : column

<div> Flex-direction : row

INPUT FIELD

SAVE BUTTON

</div>

<div> Flex-direction : column

REMINDER ROW COMPONENT

DEL

REMINDER ROW COMPONENT

DEL

REMINDER ROW COMPONENT

DEL

</div>

</div>

DEMO For today

<div> Flex-direction : column

<div> Flex-direction : row

<input/>

<button/>

</div>

<div> Flex-direction : column

REMINDER ROW COMPONENT

DEL

REMINDER ROW COMPONENT

DEL

REMINDER ROW COMPONENT

DEL

</div>

</div>

DEMO For today

<div> Flex-direction : column

<div> Flex-direction : row

<input/>

<button/>

</div>

<div> Flex-direction : column

Flex-direction : row

<div> **<div>** **</div>** **<button/>** **DEL** **</div>**

REMINDER ROW COMPONENT

DEL

REMINDER ROW COMPONENT

DEL

</div>

</div>

DEMO For today

<div> Flex-direction : column

<div> Flex-direction : row

<input/>

<button/>

</div>

<div> Flex-direction : column

{
 todoList.map(item =>

DEL

<div> Flex-direction : row

 <div> { item.content } </div>
 <button/>

DEL

</div>

)
}

DEL

</div>

</div>

DEMO For today

<div> Flex-direction : column

<div> Flex-direction : row

<input/>

<button/>

onClick={saveWhatIType}

onChange={displayWhatIType}

</div>

<div> Flex-direction : column

{

todoList.map(item =>

<div> Flex-direction : row

<div> { item.content } </div>

<button/>

onClick={deleteThisTodo}

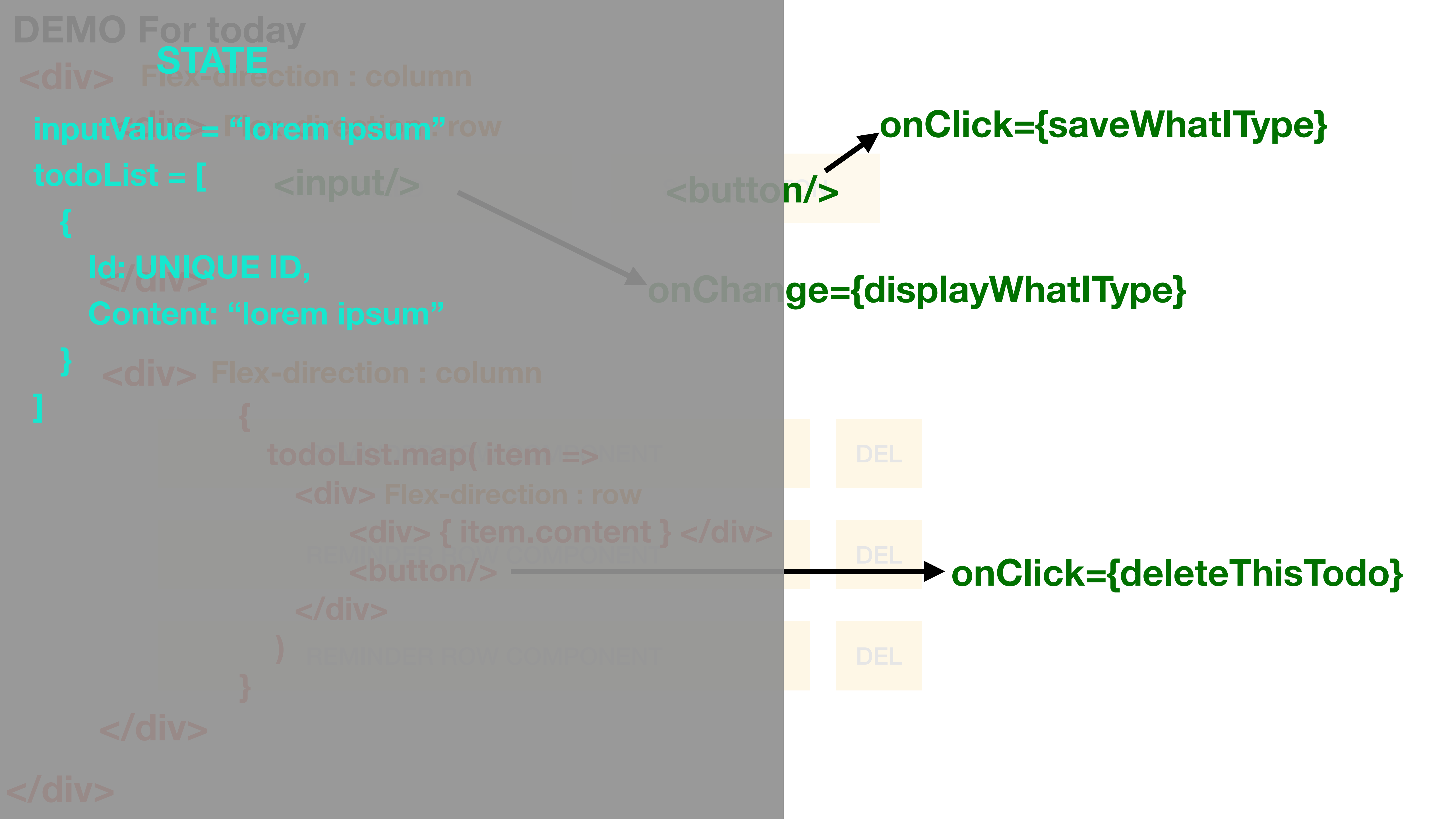
</div>

)

}

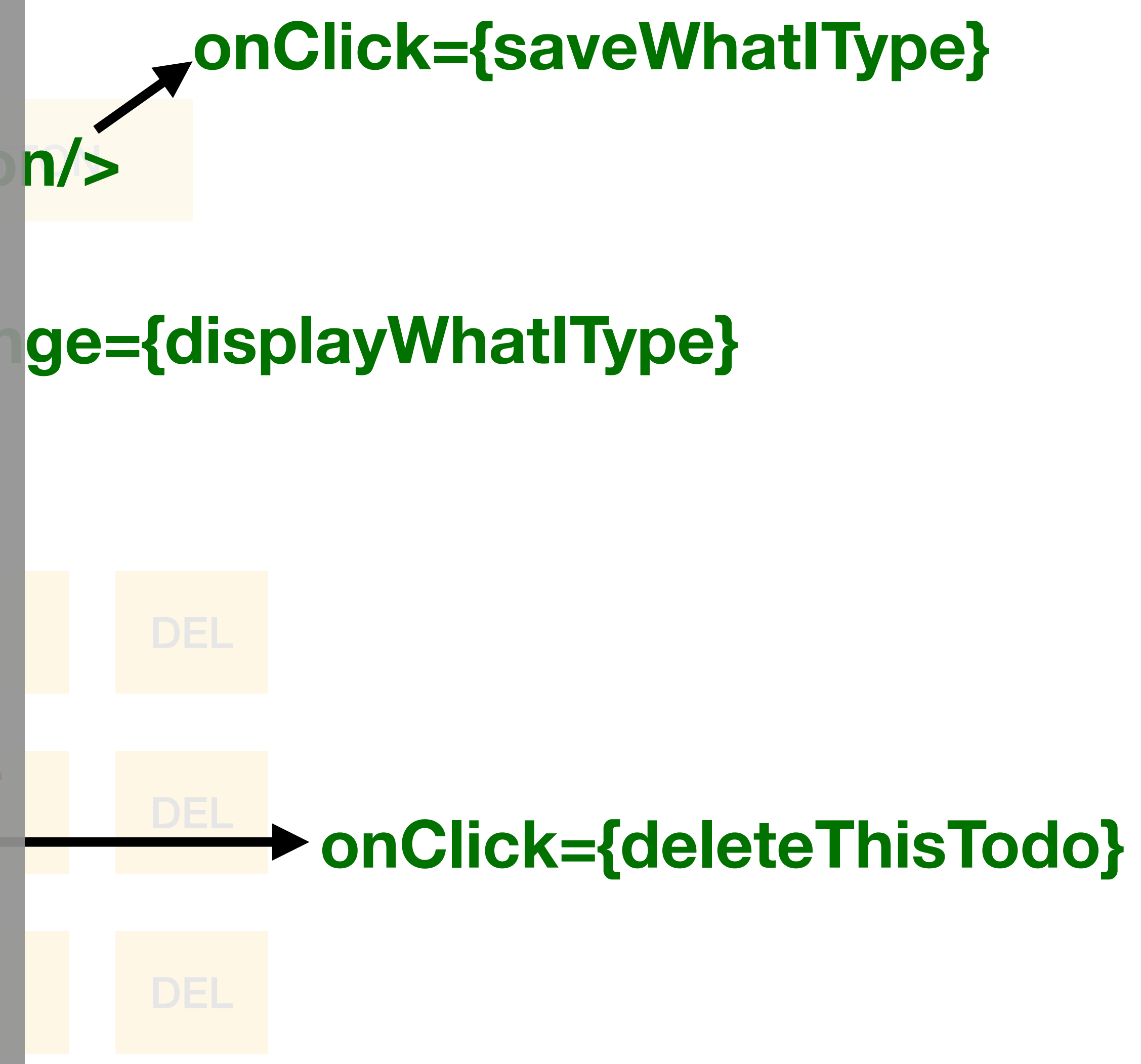
</div>

</div>



DEMO For today

```
STATE
<div>
  Flex-direction : column
  <div> Flex-direction : row
    inputValue = "lorem ipsum"
    todoList = [
      {
        Id: UNIQUE ID,
        Content: "lorem ipsum"
      }
    ]
  <div> Flex-direction : column
    {
      todoList.map( item => NEXT ROW COMPONENT
        <div> Flex-direction : row
          <div> { item.content } </div>
          REMINDER ROW COMPONENT
        <button/>
      </div>
    )
  }
</div>
</div>
```



DEMO For today

STATE

```
<div> Flex-direction : column
  <div> Flex-direction : row
    inputValue = "lorem ipsum"
    todoList = [
      {
        Id: UNIQUE ID,
        Content: "lorem ipsum"
      }
    ]
    [ inputValue, setInputValue ] = useState( "" )
    [ todoList, setTodoList ] = useState( [ ] )
  </div>
  REMINDER ROW COMPONENT
  <div> { item.content } </div>
  <button/>
</div>
</div>
```

FUNCTIONS

```
Const displayWhatIType = ( e ) => {
  setInputValue( e.target.value )
}
Const saveWhatIType = () => {
  Let newContent = {
    Id: math.random.toString().replace("0.", ""),
    Content: inputValue.trim()
  }
  Let copy = [...todoList]
  copy.push( newContent )
  setTodoList( copy )
  setInputValue( "" )
}
Const deleteThisTodo = (id) => {
  setTodoList(
    todoList.filter( item=>item.id != id ))
}
```

DEMO For today

```
<div> Flex-direction : column
  <div> Flex-direction : row
    <input value={inputValue} /> <button/>
  </div>
  <div> Flex-direction : column
    {
      todoList.map(item =>
        <div> Flex-direction : row
          <div> { item.content } </div>
          <button/>
        </div>
      )
    }
  </div>
</div>
```

onClick={saveWhatIType}

onChange={displayWhatIType}

STATE

[inputValue, setInputValue] = useState("")

[todoList, setTodoList] = useState([])

onClick={deleteThisTodo}

FUNCTIONS

```
Const displayWhatIType = ( e ) => {  
  setInputValue( e.target.value )  
}
```

```
Const saveWhatIType = () => {  
  Let newContent = {  
    Id: math.random.toString().replace("0.", ""),  
    Content: inputValue.trim()  
  }  
  Let copy = [...todoList]  
  copy.push( newContent )  
  setTodoList( copy )  
  setInputValue( "" )  
}
```

```
Const deleteThisTodo = (id) => {  
  setTodoList(  
    todoList.filter( item=>item.id != id )  
  )  
}
```

<button/>

onClick={saveWhatIType}

onChange={displayWhatIType}

</div>

onClick={deleteThisTodo}

FUNCTIONS

```
Const deleteThisTodo = (id) => {  
  setTodoList(  
    todoList.filter( item=>item.id != id ))  
}
```

<button/>

onClick={saveWhatIType}

</div>

onChange={displayWhatIType}

<div> Flex-direction : column

{

todoList.map(item =>

<div> Flex-direction : row

<div> { item.content } </div>

<button/>

</div>

)

}

</div>

</div>

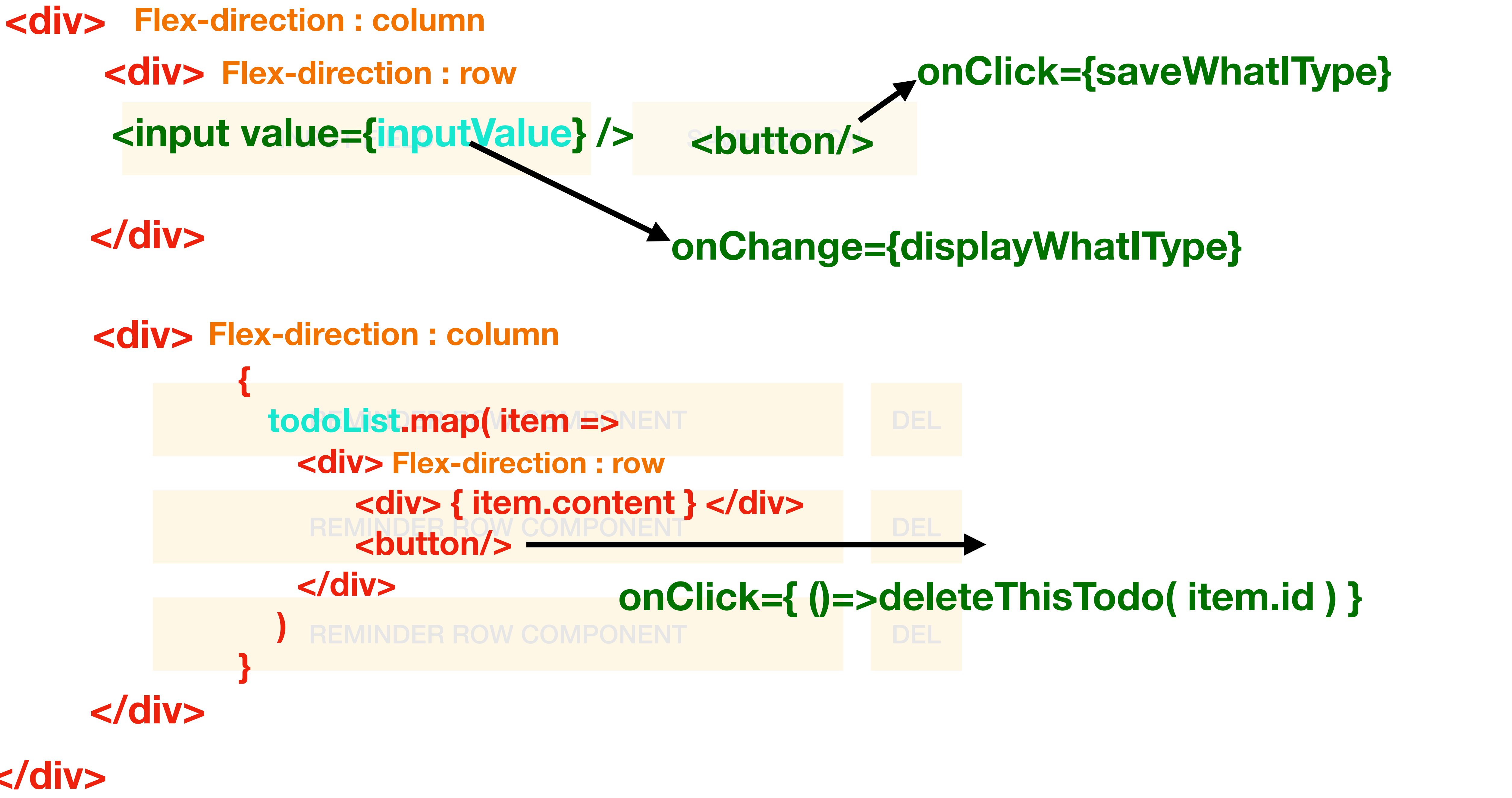
onClick={ ()=>deleteThisTodo(item.id) }

DEL

DEL

DEL

DEMO For today



DEMO For today

```
{
  todoList.map(item =>
    <TodoRow
      item={item}
      deleteThisTodo={deleteThisTodo}
    />
  )
}

const TodoRow = (props) => {
  let { item, deleteThisTodo } = props
  return(
    <div>
      <div> { item.content } </div>
      <button/> onClick={ ()=>deleteThisTodo( item.id ) }
    </div>
  )
}
```

deleteThisTodo: function,

Item: {
id:1234,
Content: "lorem ipsum"

```
const TodoRow = (props) => {
  let { item, deleteThisTodo } = props
  return(
    <div>
      <div> { item.content } </div>
      <button/> onClick={ ()=>deleteThisTodo( item.id ) }
    </div>
  )
}
```


Things to explore

REACT

React lifecycles

React Browser Router

Class based components

Redux / React context

Component libraries

- antd

MOBILE

React Native with expo

BACKEND

NodeJS Express

- API

- Middleware

- Managing sessions

- Account system

▪

▪

▪