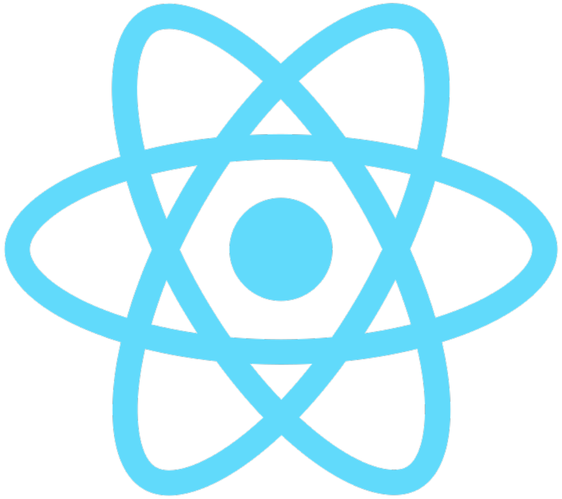
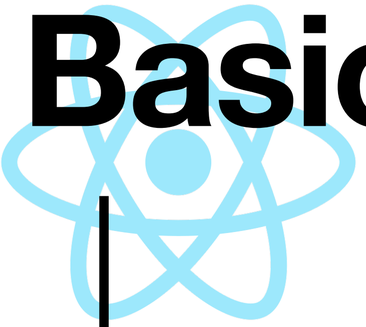


# Intro to Web Dev and React

3DC

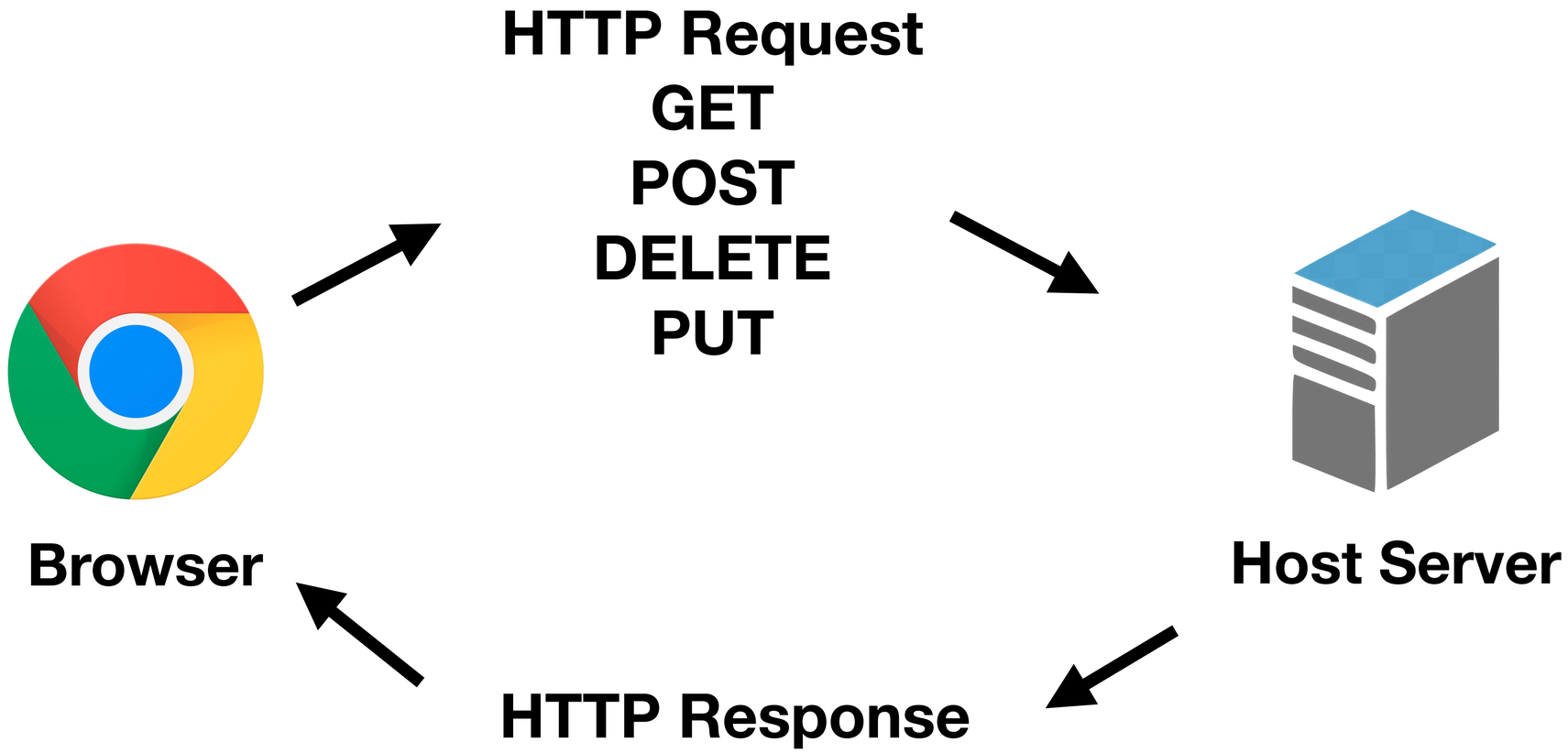
Bryce Goh

# Basic React Introduction



- Basic overview of the web
- Why react
- Javascript Refreshers
- Components and JSX
- To-do App

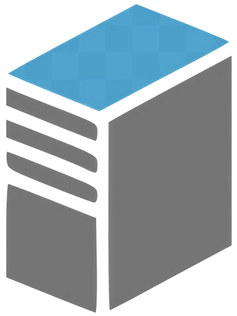
# Basic overview of the web



# Basic overview of the web

www.sutd.edu.sg

HTTP Request
GET



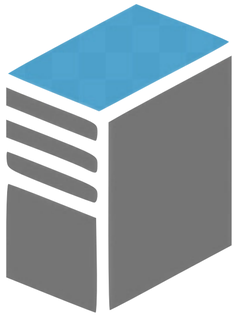
DNS Server

Browser

Host Server

# Basic overview of the web

www.sutd.edu.sg

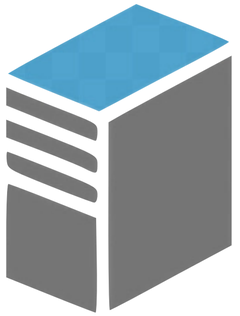


DNS Server

Browser

Host Server

# Basic overview of the web



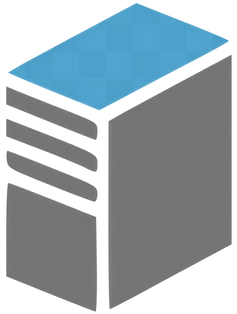
IP address

DNS Server

Browser

Host Server

# Basic overview of the web



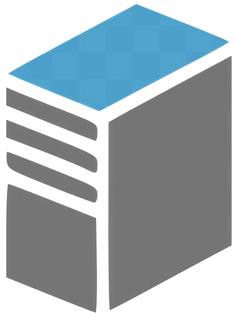
DNS Server

Browser

IP address

Host Server

# Basic overview of the web



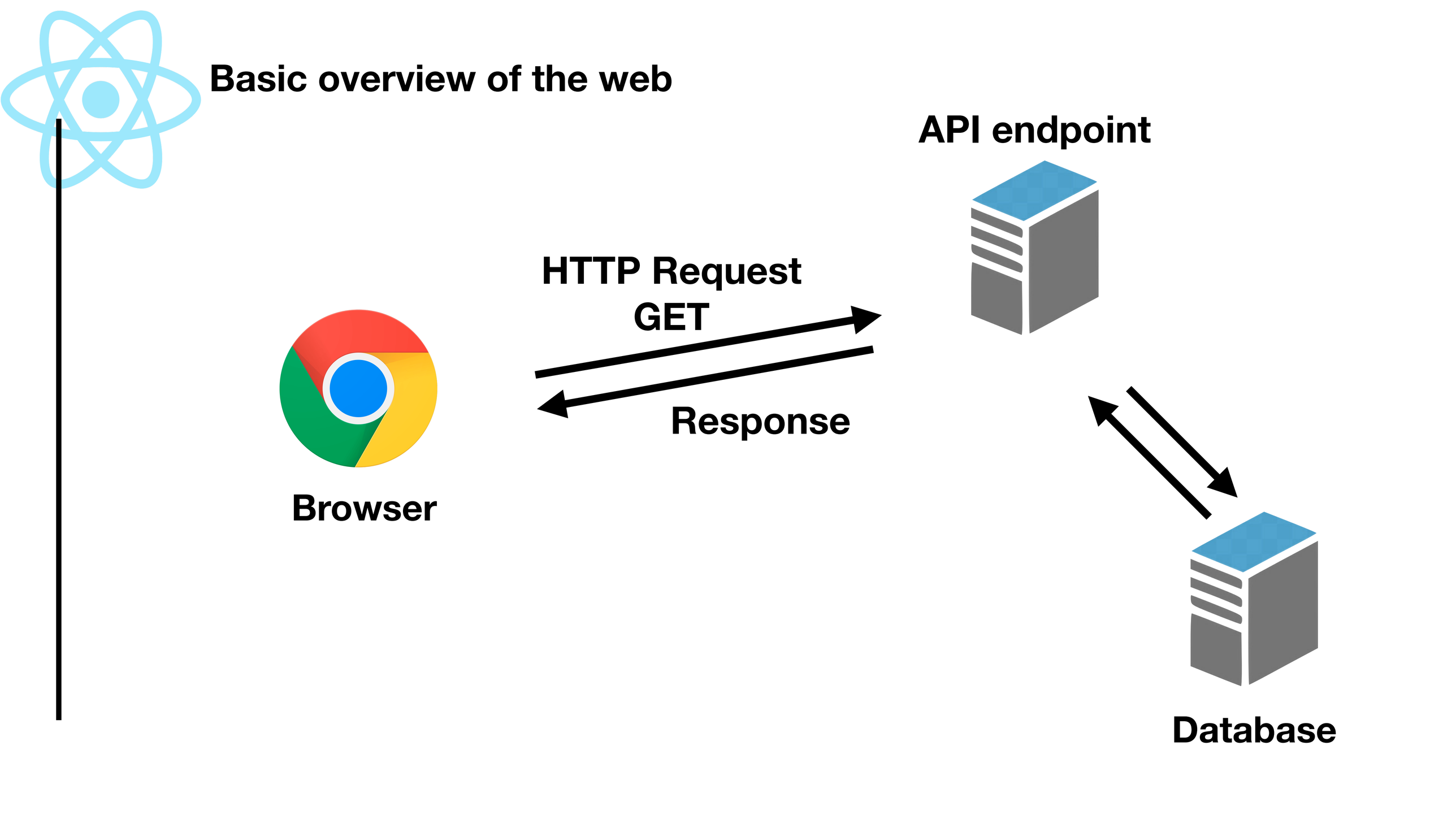
DNS Server

Browser

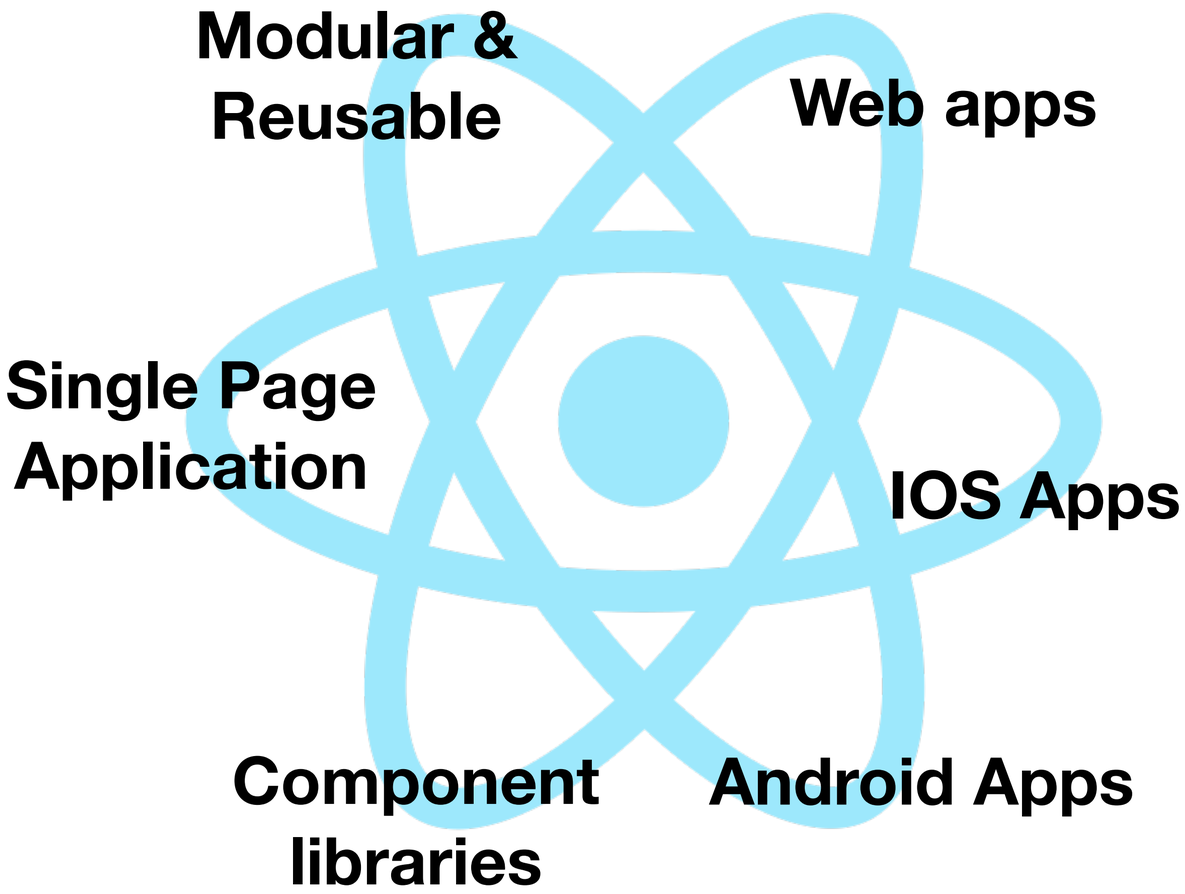
IP address

Host Server

# Basic overview of the web



# Why react


Modular &
Reusable
Web apps
Single Page
Application
IOS Apps
Component
libraries
Android Apps

# Javascript Refreshers

| | | |
|---|---|---|
| Defining Variables | variable_name = “hello world” | Const variableName = “hello world”<br>Let variableName = “hello world” |
| List/Array & Objects/ Dictionary | List = [….]<br><br>Dictionary = {<br>Key: value<br>} | Const list = [….]<br><br>Const dictionary = {<br>Key: value<br>} |

# Javascript Refreshers

**Functions**

```
Def function_name(argument):
  …
  …
  return(…)
```

```
Const functionName = (argument) => {
  …
  …
  return(…)
}
```

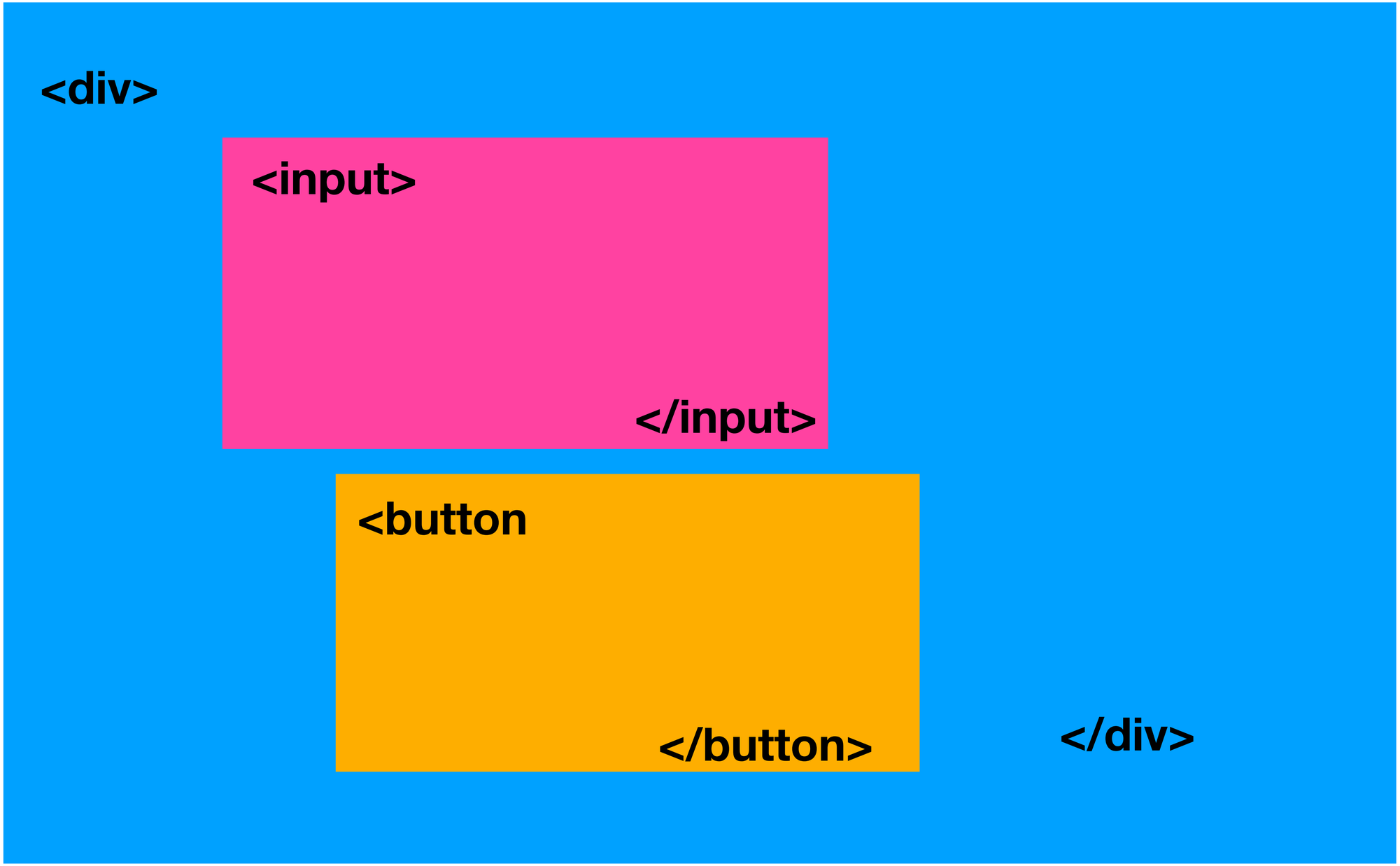
**Loops**

```
For x in list:
  print(x)
```

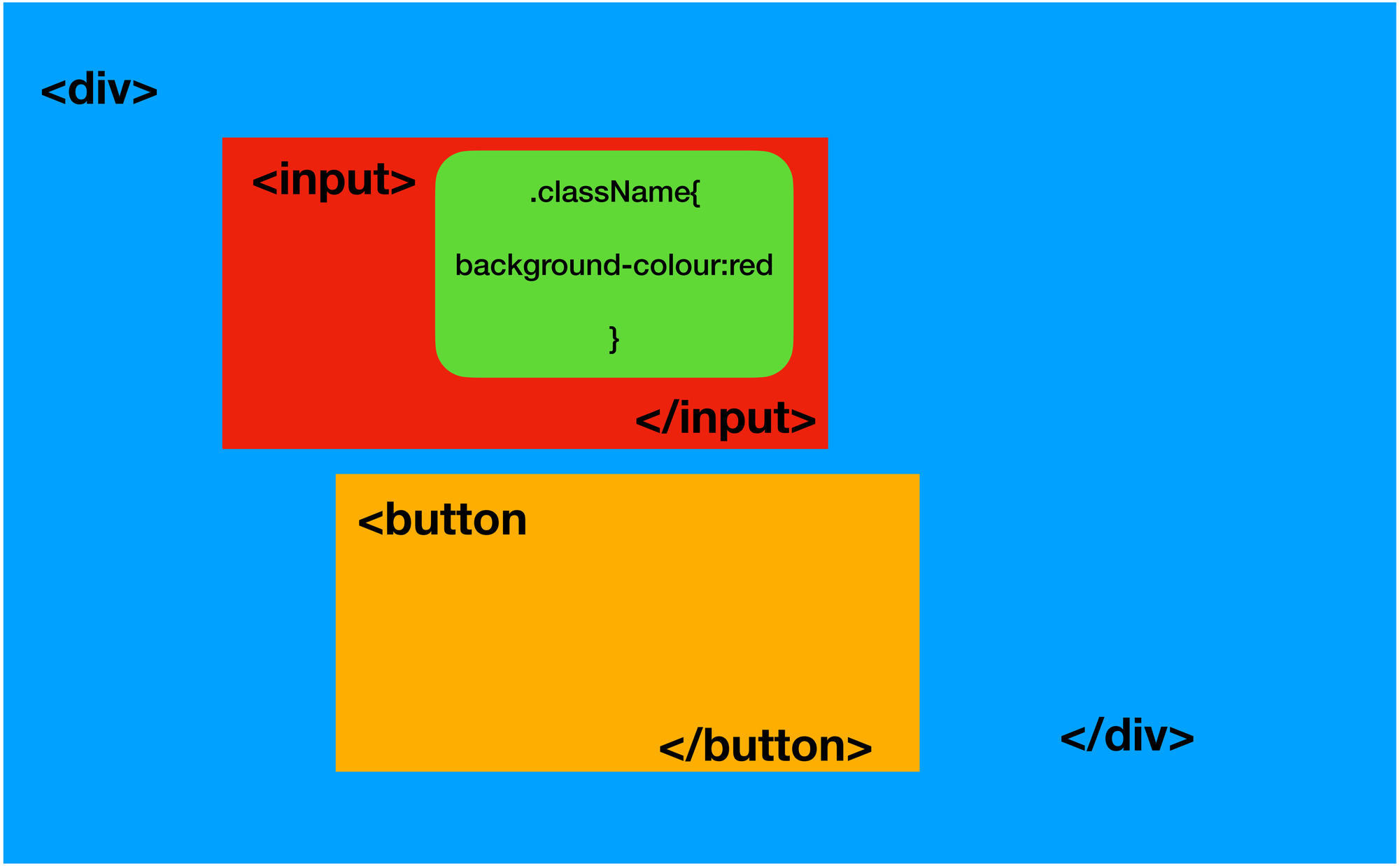
```
list.map( ( x , index ) => {
  console.log(x)
  }
)
```

# HTML/CSS Refreshers

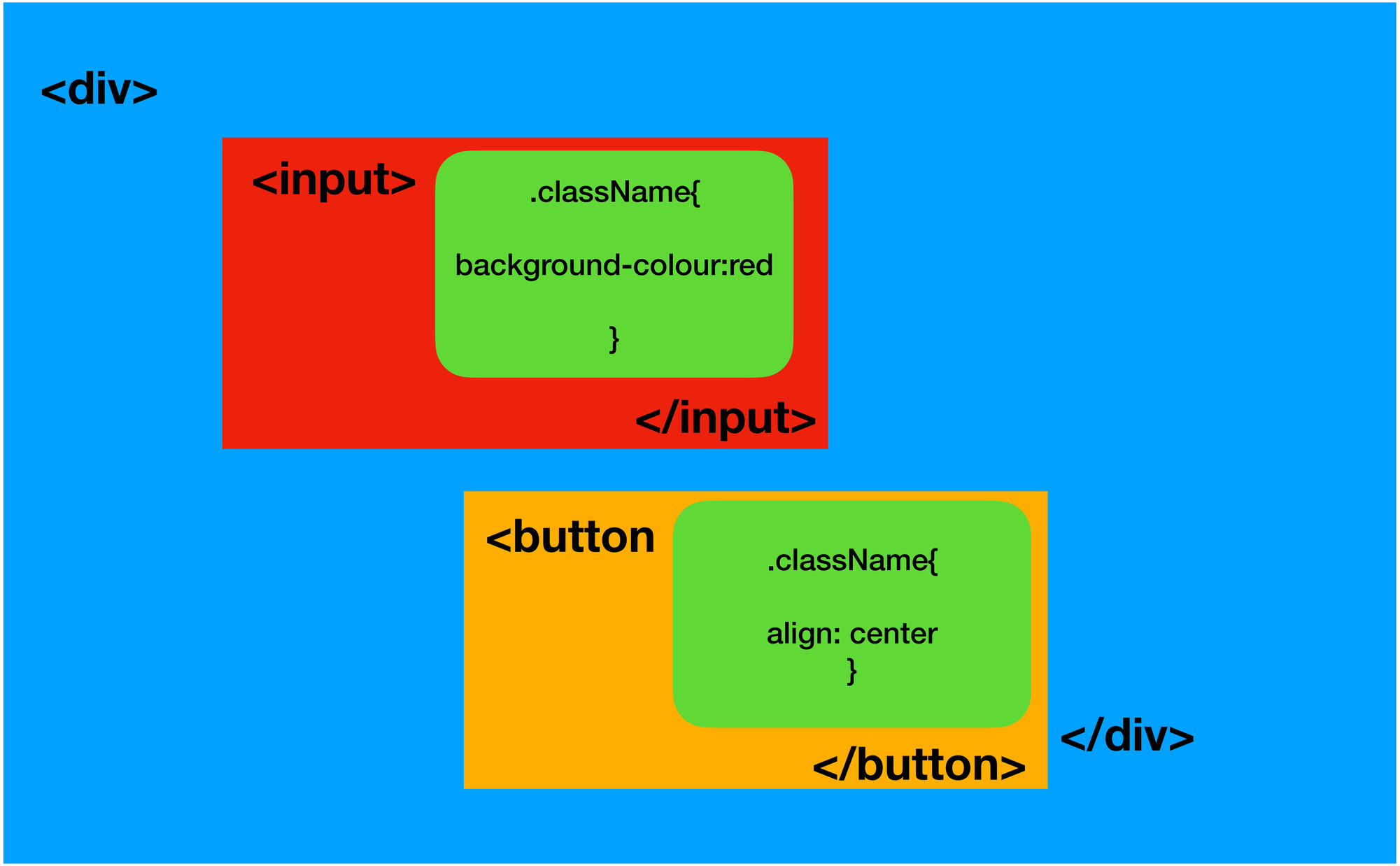
# HTML/CSS Refreshers



# HTML/CSS Refreshers

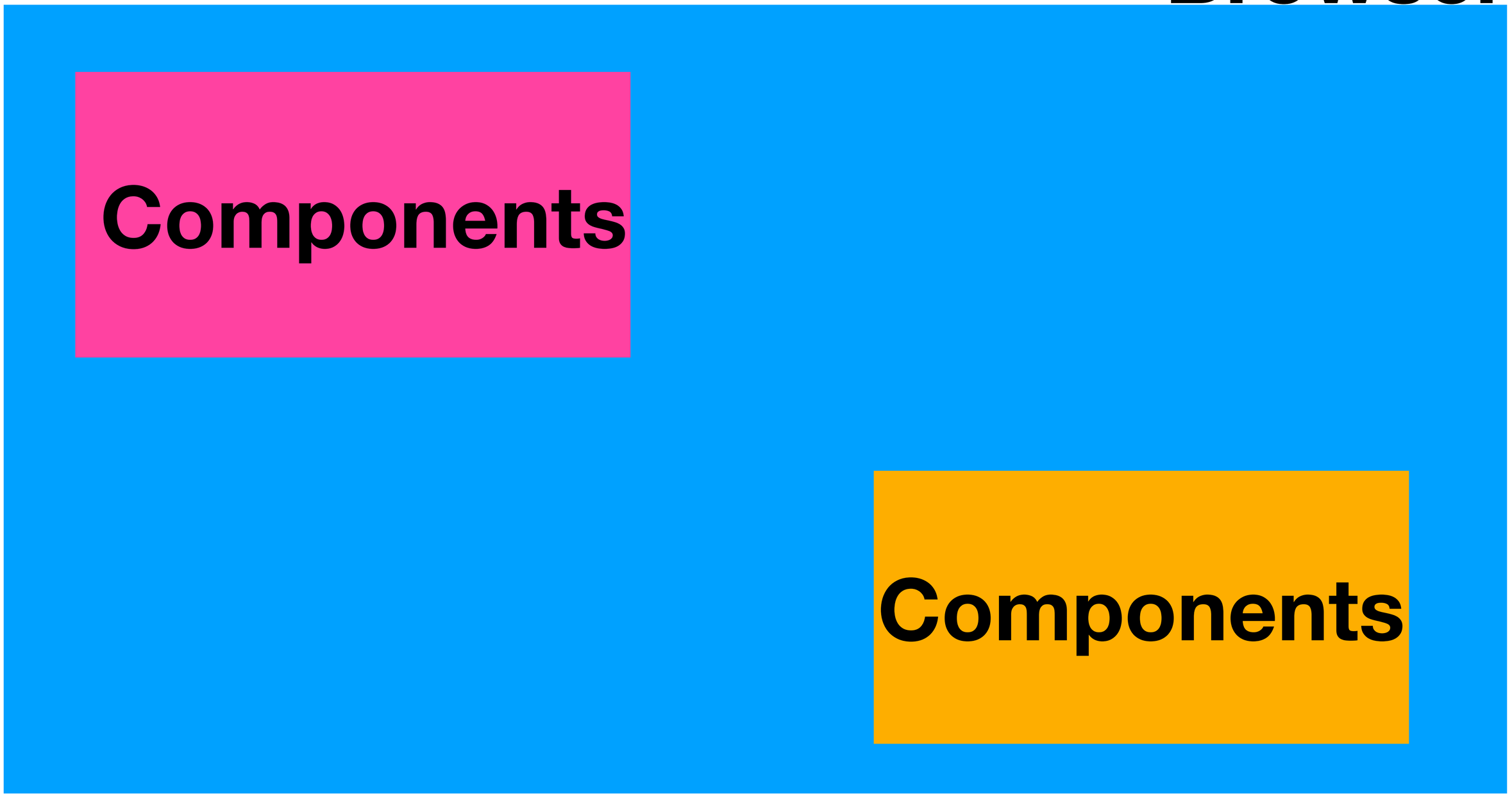


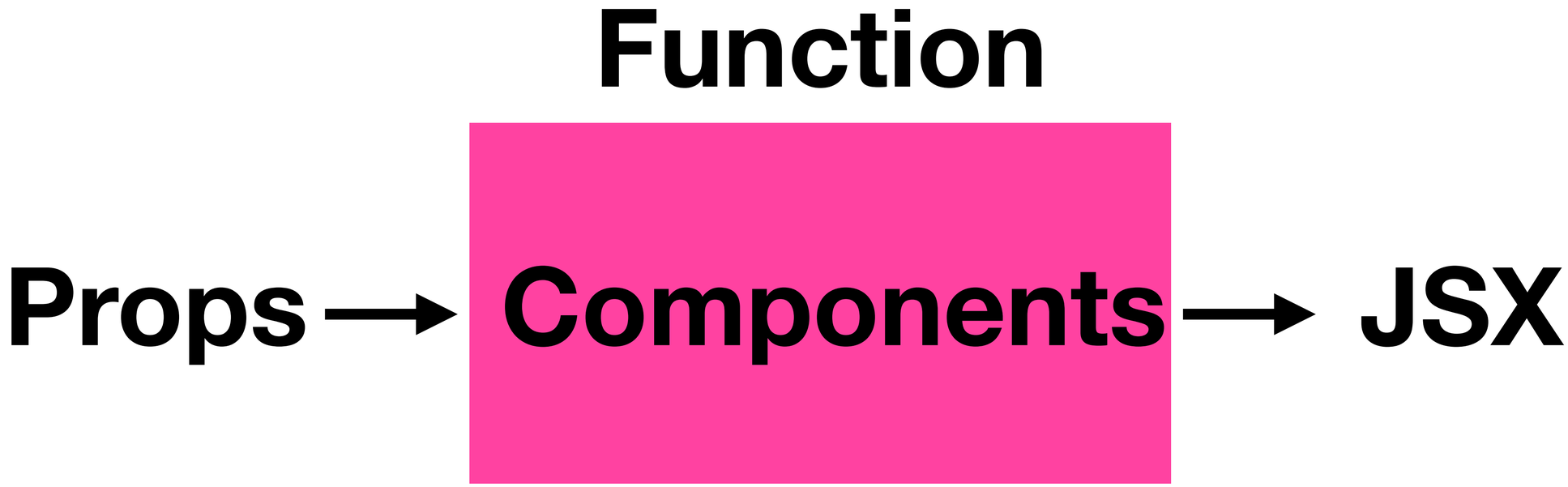
# HTML/CSS Refreshers

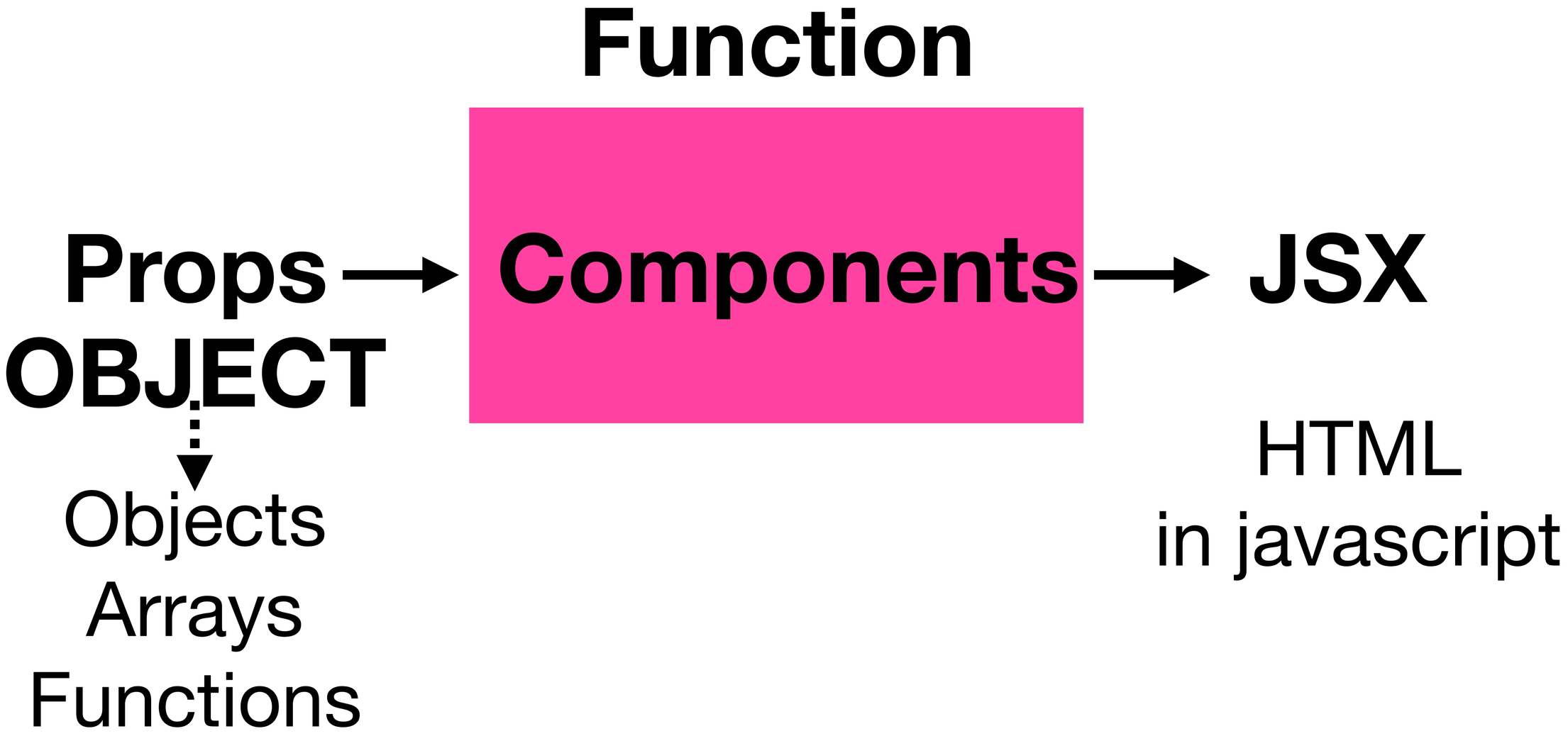


REACT

Browser




Function
Props
Components
JSX


Function
Props
OBJECT
Components
JSX
Objects
Arrays
Functions
HTML
in javascript

REACT

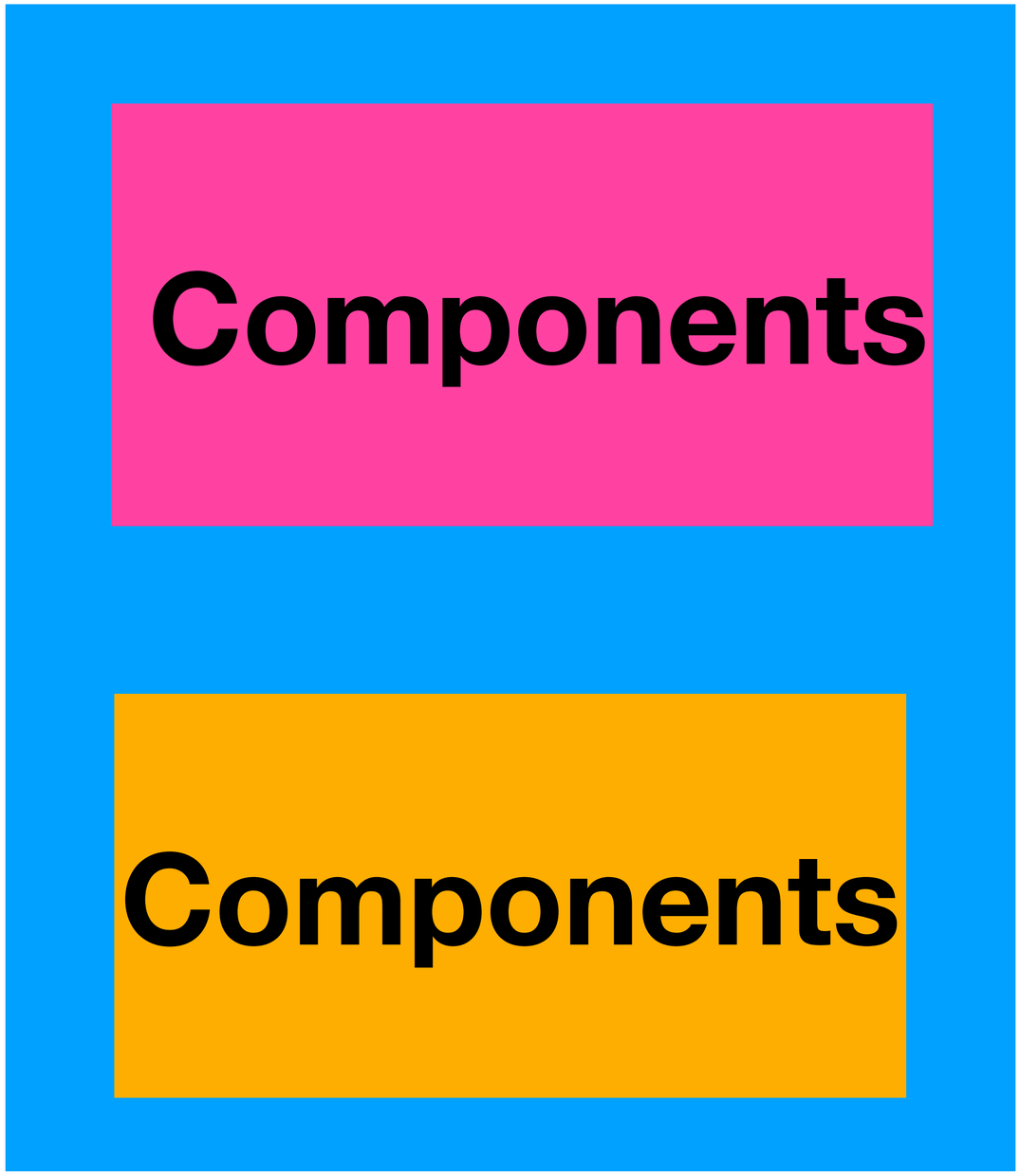
# Props → Components → JSX

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

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

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

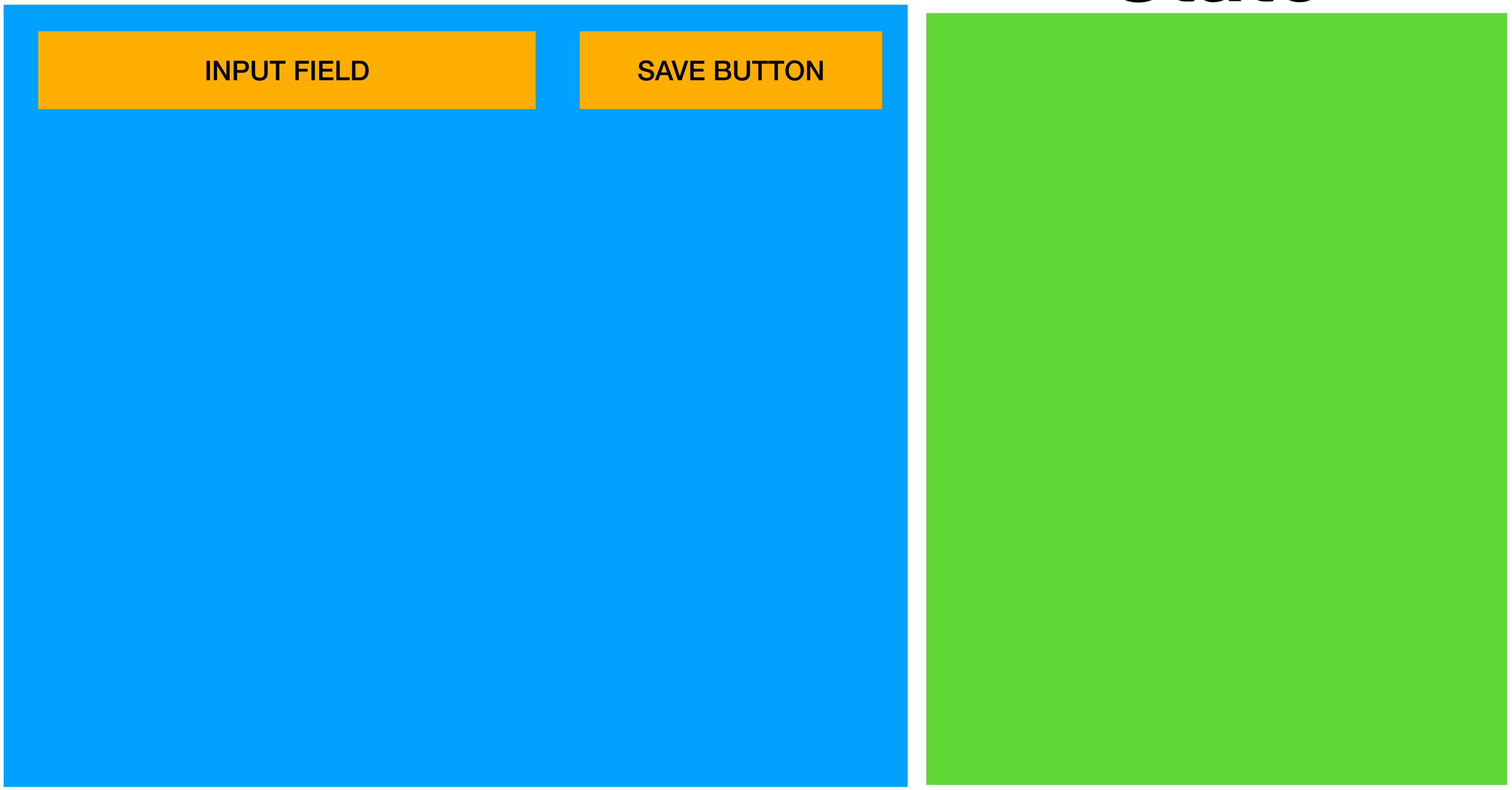
REACT



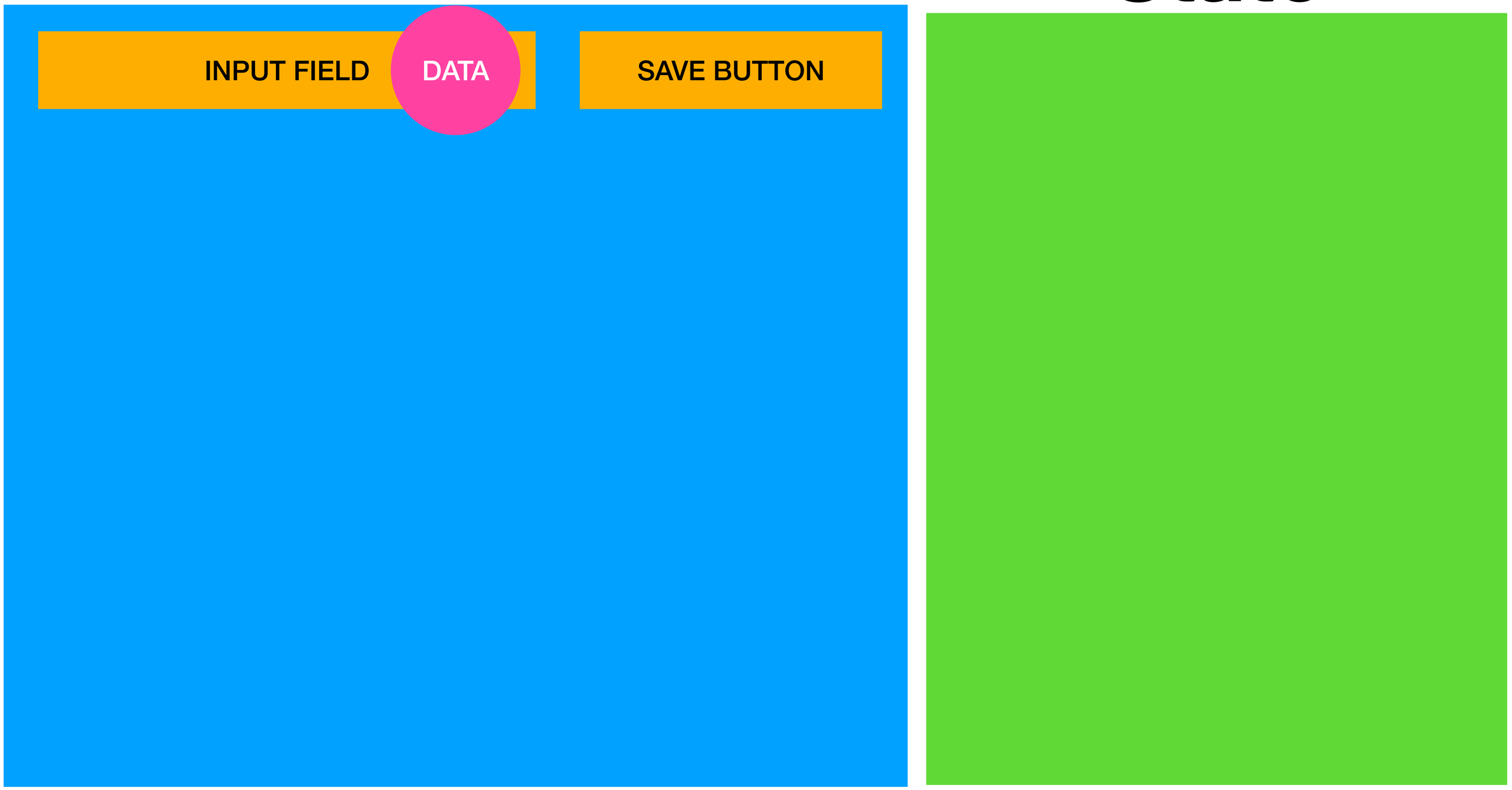

# State

TEMP
STORAGE

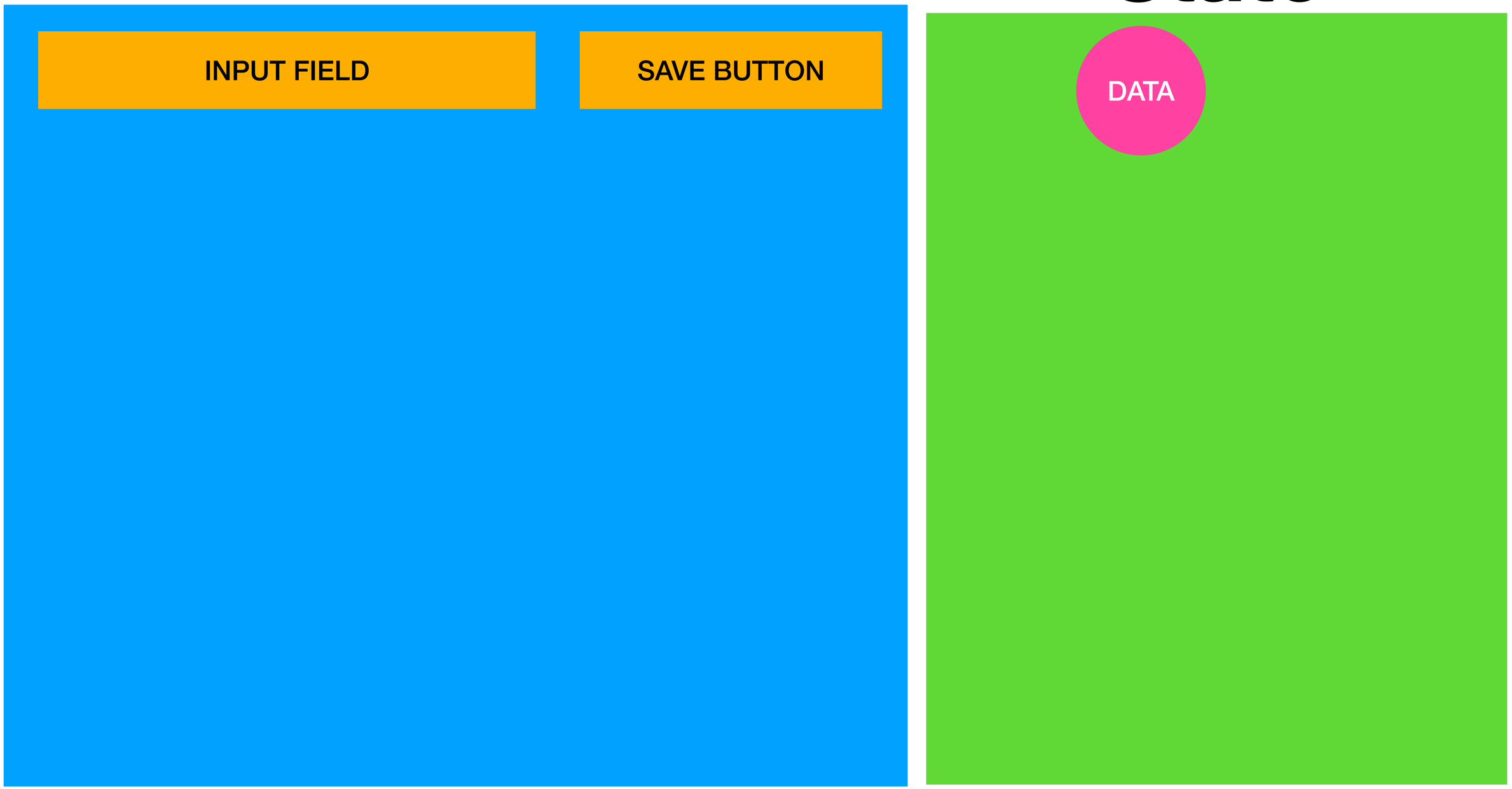
# DEMO For today



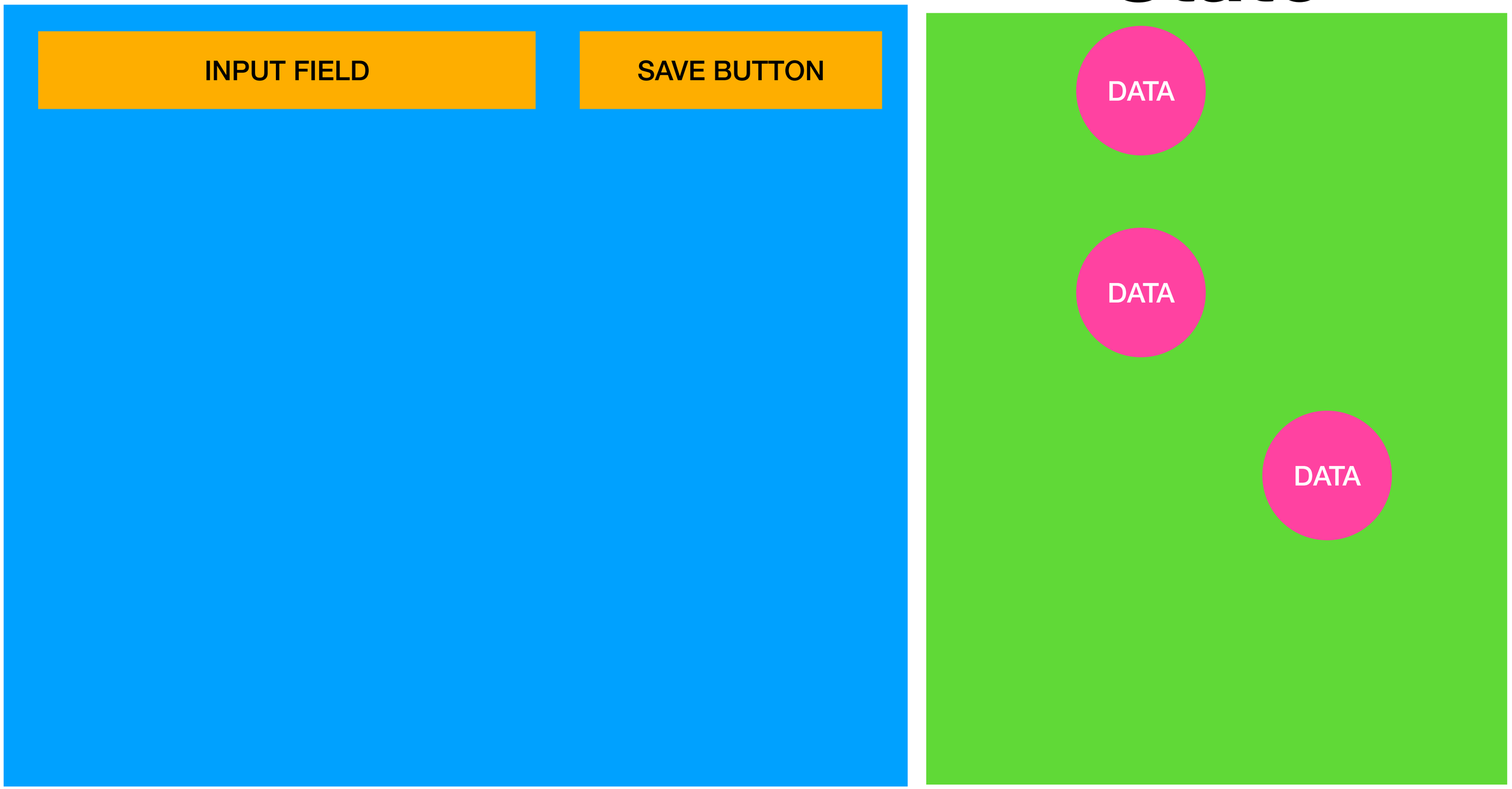
# DEMO For today



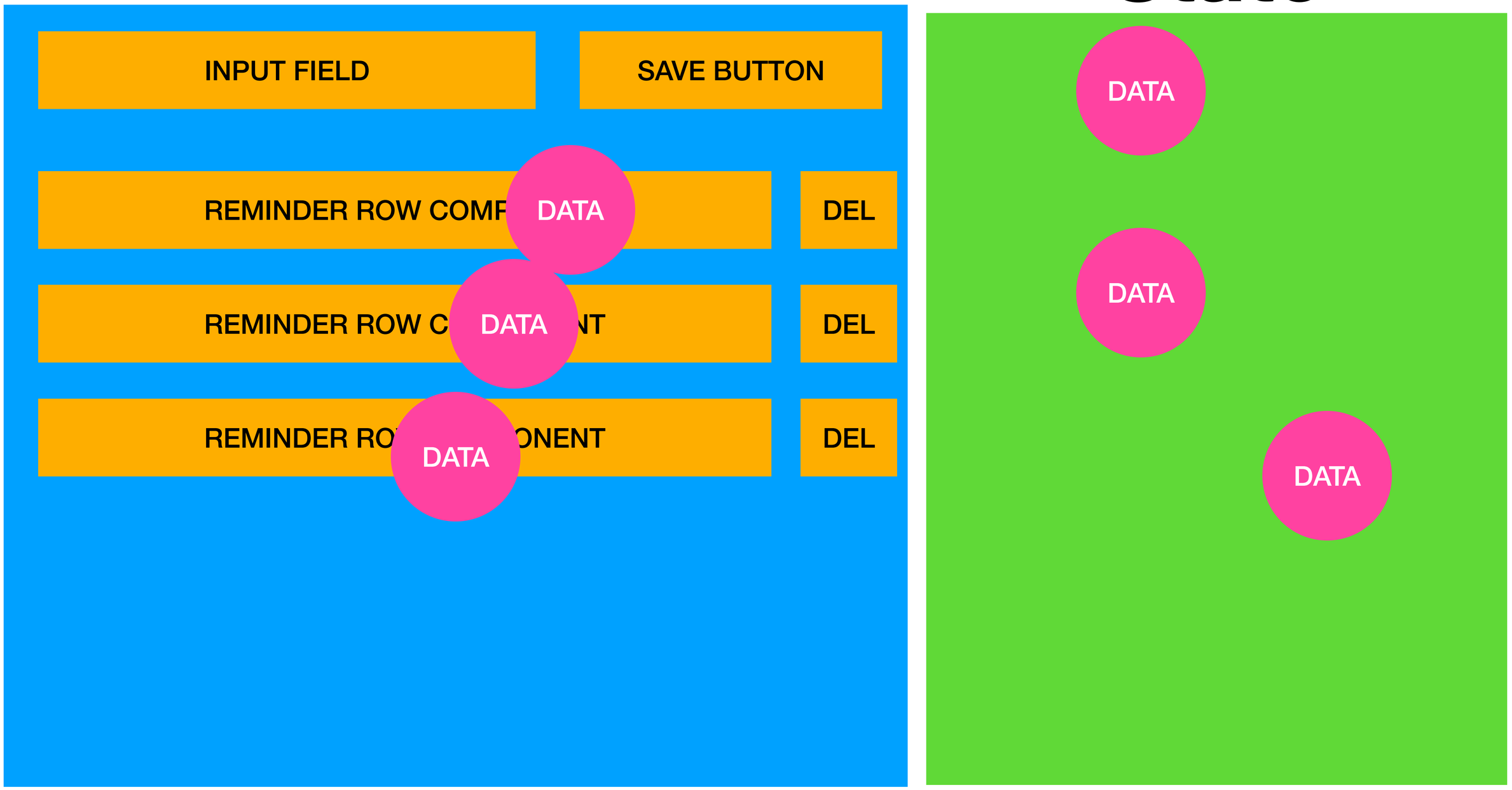
# DEMO For today



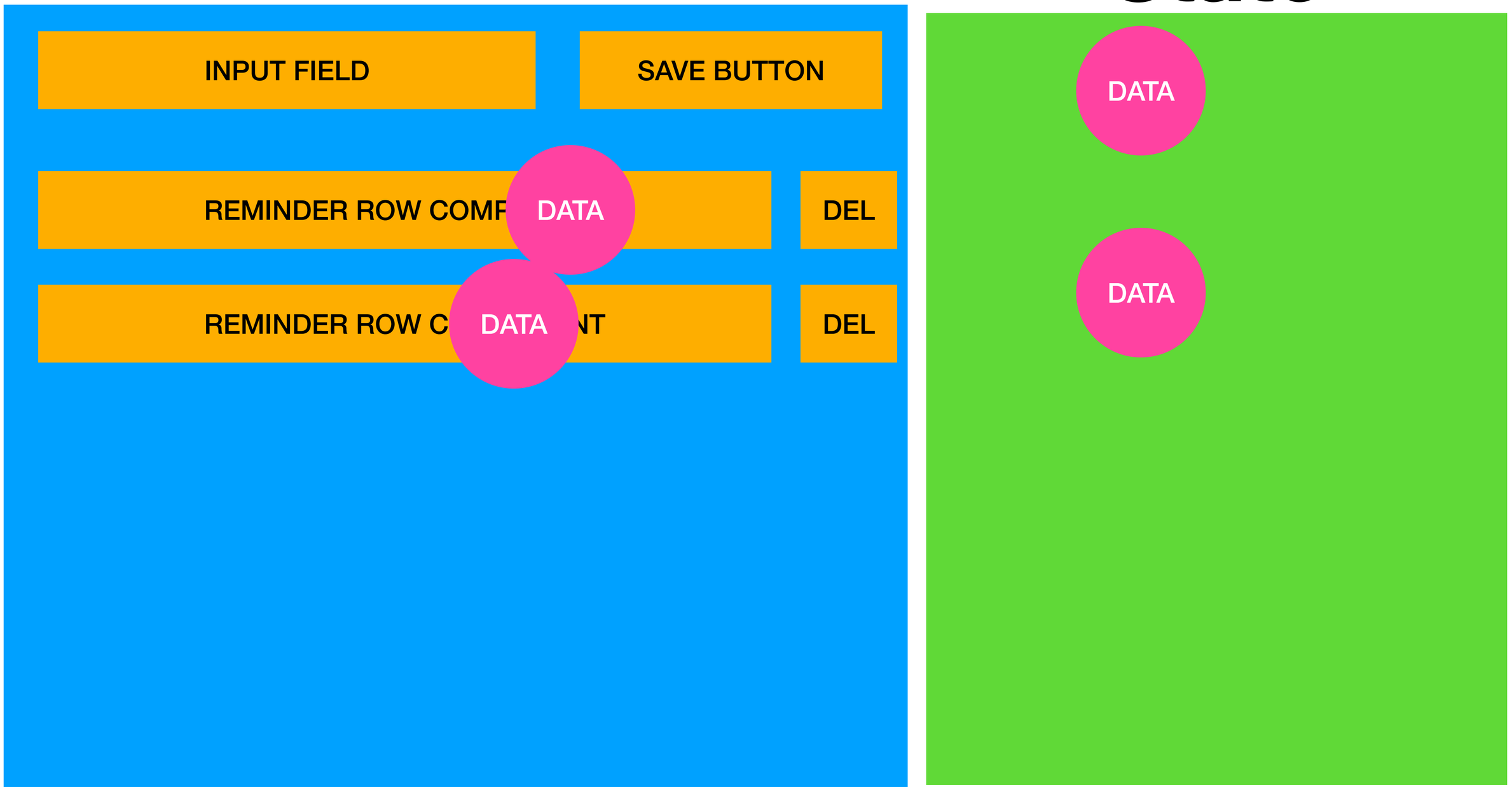
# DEMO For today



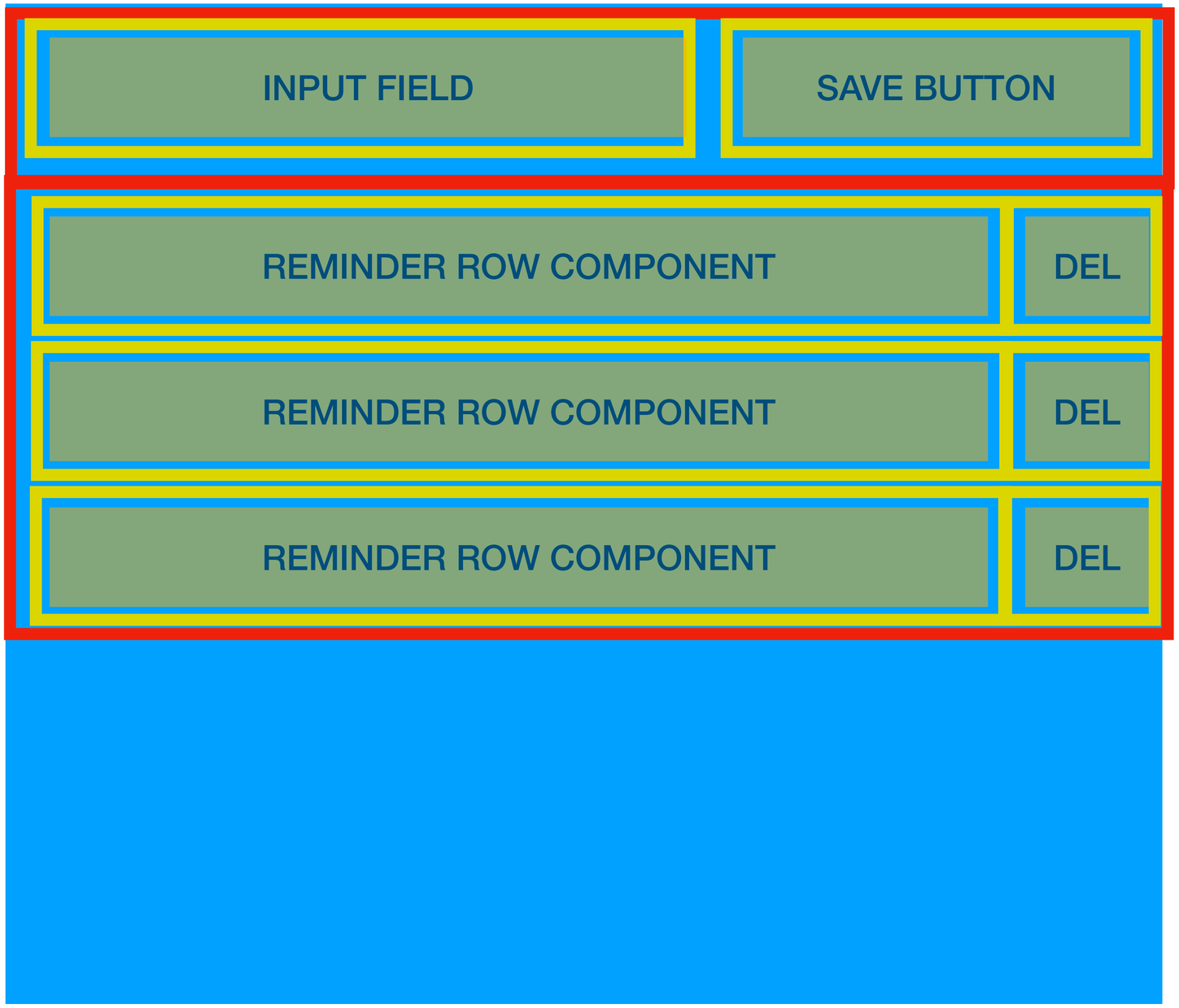
# DEMO For today



# DEMO For today



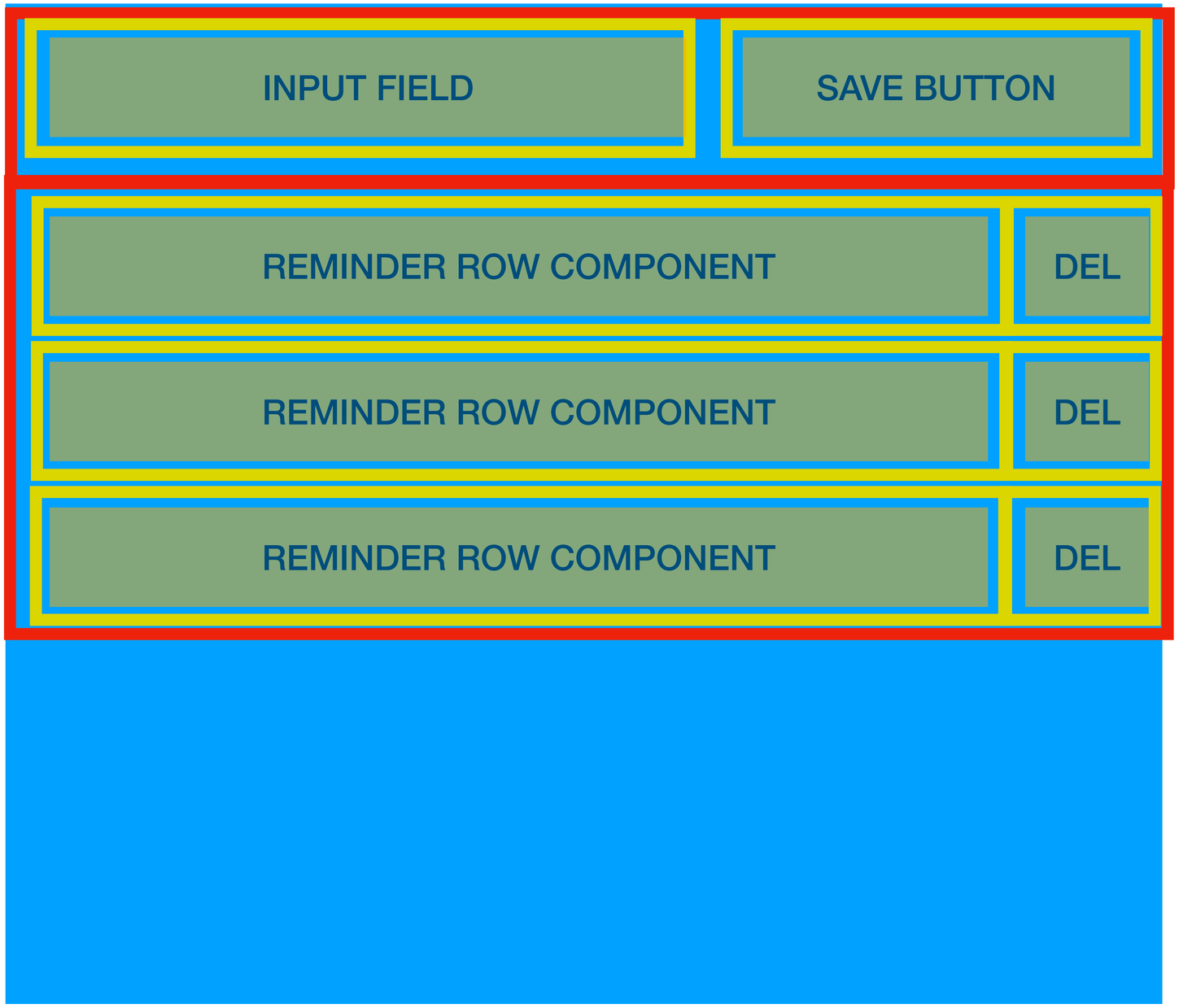
# DEMO For today



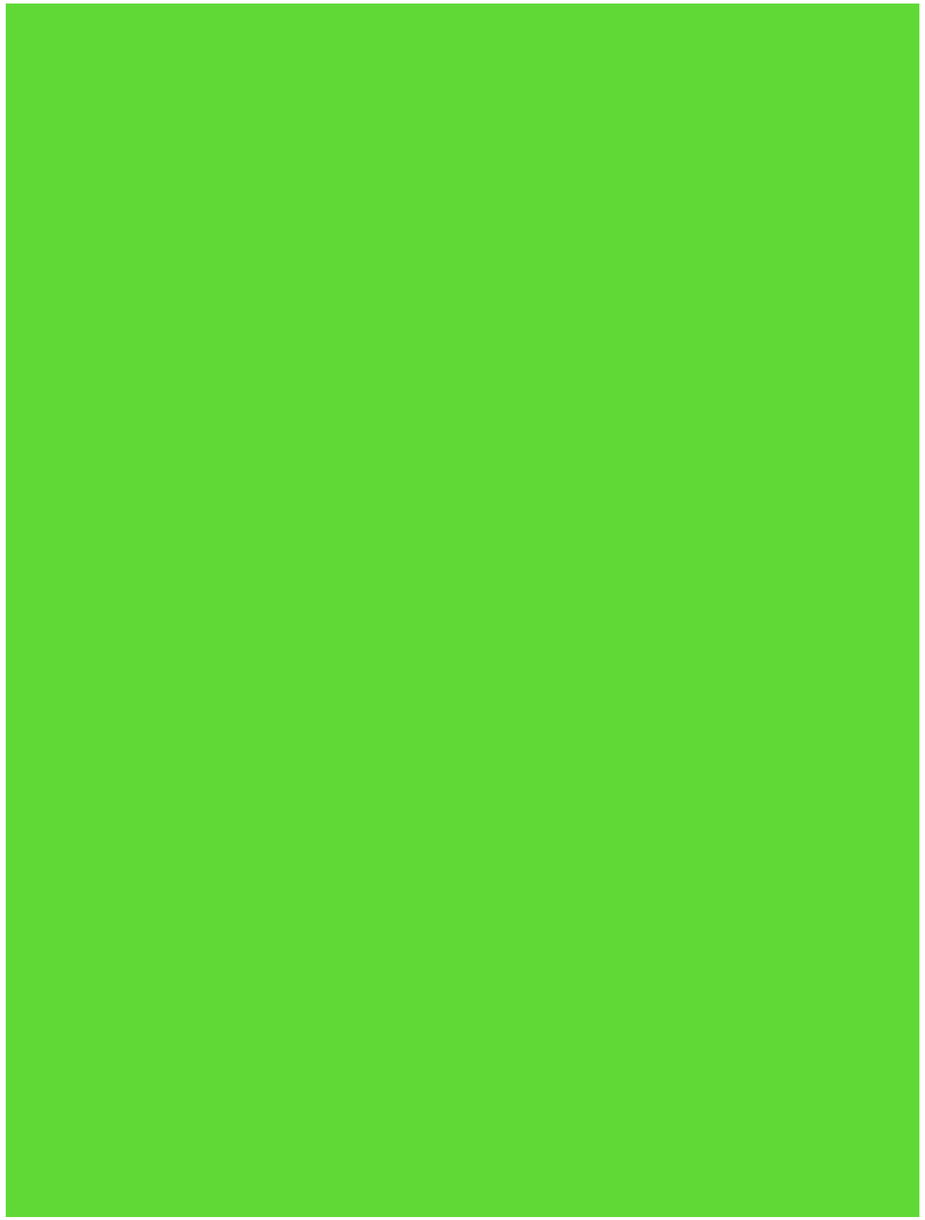

# State

DEMO For today

# Flexbox

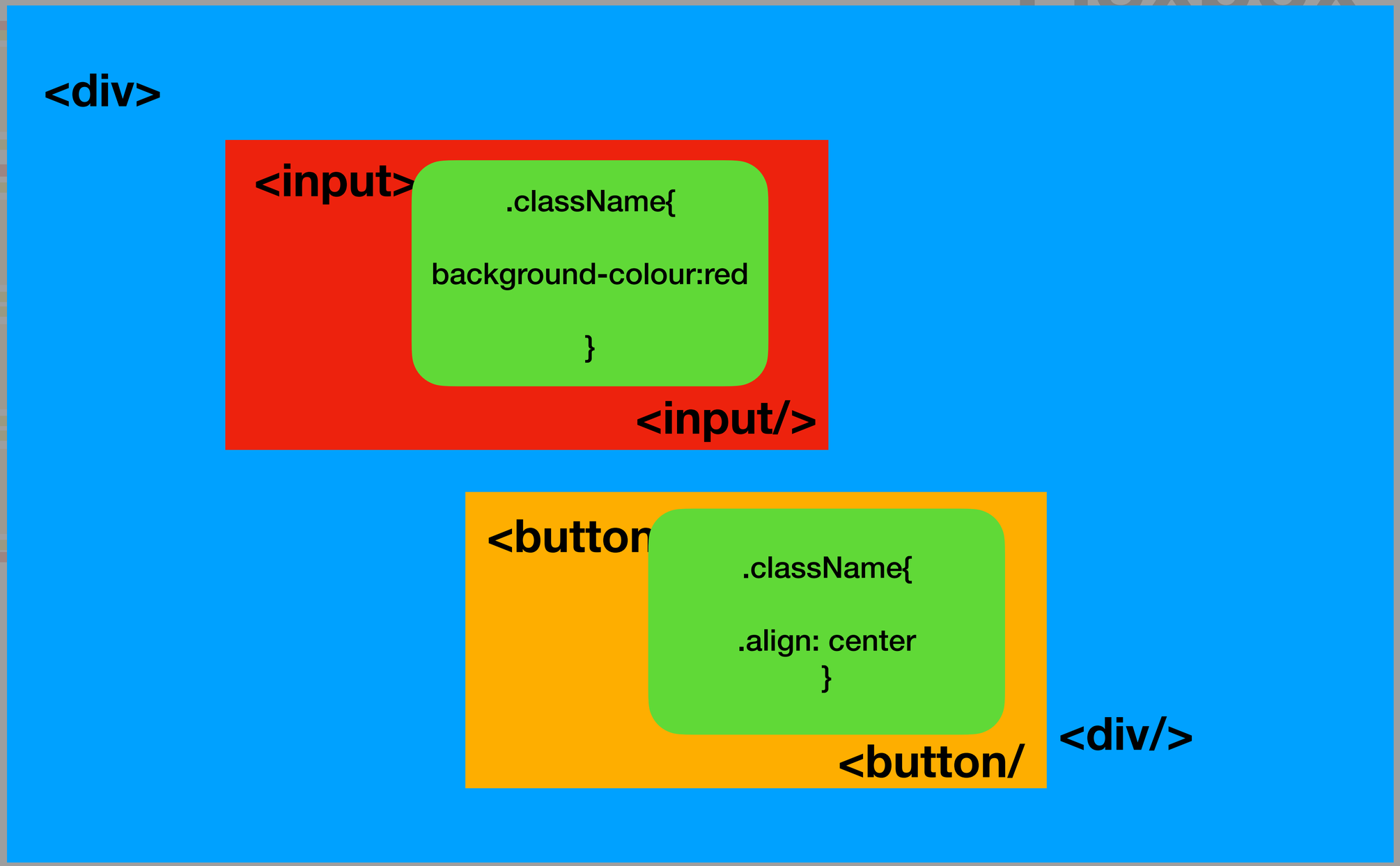



# State



DEMO For today

Flexbox



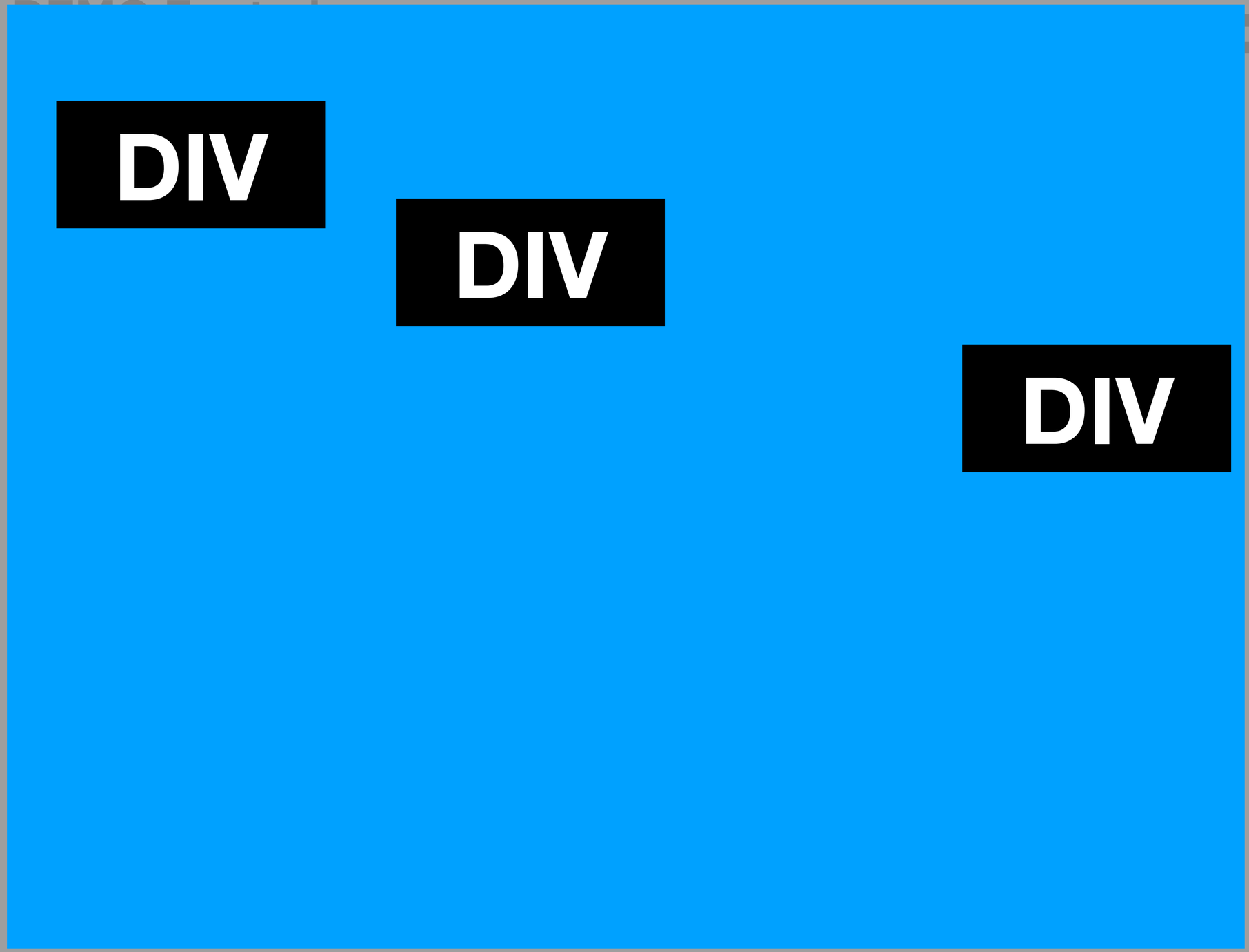
# className = “mainContainer”

DIV

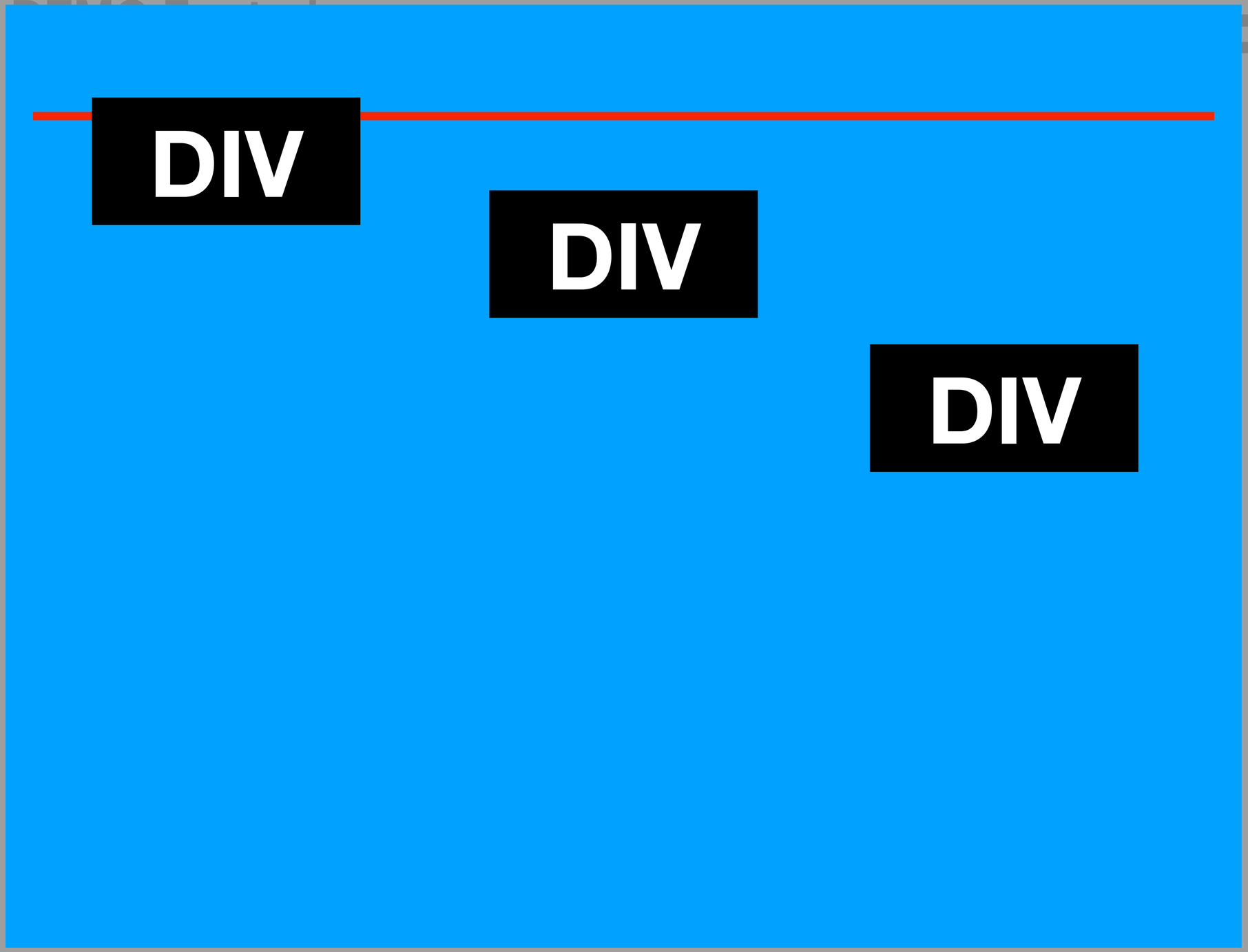
DIV

DIV

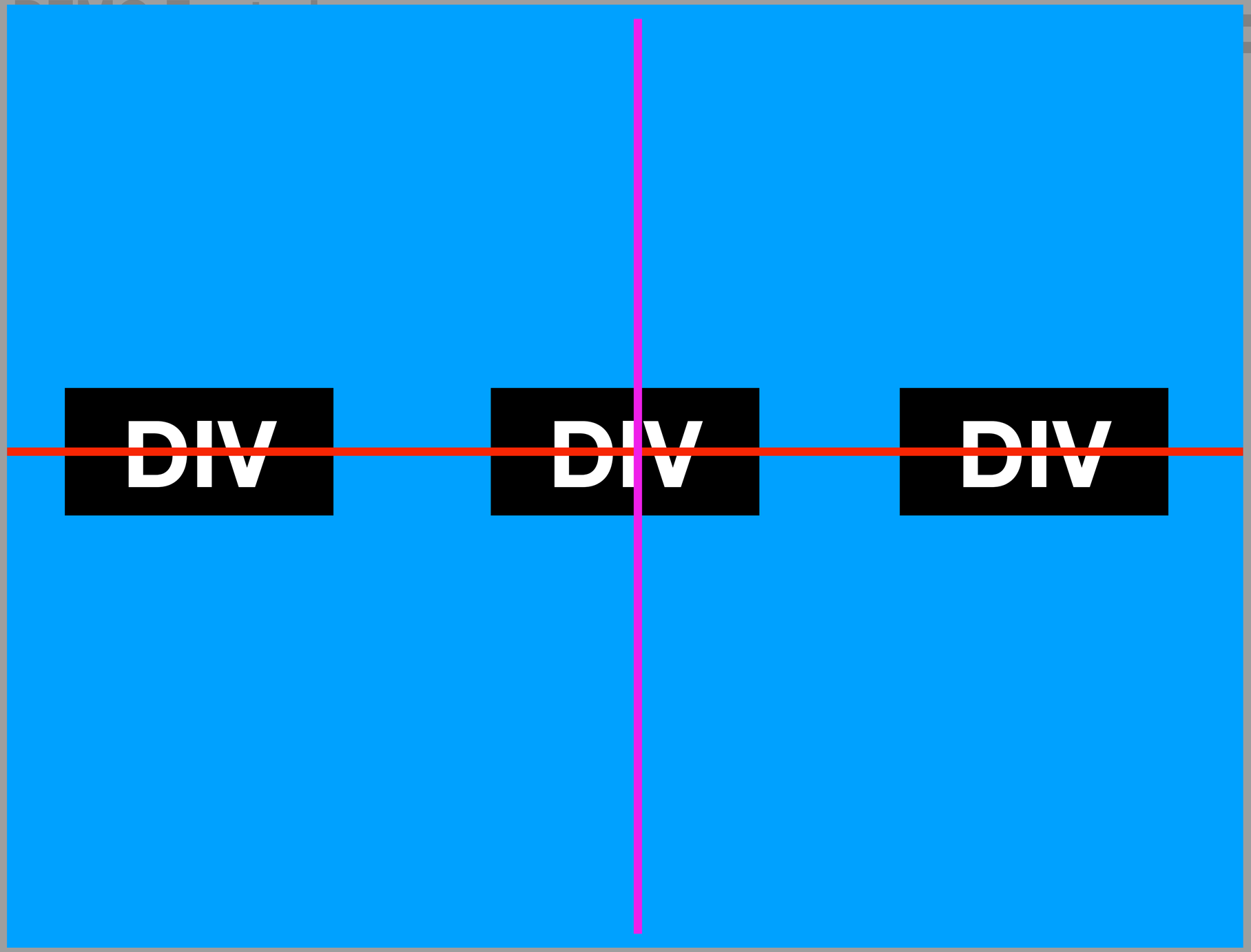
```
.mainContainer{
    display: flex;
}
```




```
.className{

    display: flex;
    flex-direction: row;

}
```




```
.className{

    display: flex;
    flex-direction: row;

    Justify-content:
    space-around;

    }
```



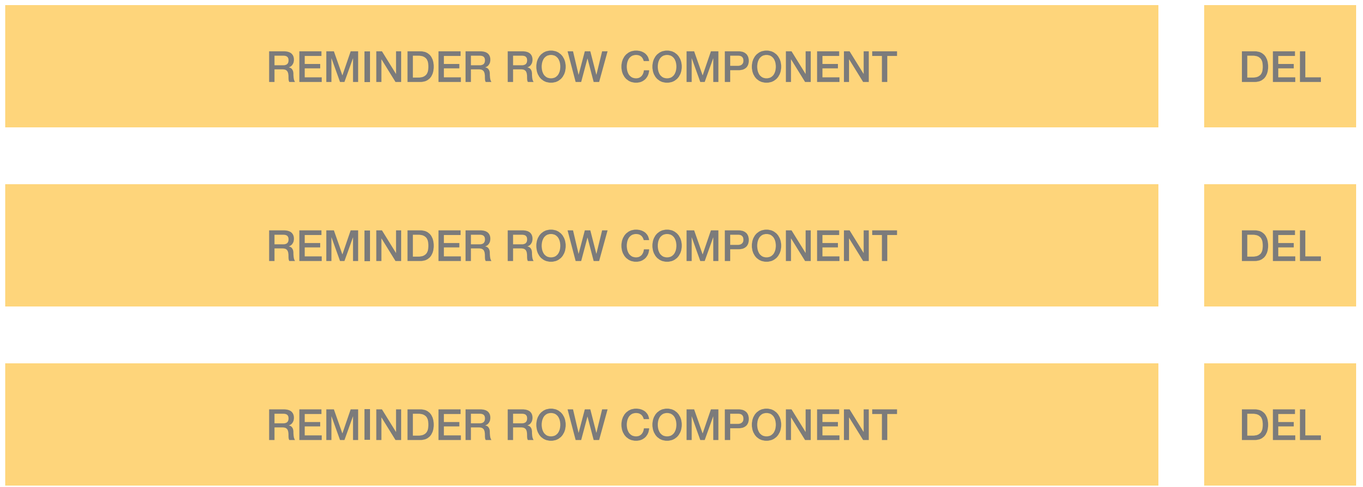

```
.className{

    display: flex;
    flex-direction: row;

    Justify-content:
    space-around;

    Align-content:
    Centre
    }
```

# DEMO For today

 Flex-direction : column

 Flex-direction : row

 Flex-direction : column



# DEMO For today

 Flex-direction : column

 Flex-direction : row

<input/> <button/>

 Flex-direction : column

REMINDER ROW COMPONENT DEL

REMINDER ROW COMPONENT DEL

REMINDER ROW COMPONENT DEL

# DEMO For today

 Flex-direction : column

 Flex-direction : row

<input/> <button/>

 Flex-direction : column

Flex-direction : row

   <button/> DEL 

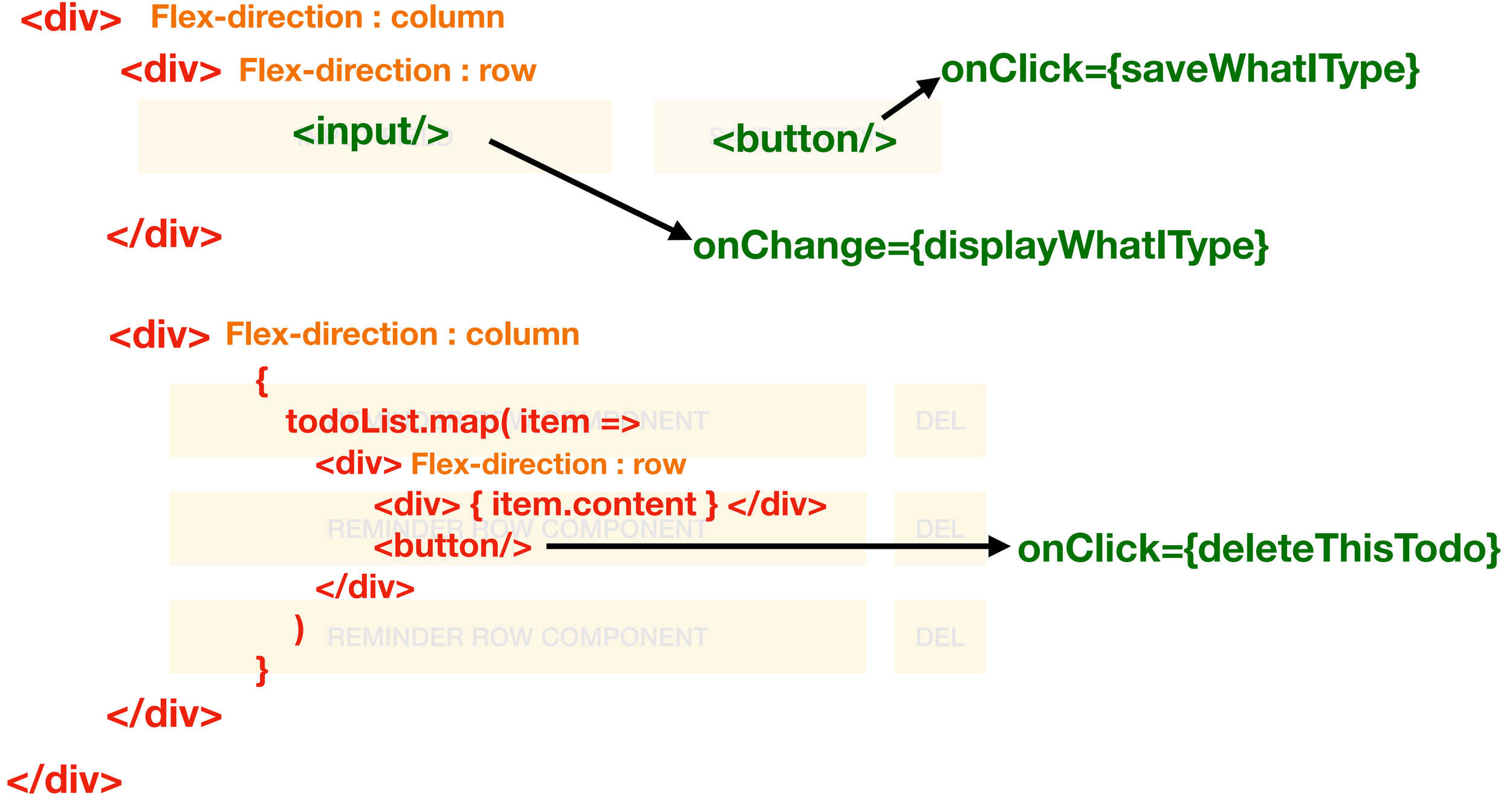
REMINDER ROW COMPONENT DEL

REMINDER ROW COMPONENT DEL

# DEMO For today

```
<div>   Flex-direction : column
    <div>  Flex-direction : row
        <input/>        <button/>
    </div>

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

# DEMO For today



DEMO For today

STATE

inputValue = “lorem ipsum”

todoList = [

{

Id: UNIQUE ID,

Content: “lorem ipsum”

}

]

onClick={saveWhatIType}

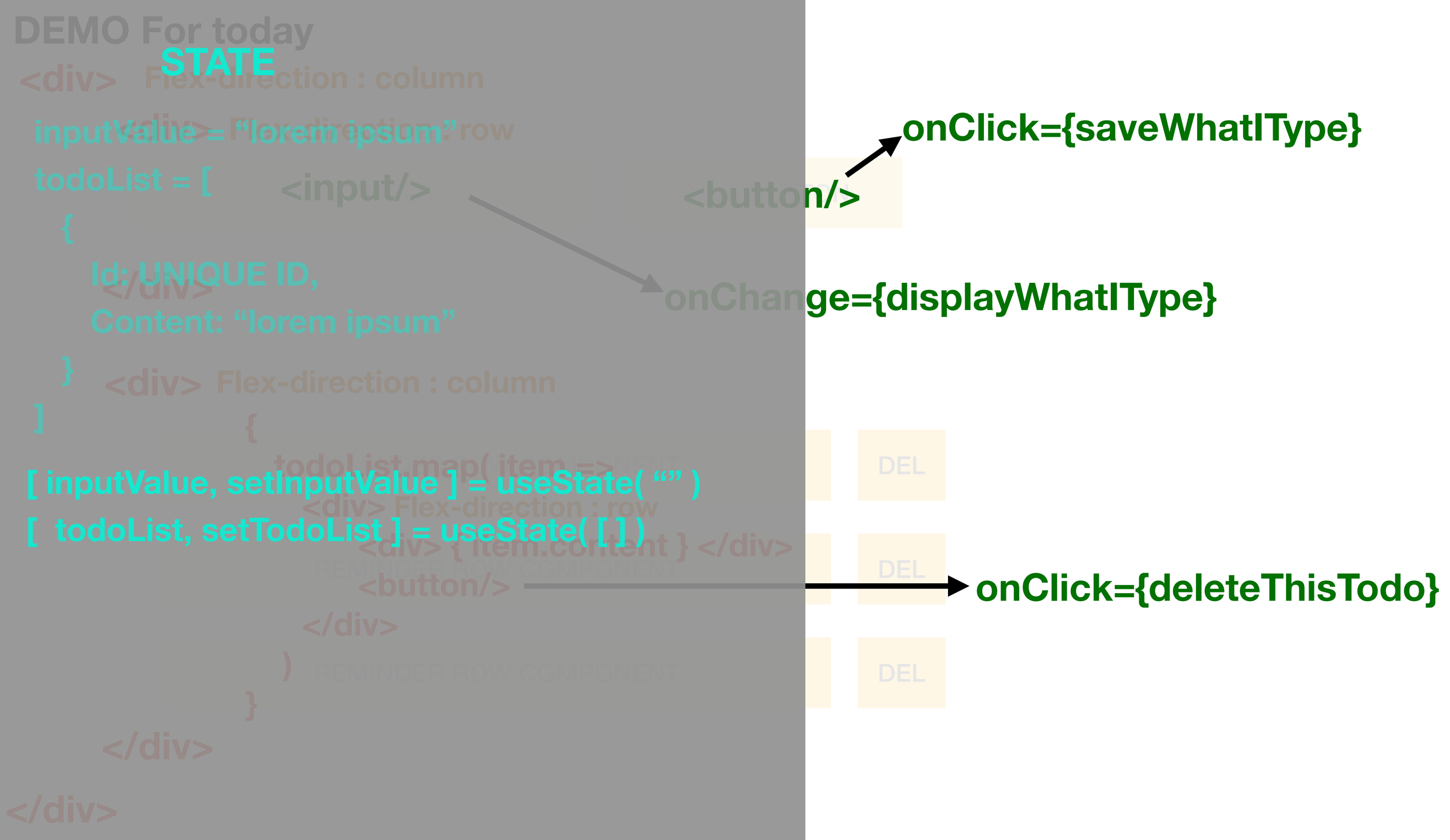
onChange={displayWhatIType}

onClick={deleteThisTodo}

DEMO For today

STATE

```
<div>
  inputValue = “lorem ipsum”
  todoList = [
    {
      Id: UNIQUE ID,
      Content: “lorem ipsum”
    }
  ]

  [ inputValue, setInputValue ] = useState( “” )
  [  todoList, setTodoList ] = useState( [ ] )

</div>
```



DEMO For today

STATE

```
<div>
  inputValue = “lorem ipsum”
  todoList = [
    {
      Id: UNIQUE ID,
      Content: “lorem ipsum”
    }
  ]

[ inputValue, setInputValue ] = useState( “” )
[  todoList, setTodoList ] = useState( [ ] )

</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}

onClick={deleteThisTodo}

**STATE**

[ inputValue, setInputValue ] = useState( “” )

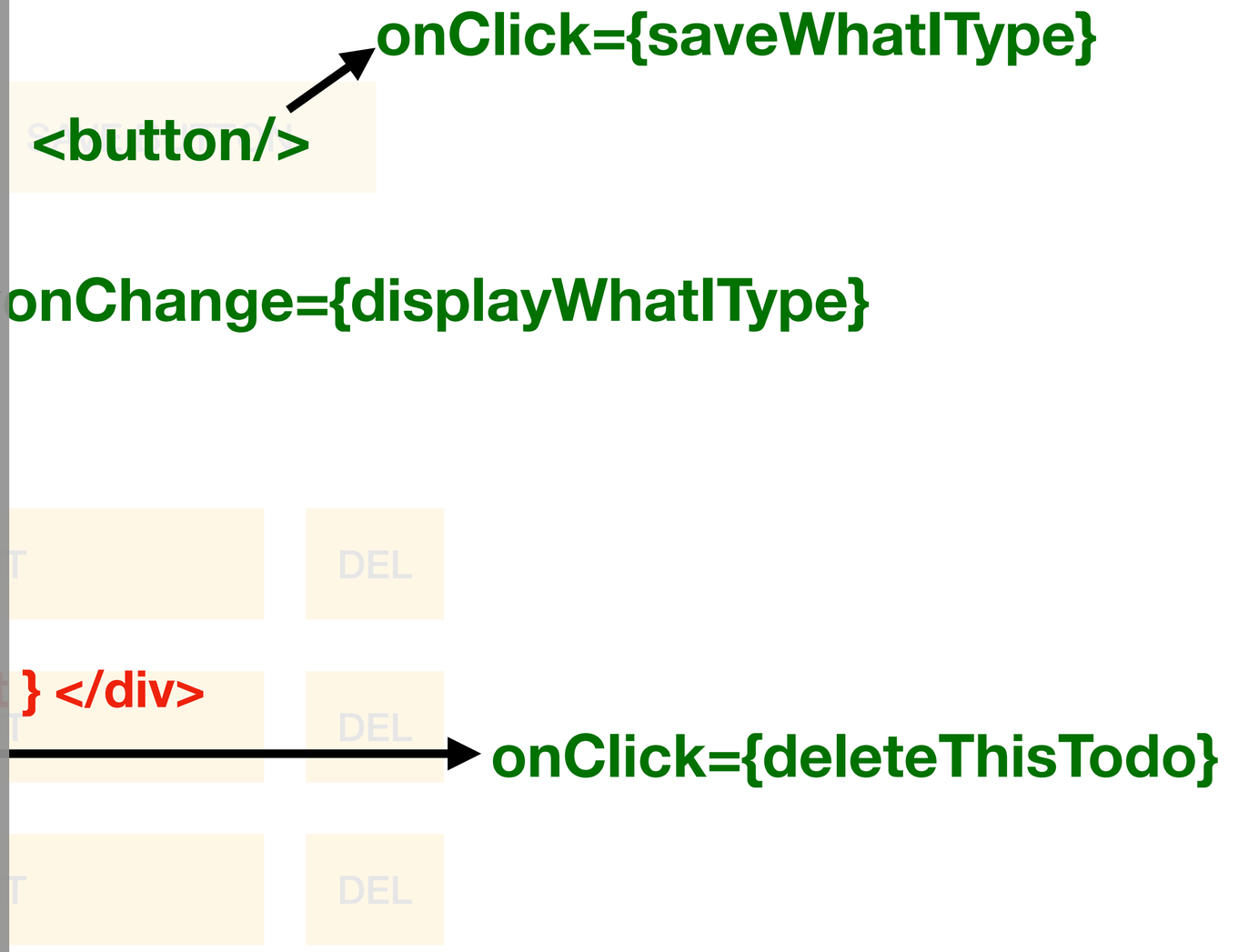
[ todoList, setTodoList ] = useState( [ ] )

**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 ))
}
```



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

onClick={saveWhatIType}

<button/>

onChange={displayWhatIType}

```
    </div>

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

</div>
```

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

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

# 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

.
.
.

Bryce - telegram @brycerice