Intro to Web Dev and React

3DC

Bryce Goh

Basic React Introduction

Basic overview of the web

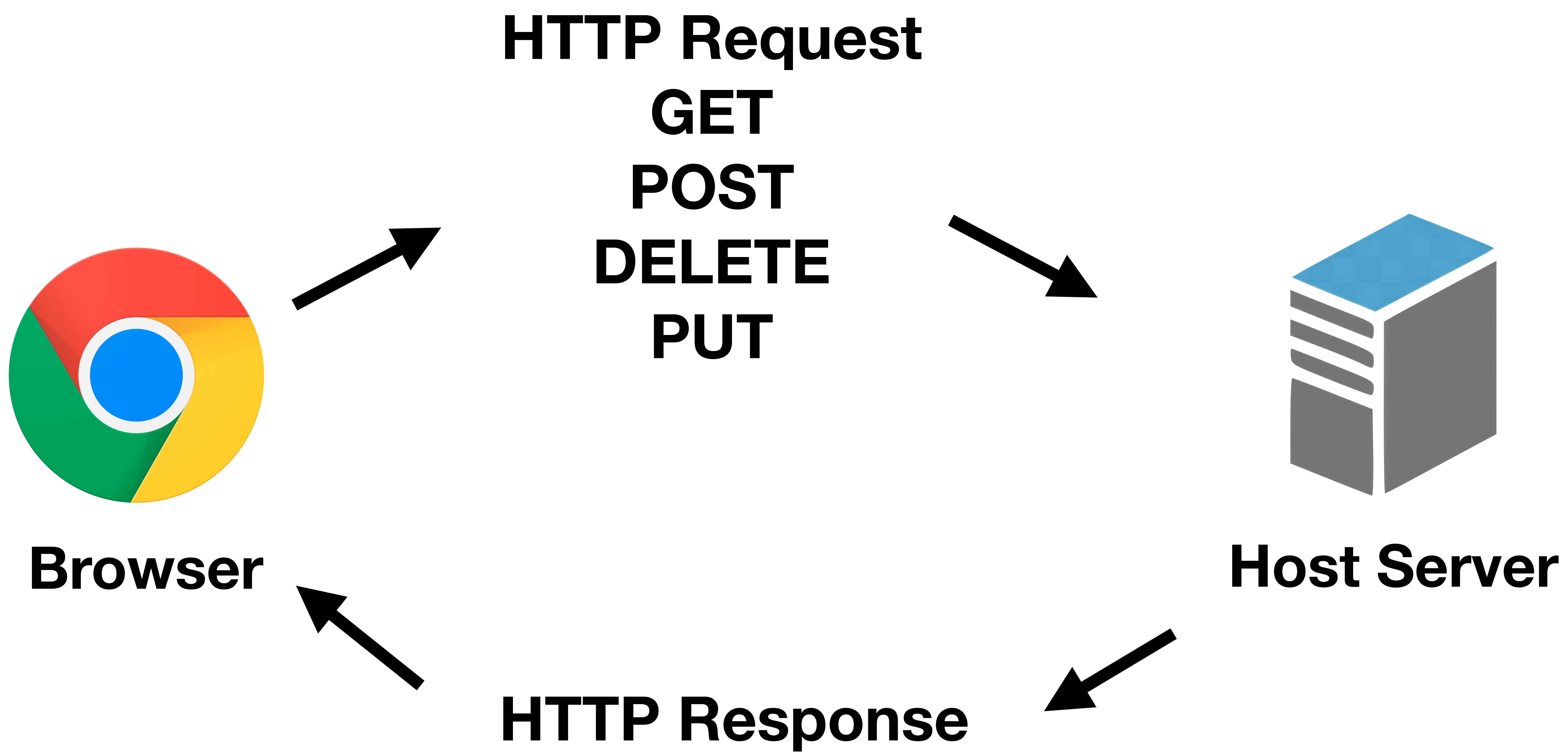
Why react

Javascript Refreshers

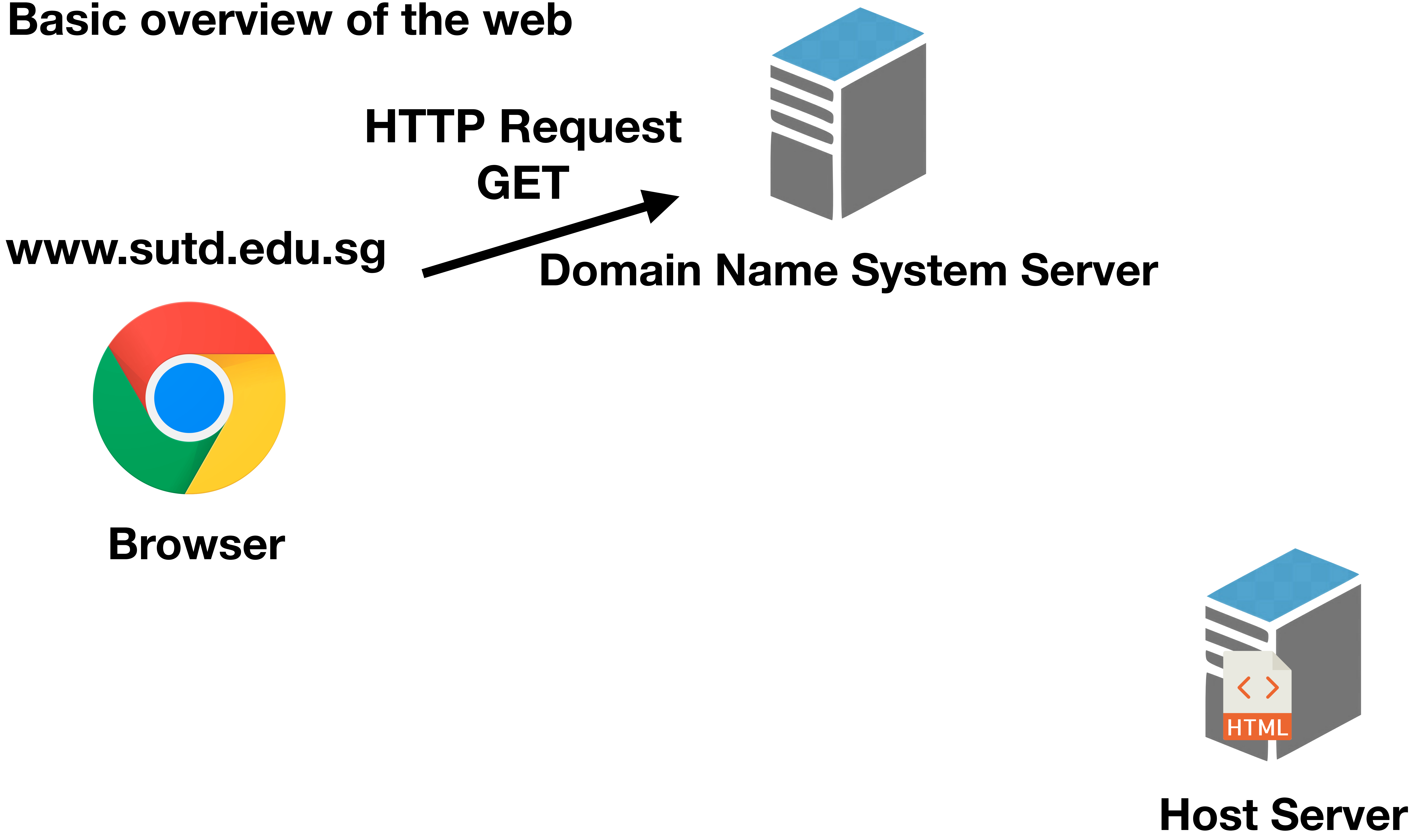
Components and JSX

To-do App

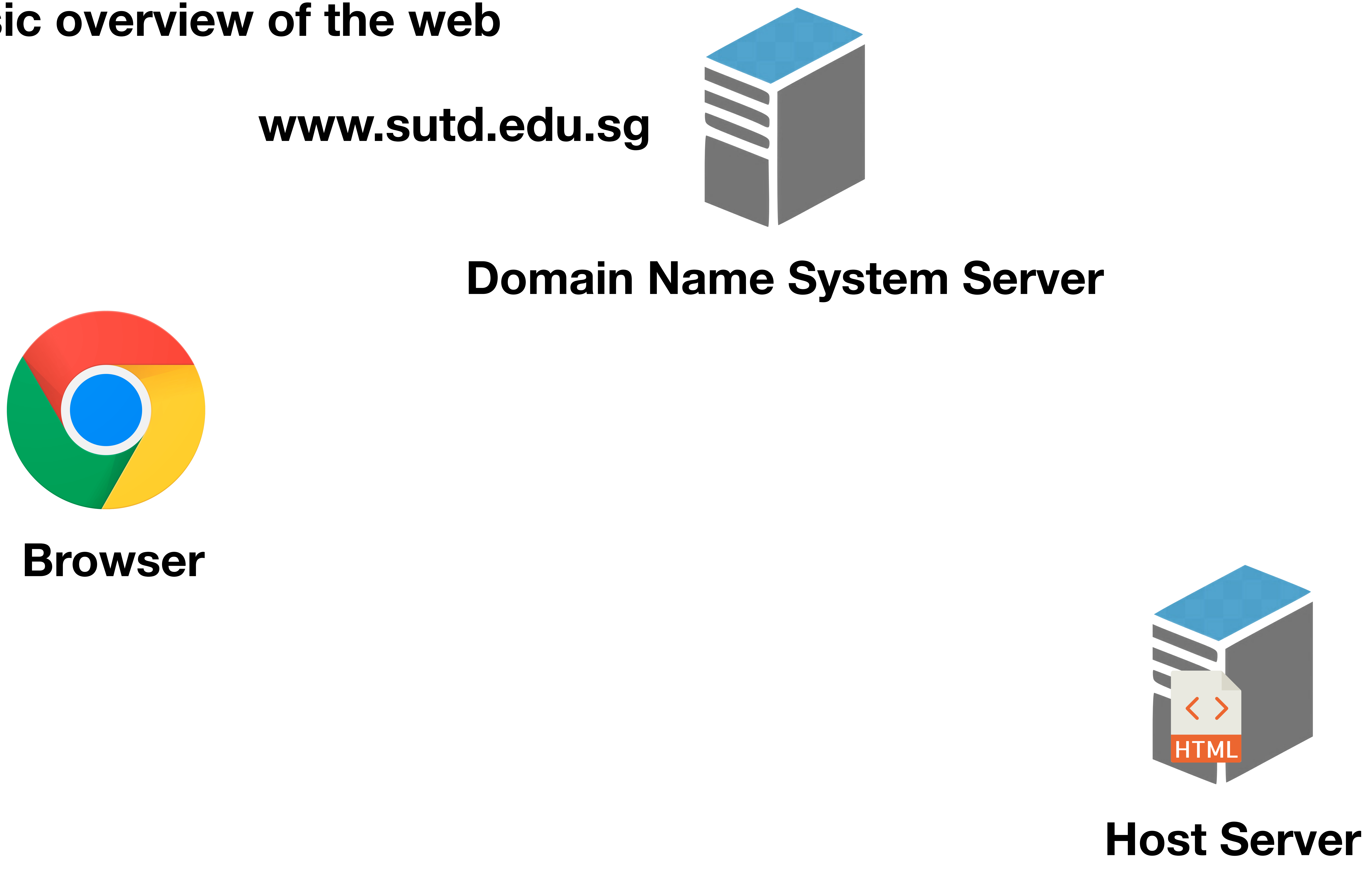
Basic overview of the web



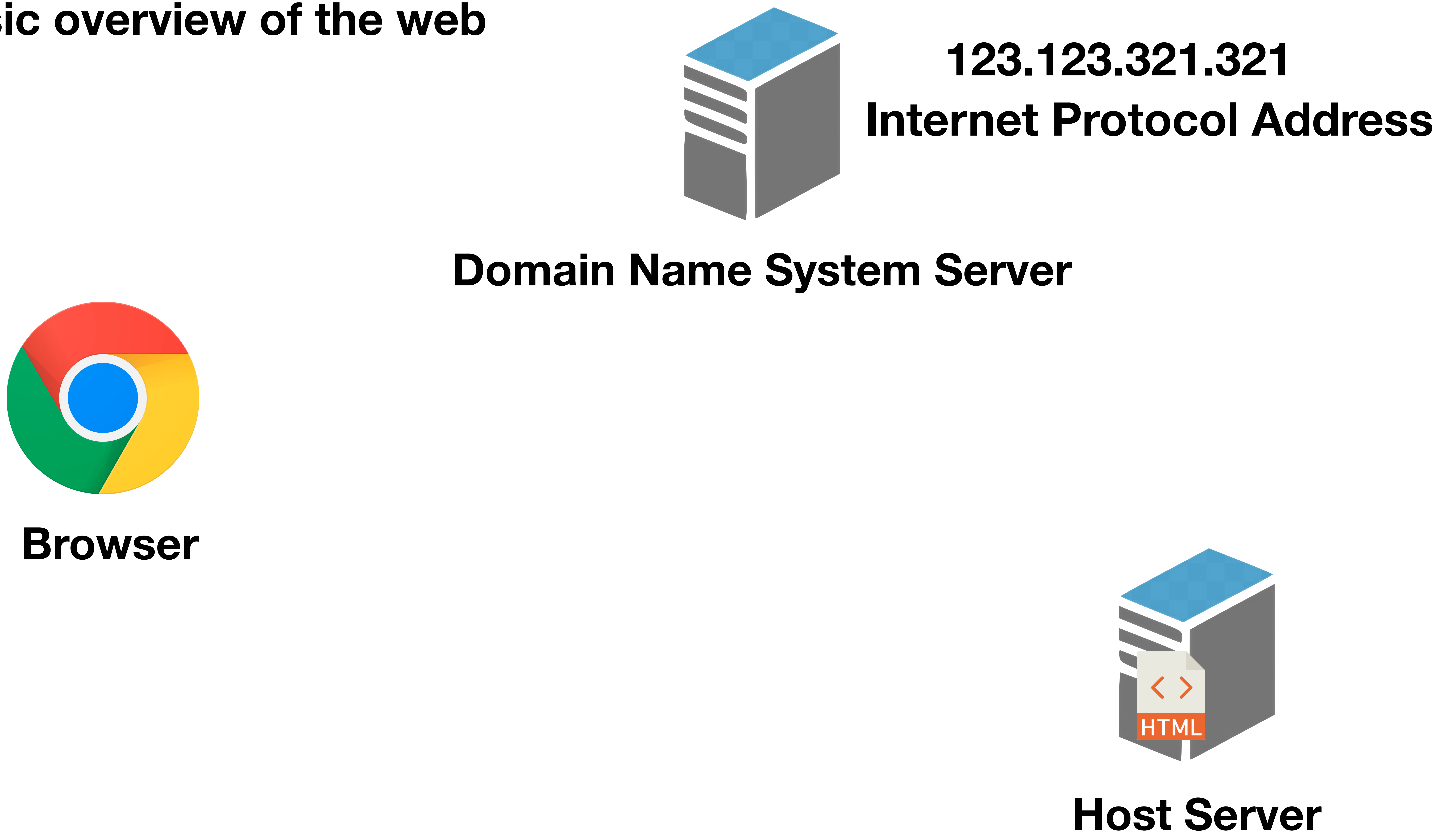
Basic overview of the web



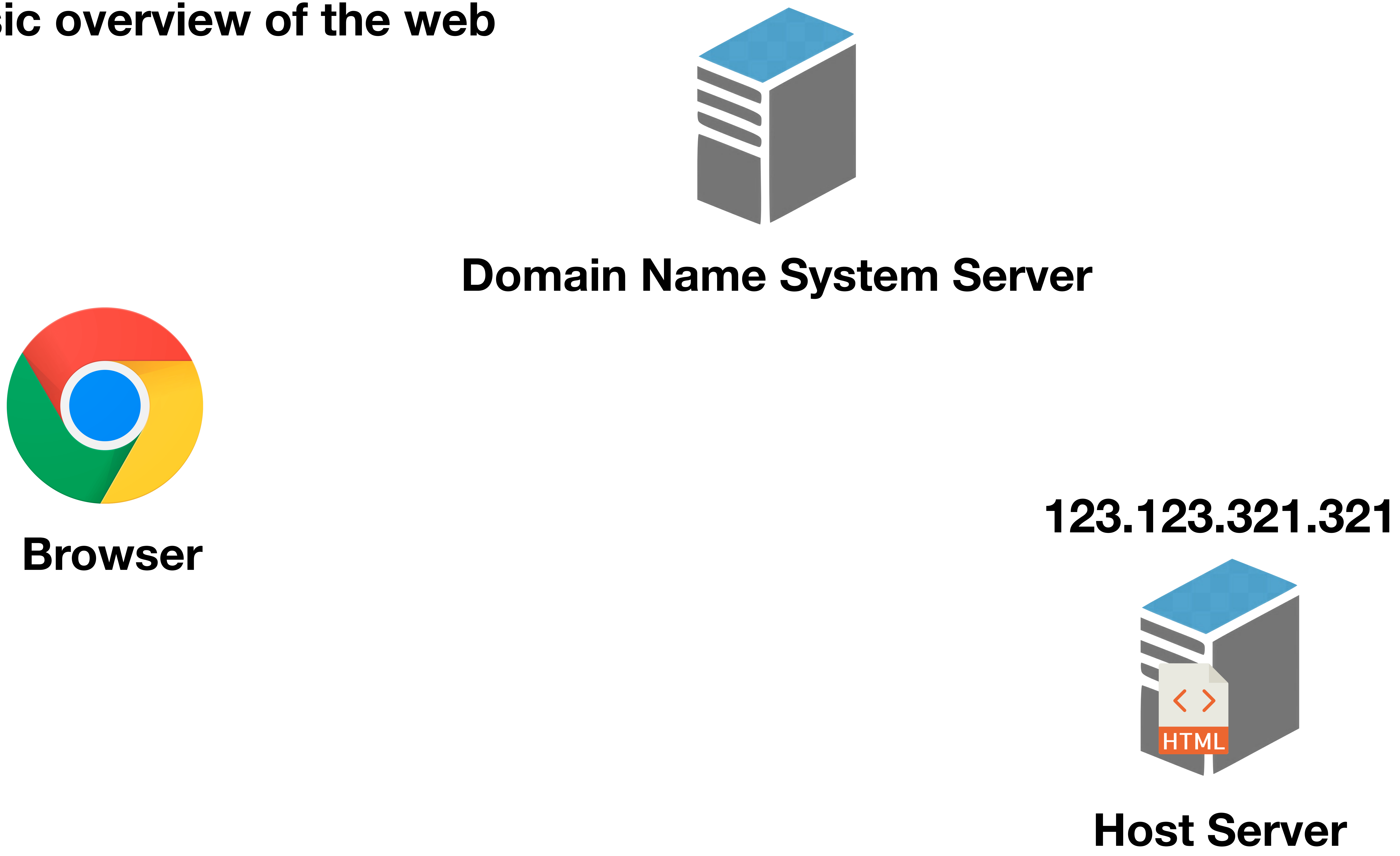
Basic overview of the web



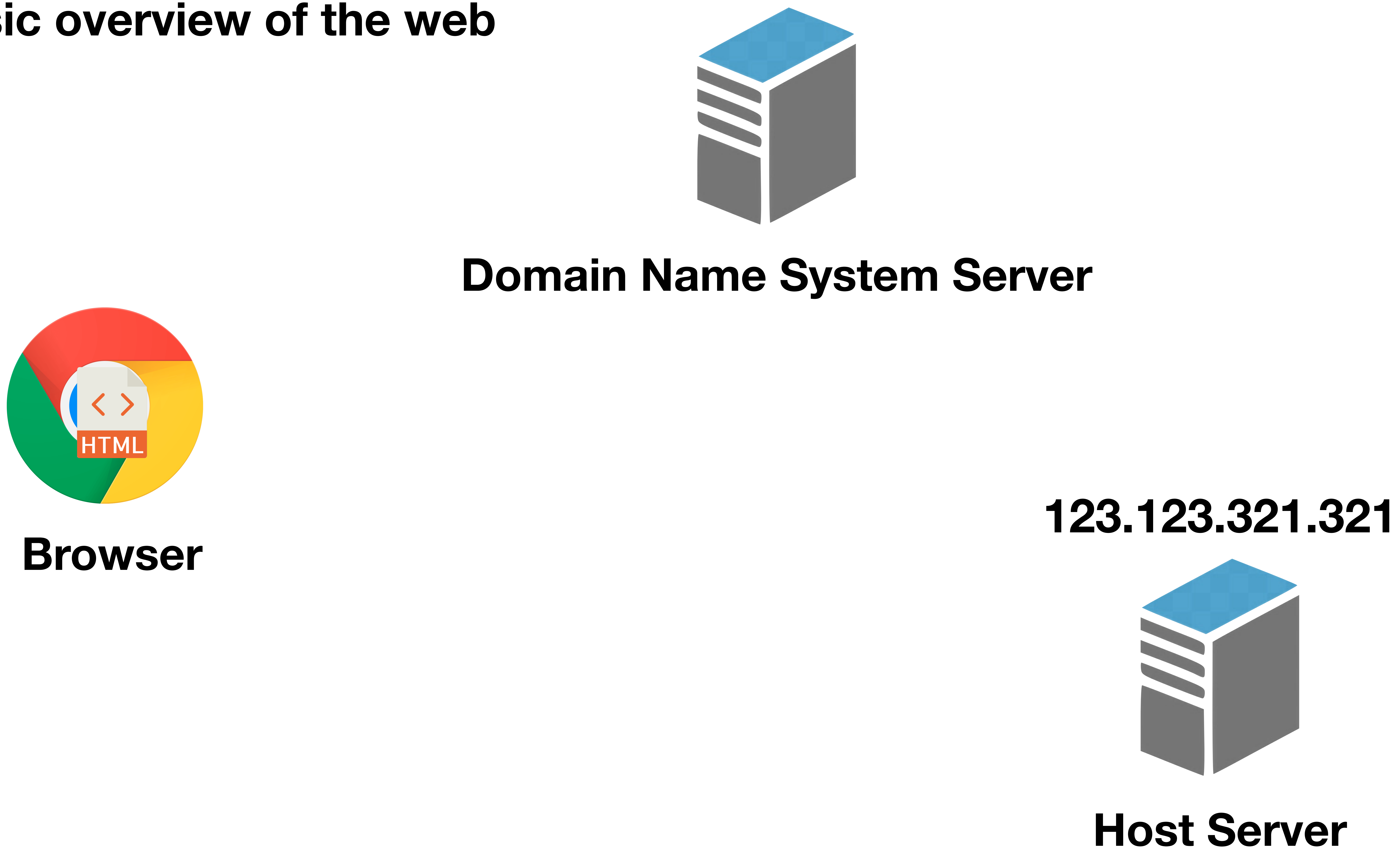
Basic overview of the web



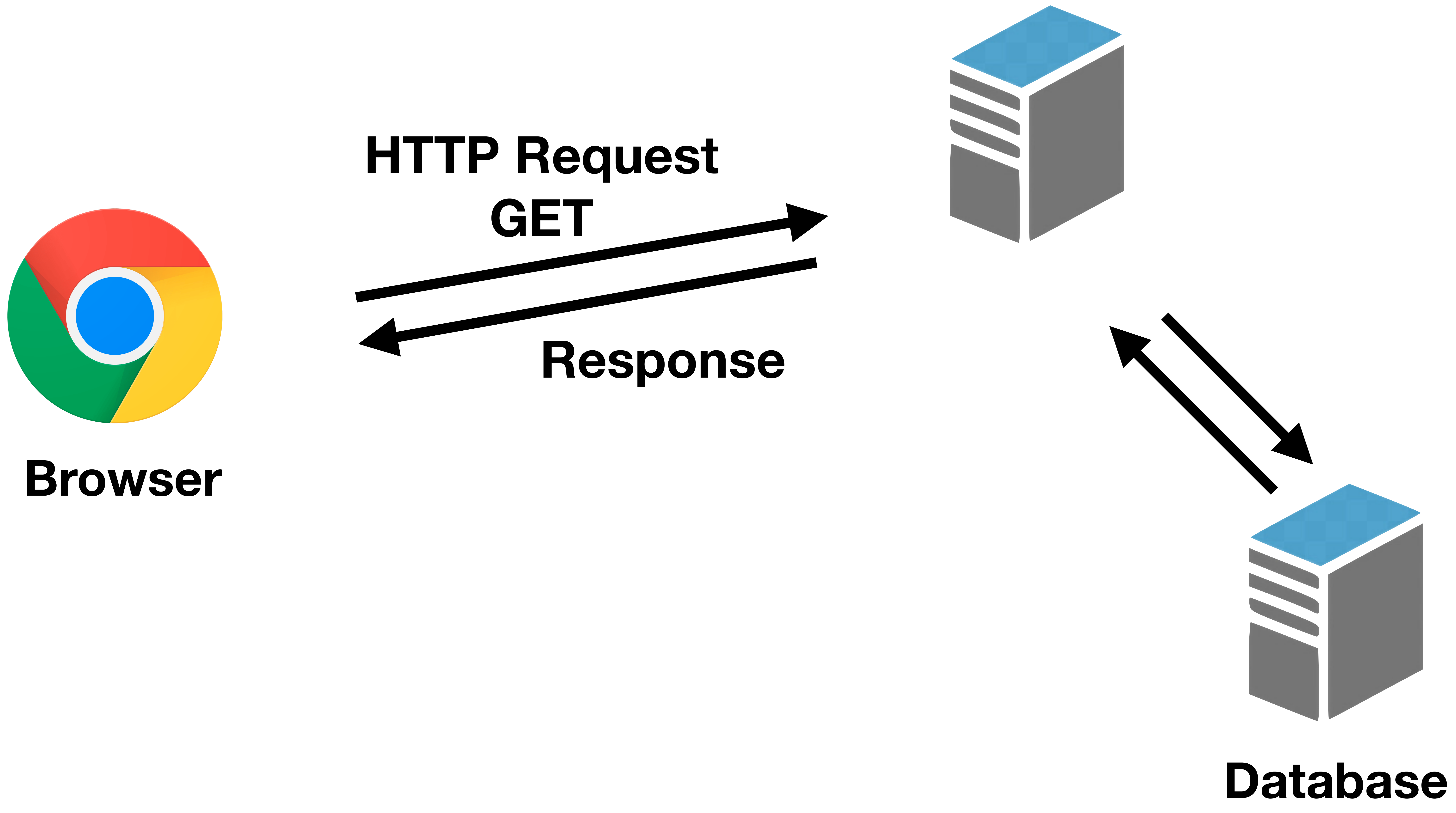
Basic overview of the web

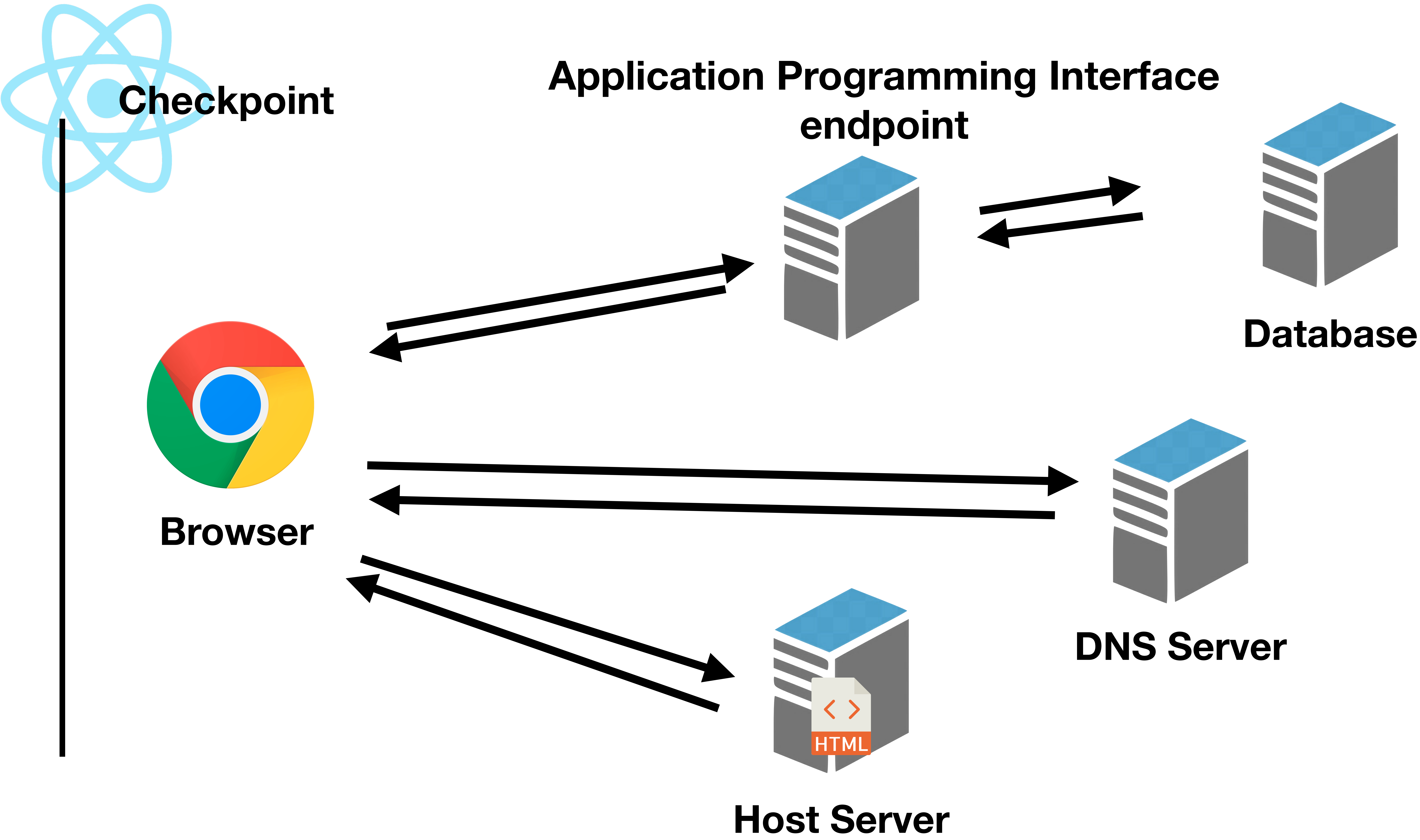


Basic overview of the web



Basic overview of the web

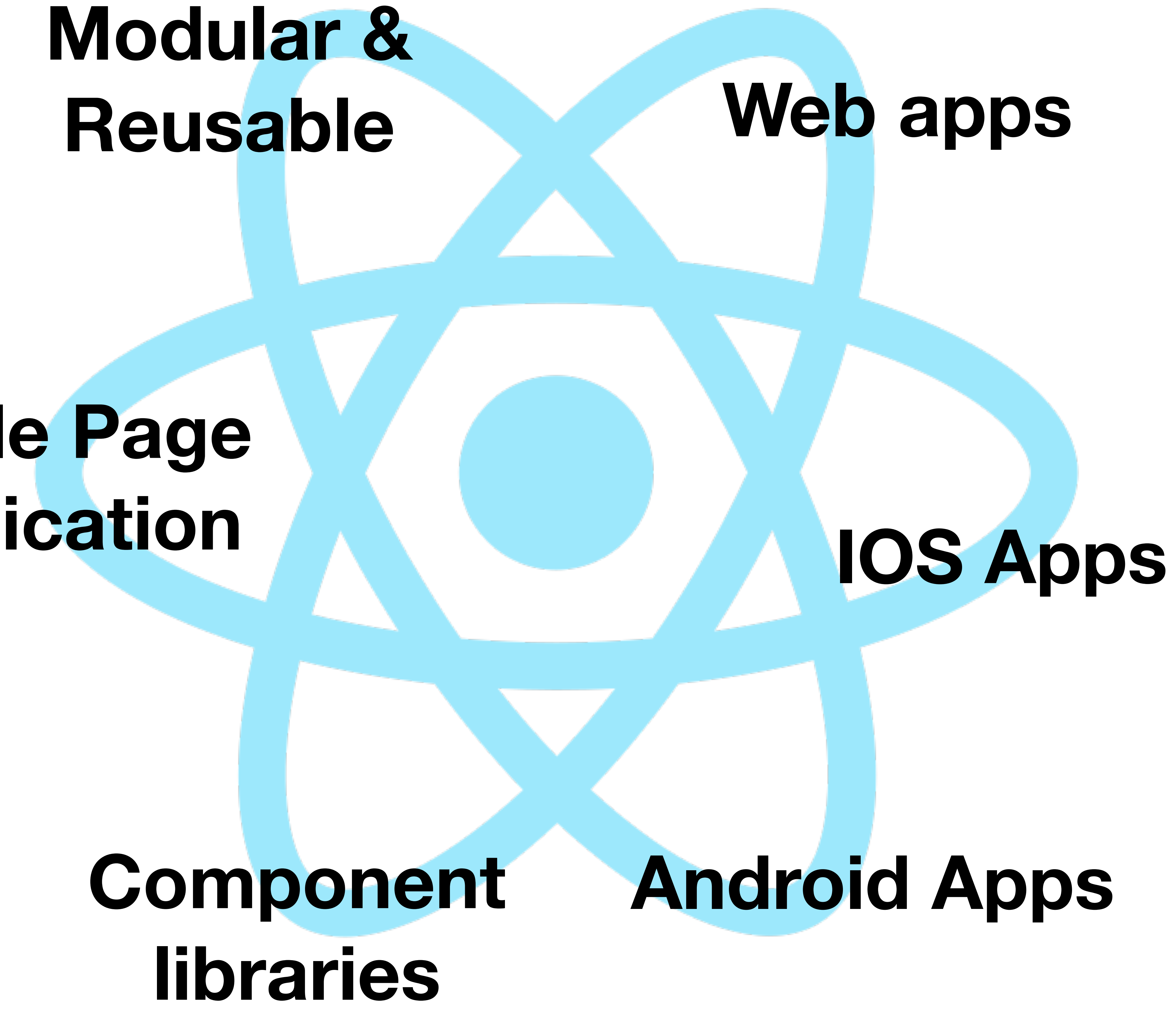



Checkpoint
Application Programming Interface endpoint
Database
Browser
DNS Server
HTML
Host Server

Checkpoint

Questions?

Why react


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

Javascript Refreshers: Variables

```
# PYTHON
school = "SUTD" #creates a variable named school and assign a string "SUTD" to it
```

```
// JAVASCRIPT
const school = "SUTD" //creates a constant variable named school
let school = "SUTD" //creates a variable named school and assign a string "SUTD" to it
```

Javascript Refreshers: Arrays

```python
# PYTHON
thisIsAnArray = [] #creates a variable named thisIsAnArray and assigns an empty array to it
```

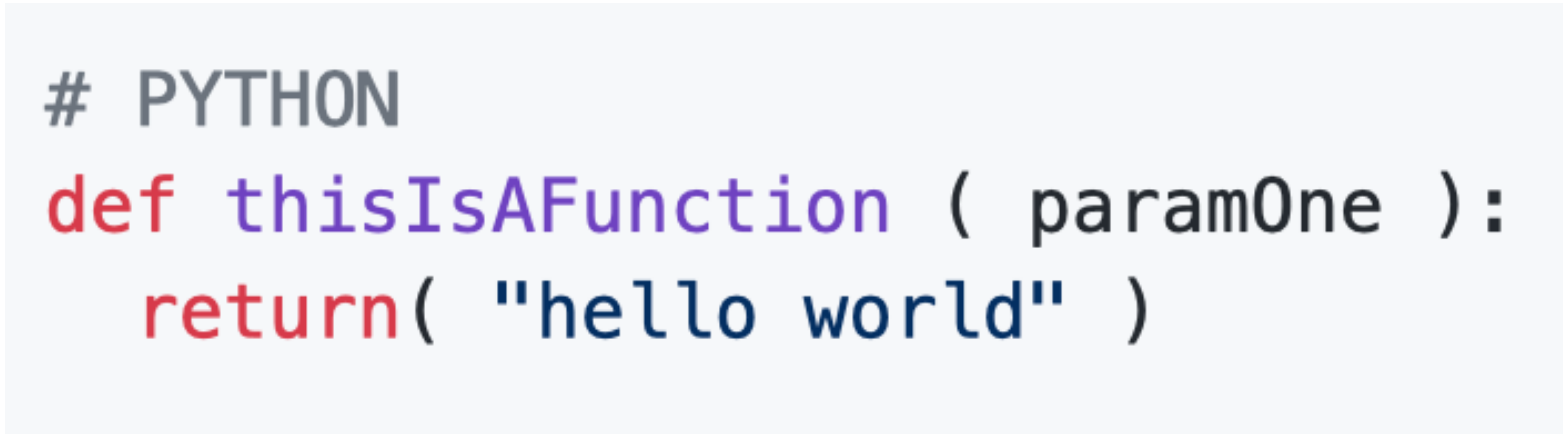
```javascript
// JAVASCRIPT
const thisIsAnArray = [] //creates a constant variable named thisIsAnArray and assigns an empty a
let thisIsAnArray = [] //creates a variable named thisIsAnArray and assigns an empty array to it
```

Javascript Refreshers: Objects

```
# PYTHON
thisIsAnArray = {
  key: "value"
} #creates a variable named thisIsAnArray and assigns an object to it.
```

```
// JAVASCRIPT
const thisIsAnArray = {
  key: "value"
} //creates a constant variable named thisIsAnArray and assigns an object to it.
let thisIsAnArray = {
  key: "value"
} //creates a variable named thisIsAnArray and assigns an object to it.
```

Javascript Refreshers: Functions



```
# PYTHON
def thisIsAFunction ( paramOne ):
  return( "hello world" )
```

Javascript Refreshers: Functions

```
----Function Expression----
Function is loaded when the line is reached
*/

// The below function is called a Arrow Function (I guess because of the => )
const thisIsAFunction = ( paramOne ) => {
  return( "hello world" )
}
const thisIsAFunction = ( paramOne ) => "hello world"
// Notice how this version of the Arrow Function, doesn't have the curly braces {}
// Without the curly braces, the function automatically return whatever after that. In this case,
const thisIsAFunction = function( paramOne ){
  return("hello world)
}
```

```
/*
----Function Declaration----
Function Declaration are hoisted to the top of the code.
Meaning the funtion is loaded before anything else
*/

function thisIsAFunction( paramOne ){
  return( "hello world" )
}
```

Javascript Refreshers: Looping over array

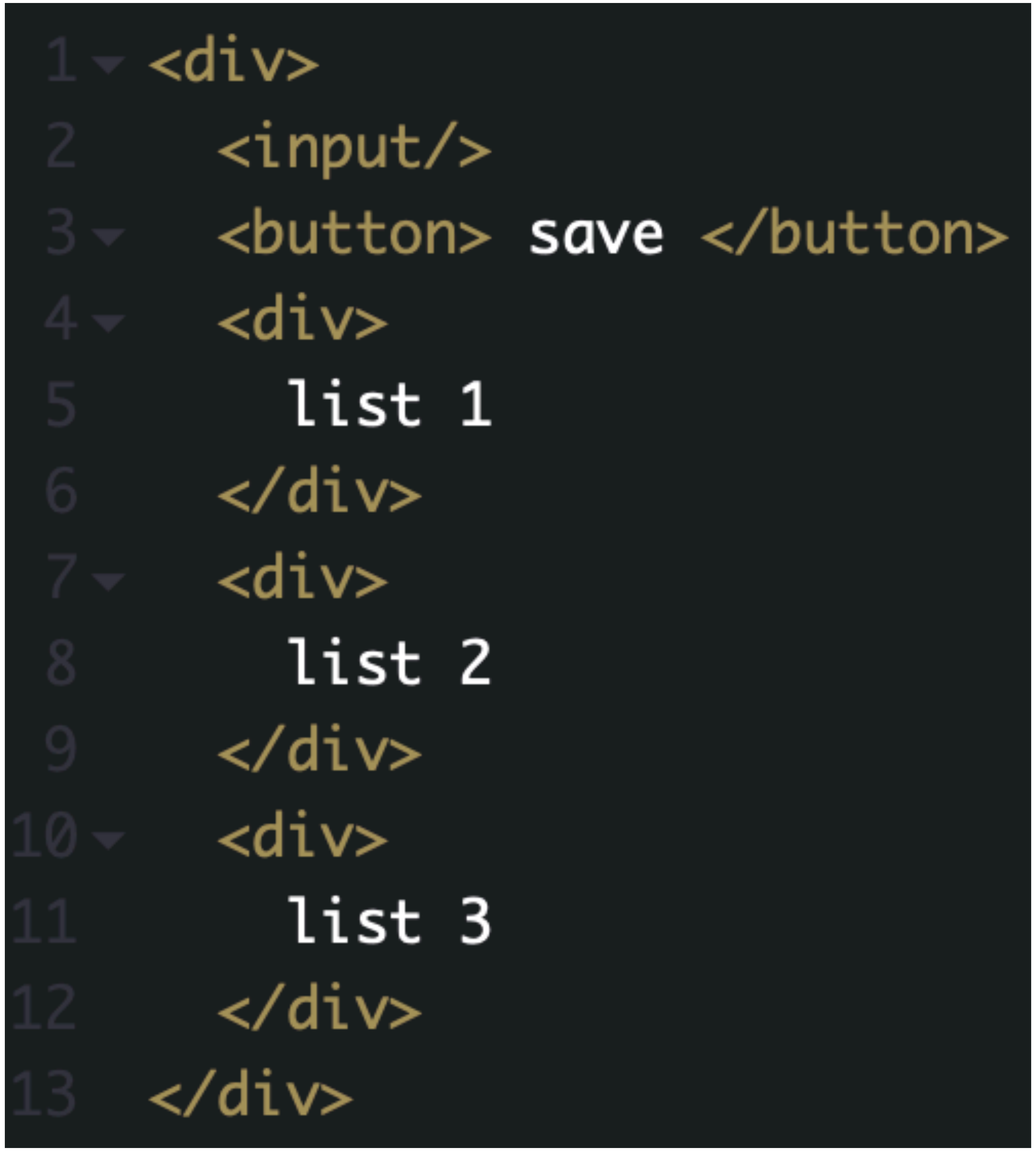
```
oneToTen = [ 1,2,3,4,5,6,7,8,9,10 ]

for eachElement in oneToTen:
  print(eachElement)
# prints 1 2 3 4 ...
```

```

const oneToTen = [ 1,2,3,4,5,6,7,8,9,10 ]

oneToTen.forEach( eachElement => console.log(eachElement) )
// prints 1 2 3 4 ...

// Another method to reiterate over an array is
oneToTen.map( (eachElement, eachIndex)=> console.log(eachElement, eachIndex) )

/*
  Take note that .map iterates over the array and returns a new array with the s
  .forEach only iterates over the array and does not return anything.
*/
//For example
const newArray = oneToTen.map(x => x * x)
console.log(newArray) // this will print a new array named newArray in which eac
```

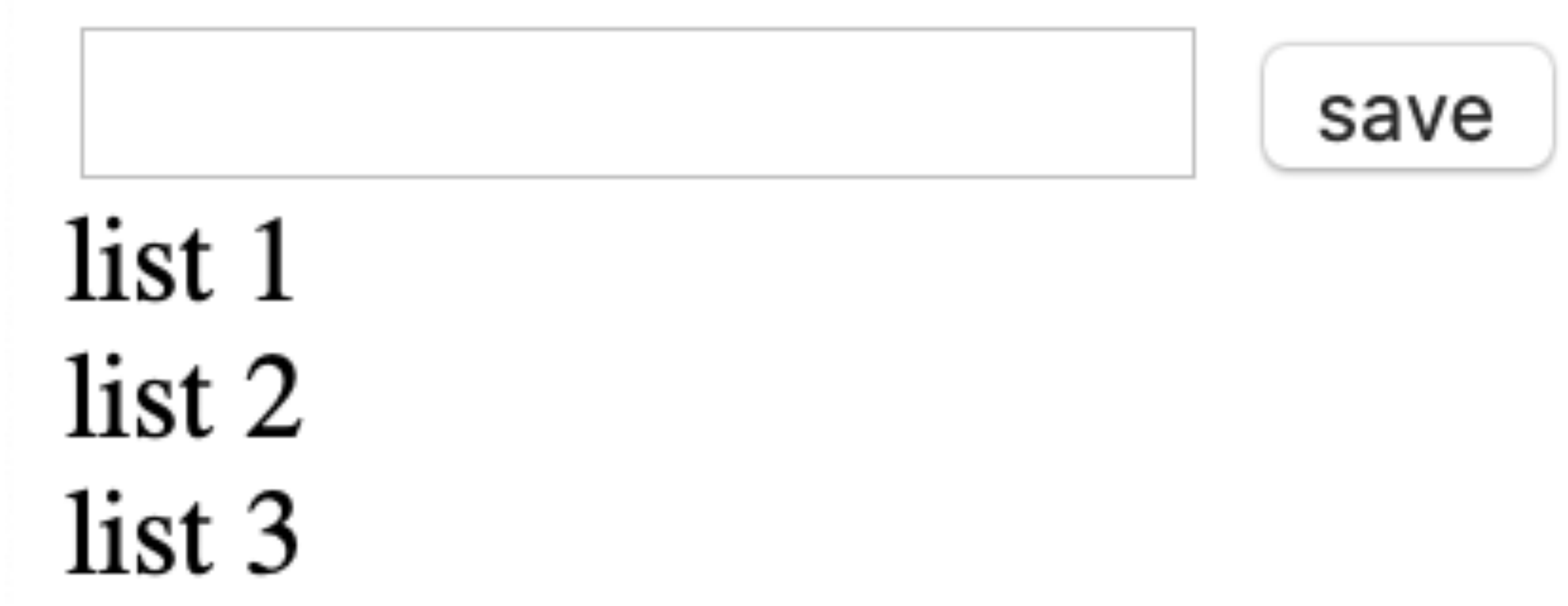
Checkpoint

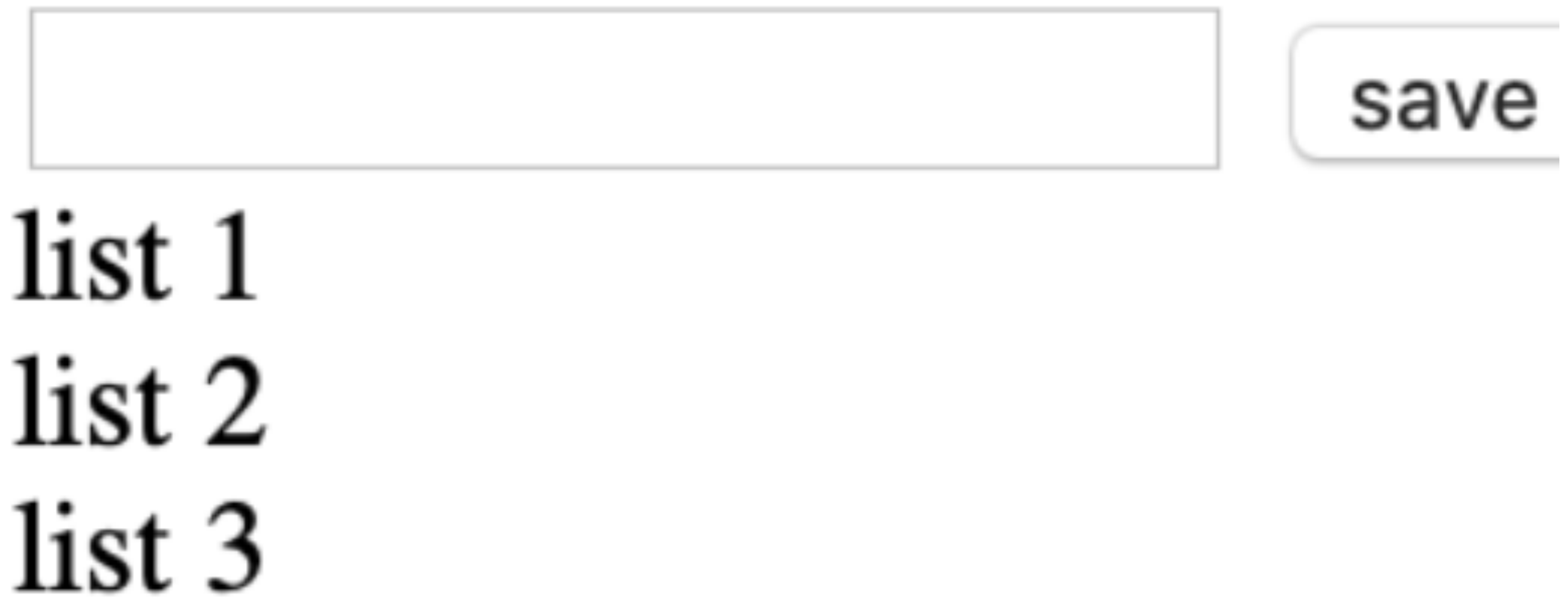
Questions?

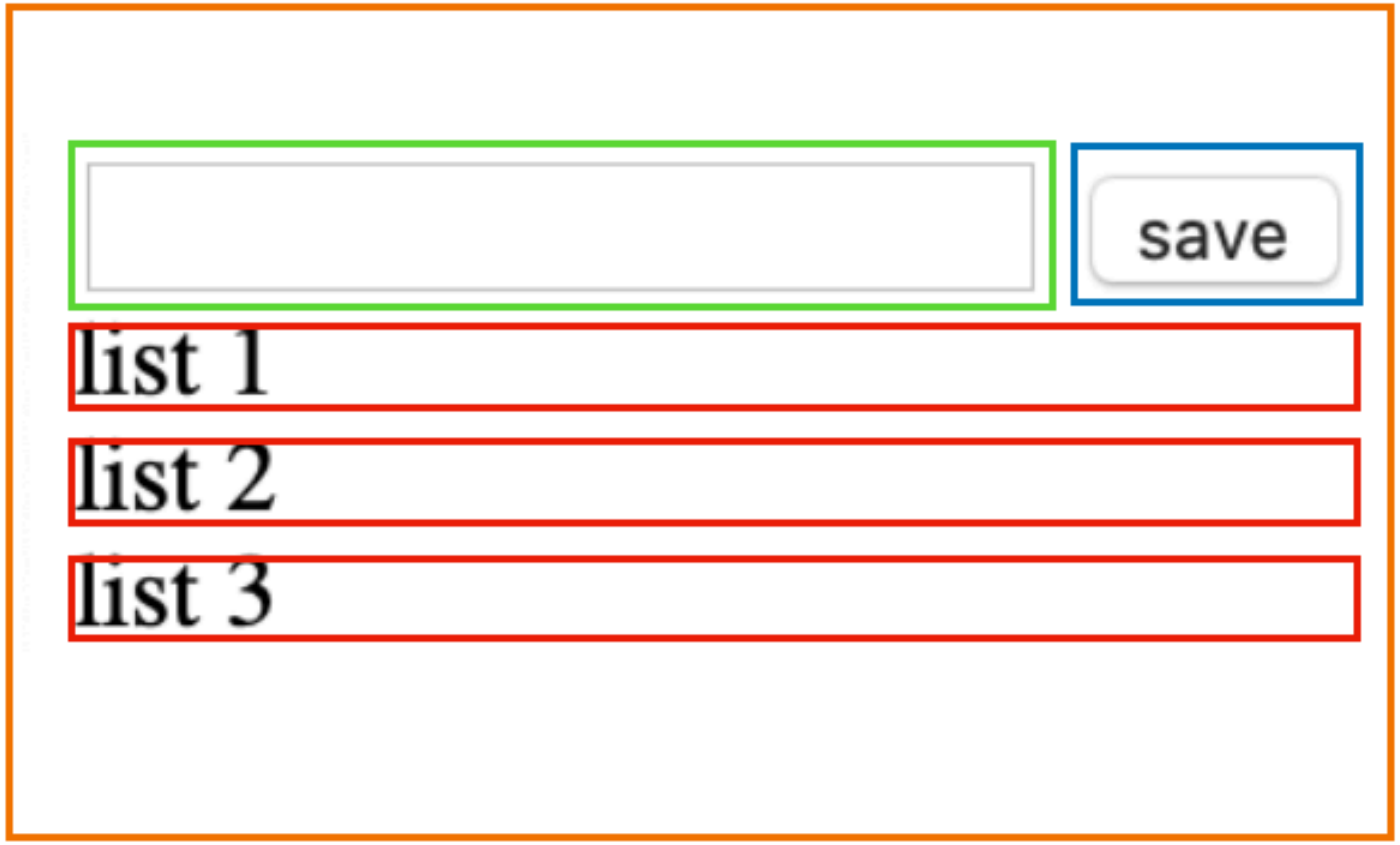
HTML/CSS Refreshers



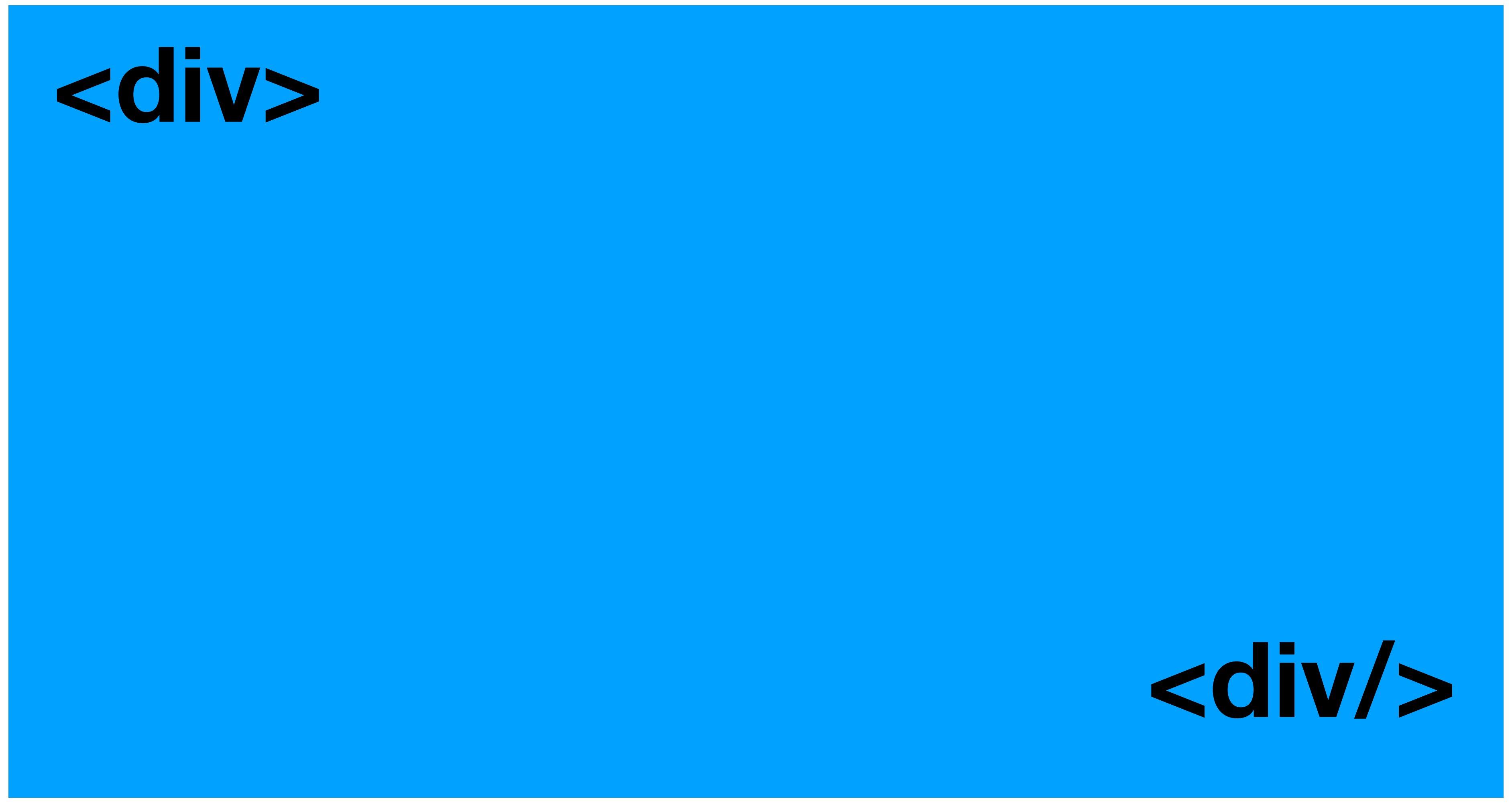
```
<div>
  <input/>
  <button> save </button>
  <div>
    list 1
  </div>
  <div>
    list 2
  </div>
  <div>
    list 3
  </div>
</div>
```



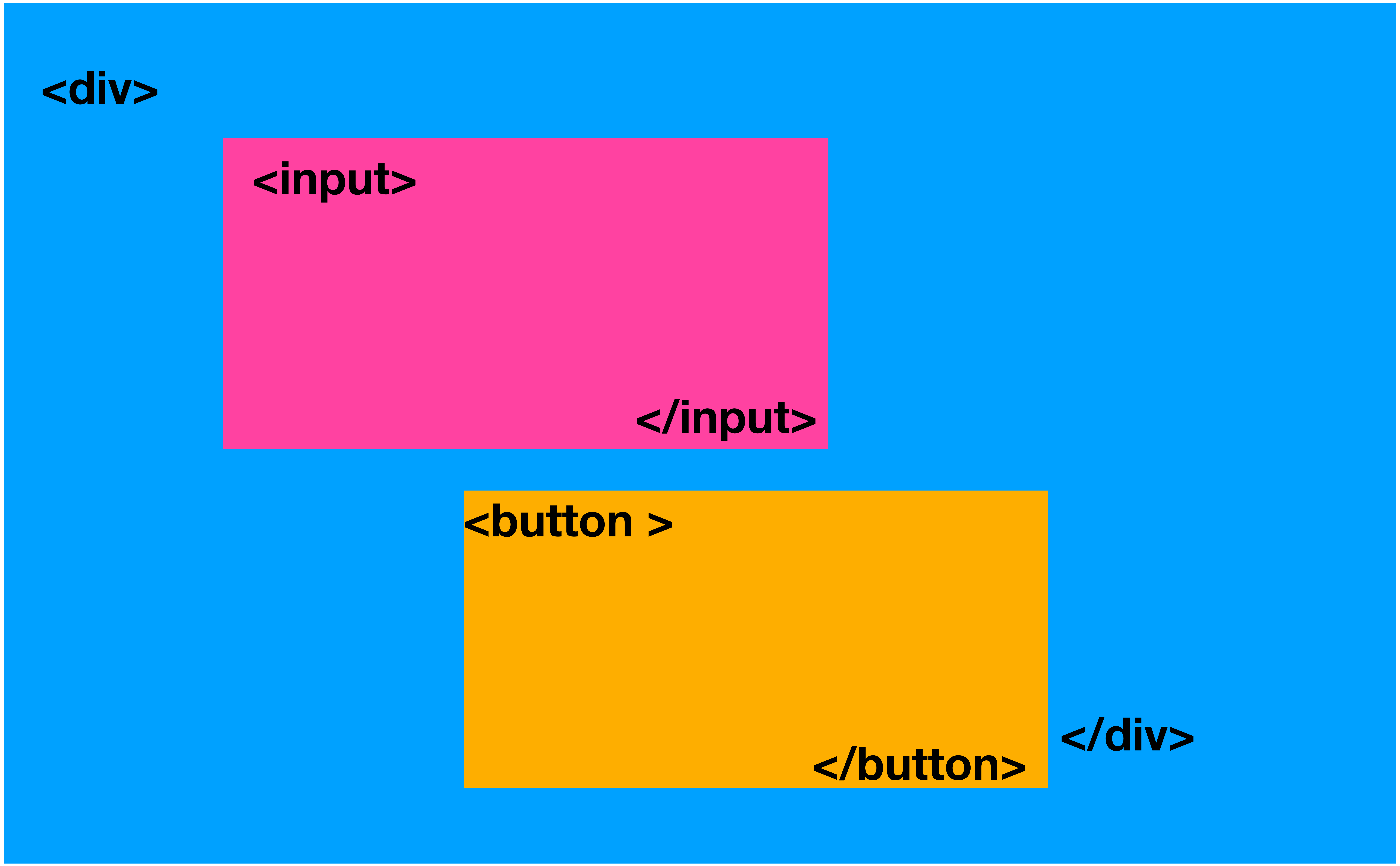

save
list 1
list 2
list 3


save
list 1
list 2
list 3

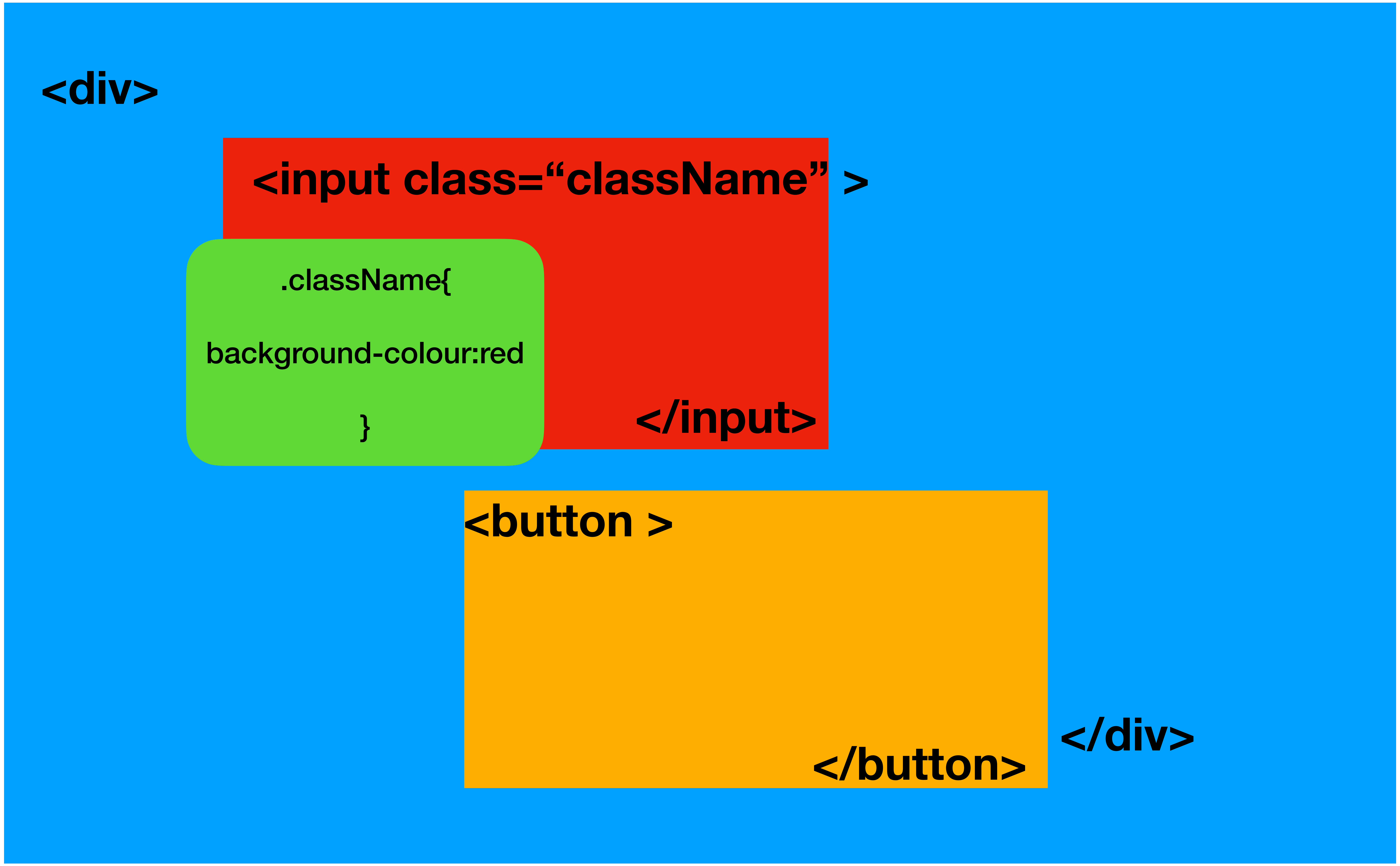
HTML/CSS Refreshers



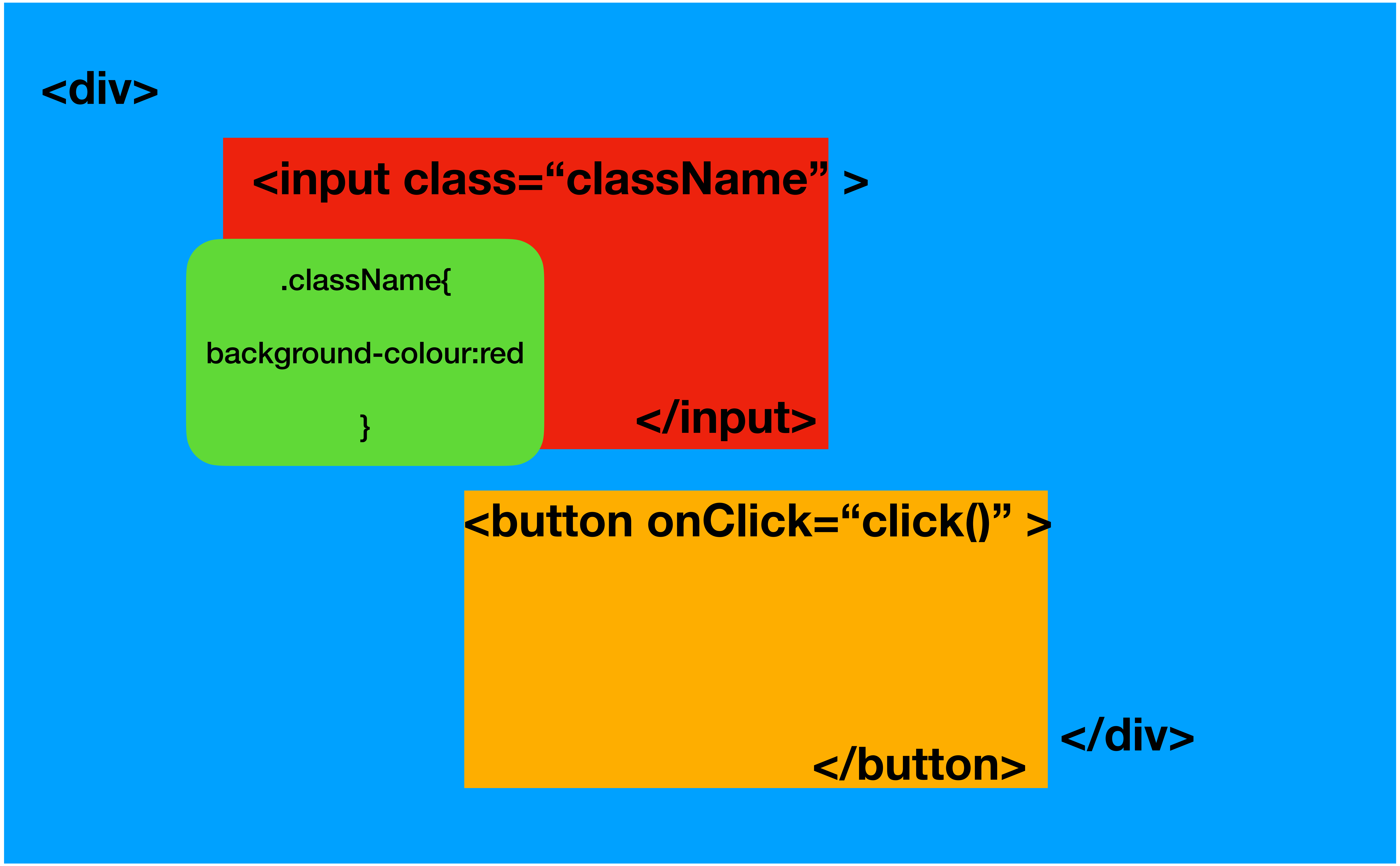
HTML/CSS Refreshers



HTML/CSS Refreshers

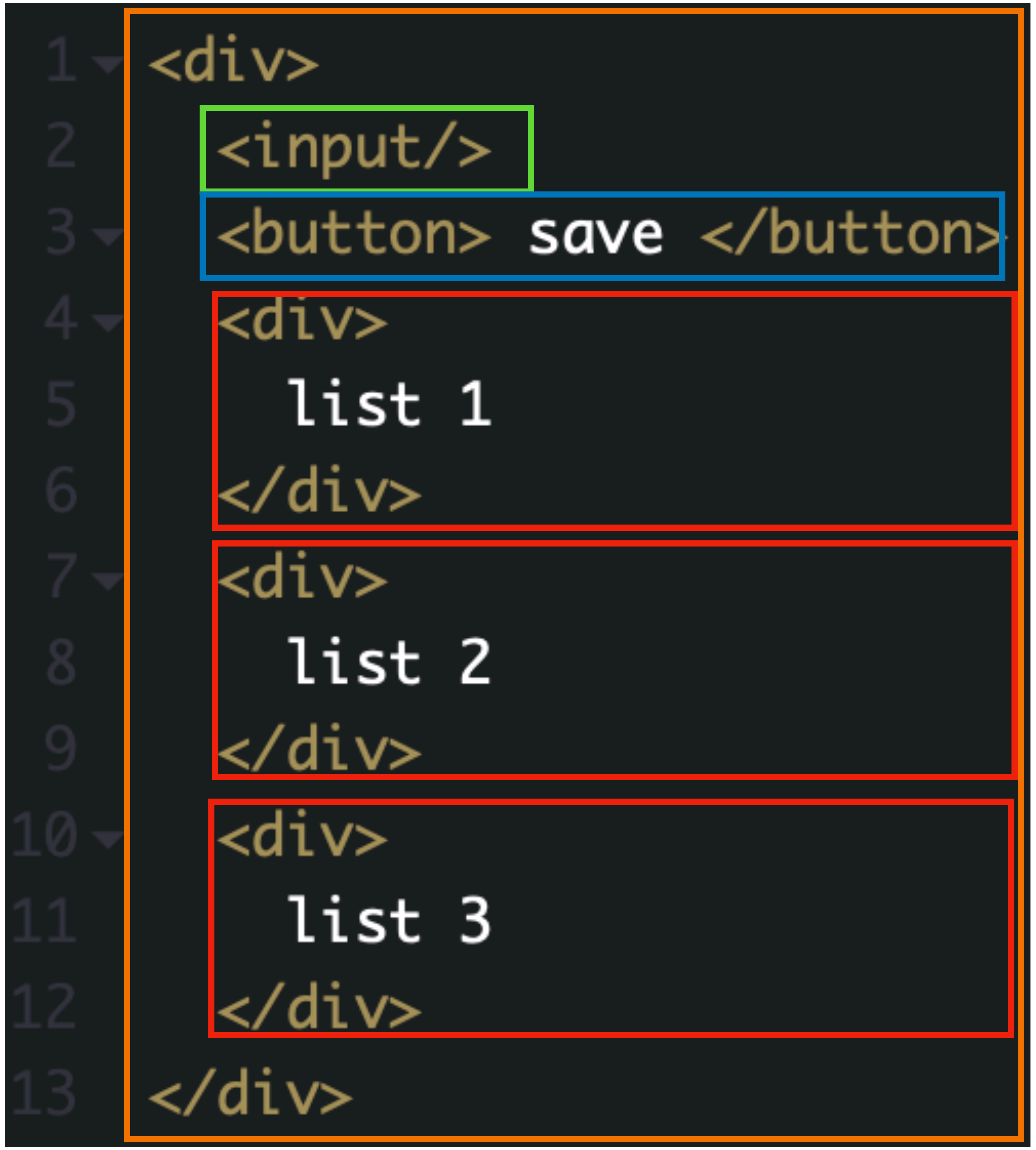


HTML/CSS Refreshers

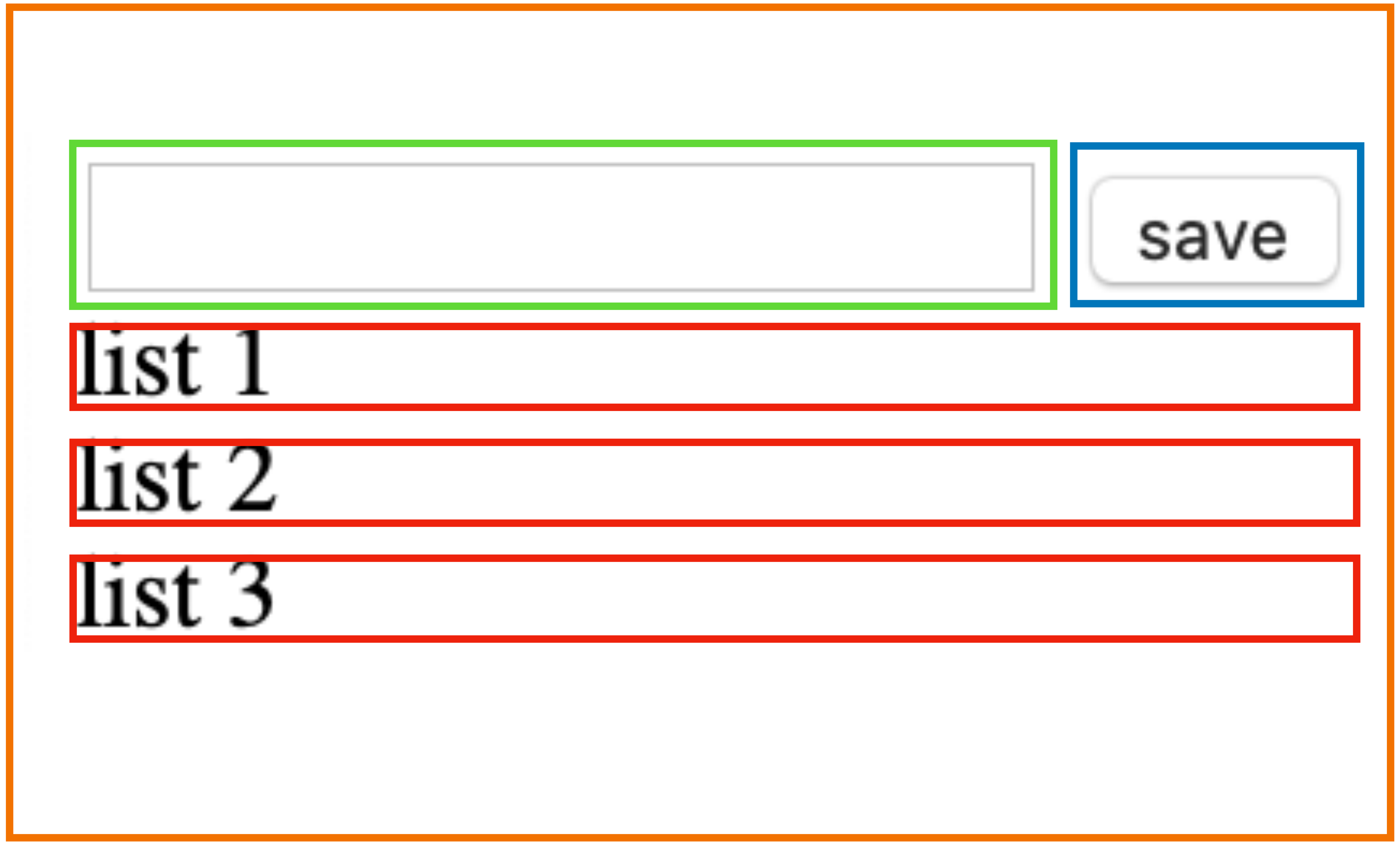


GOAL

HTML/CSS Refreshers



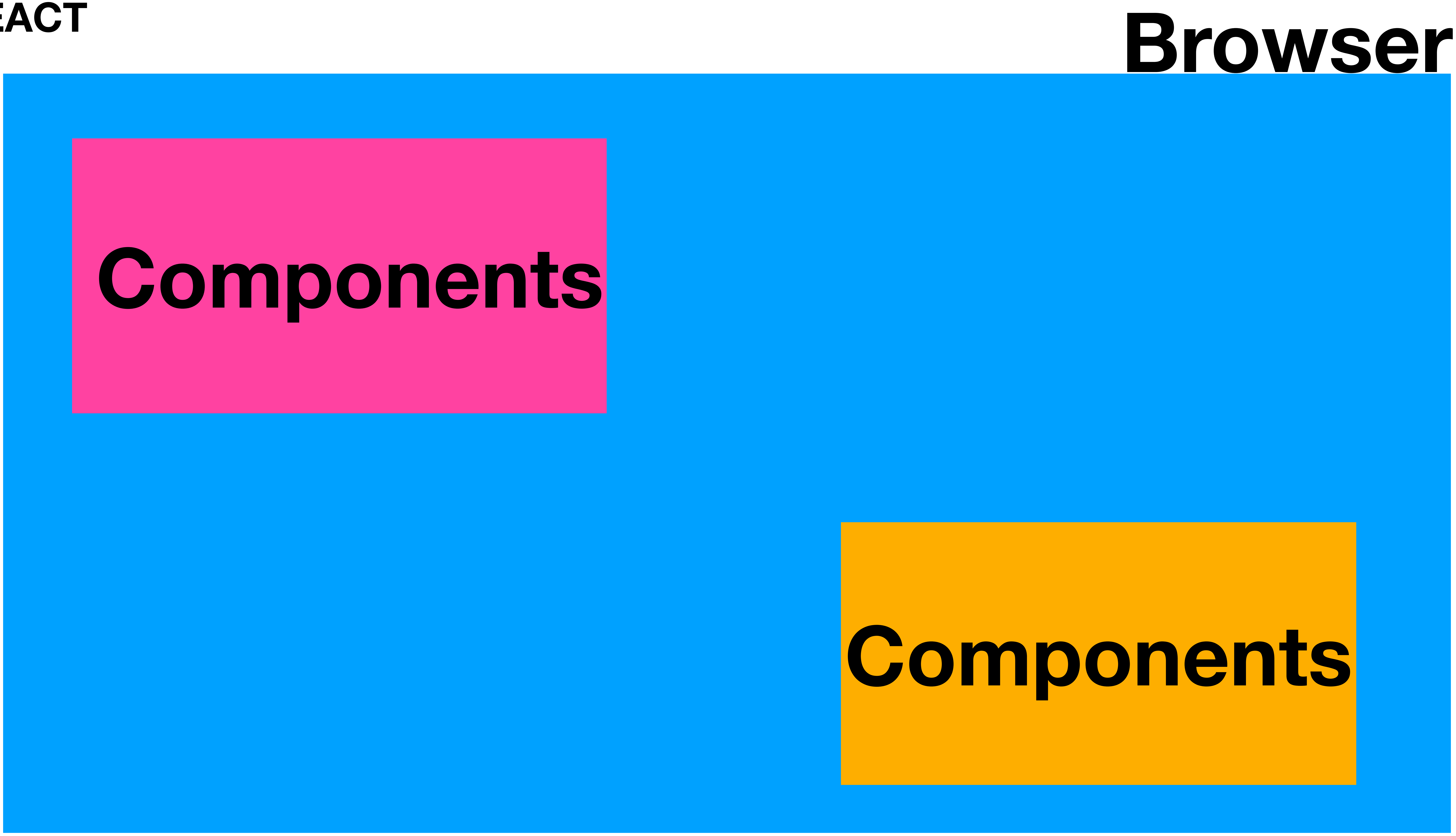
```
<div>
  <input/>
  <button> save </button>
   <div>
     list 1
   </div>
   <div>
     list 2
   </div>
   <div>
     list 3
   </div>
</div>
```

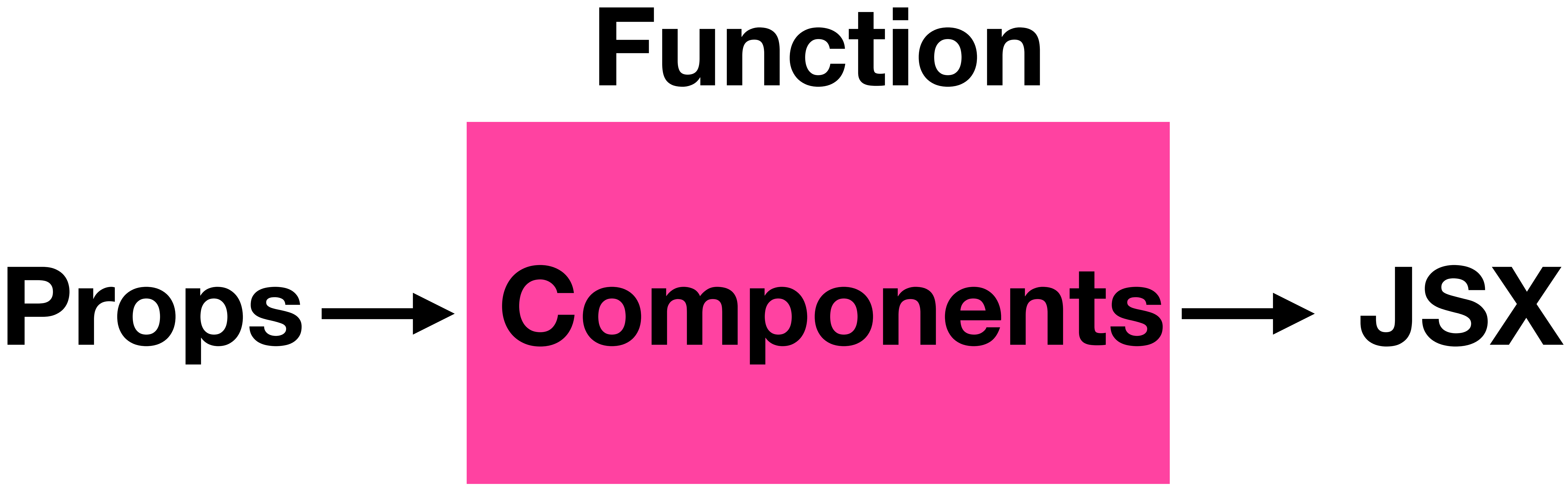


Checkpoint

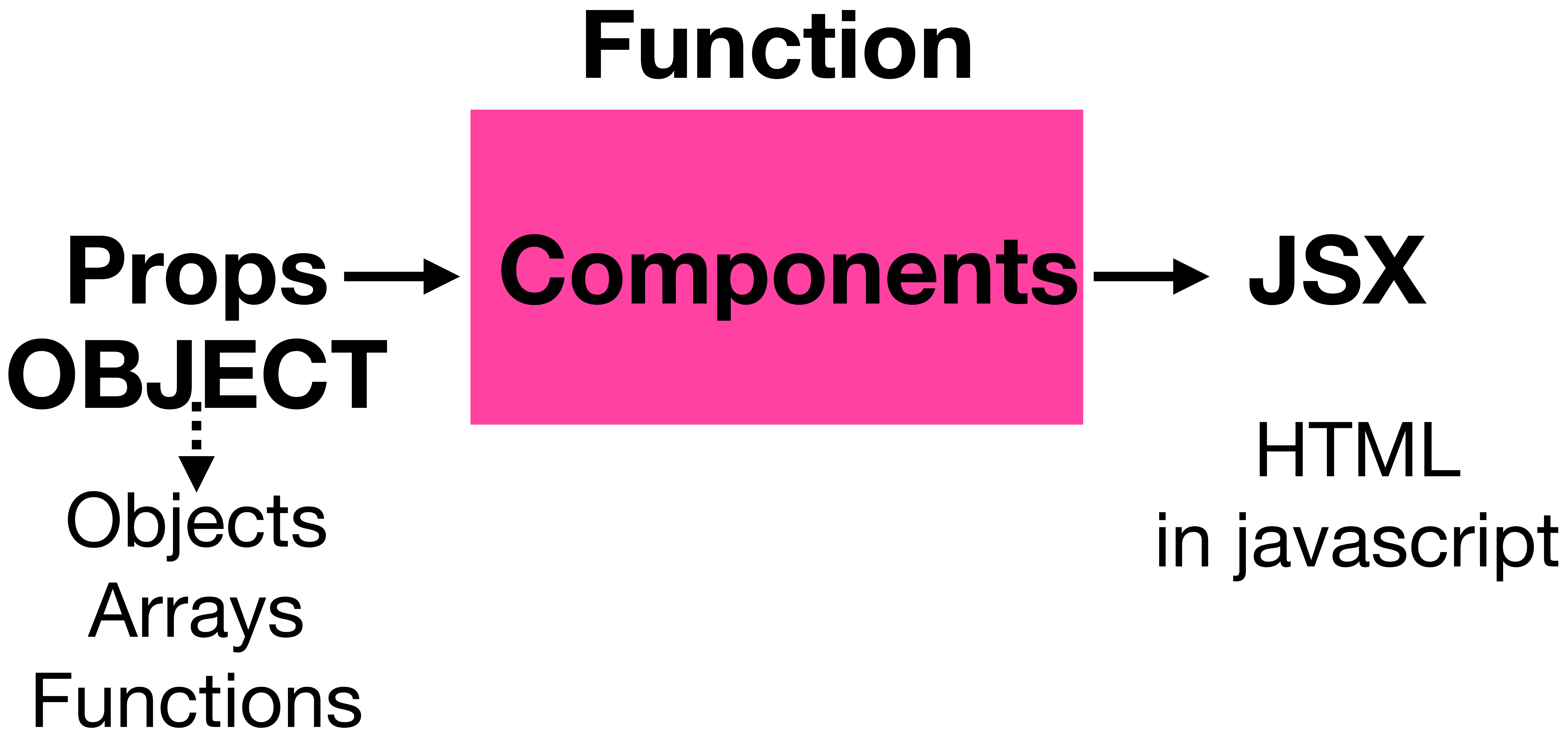
Questions?

REACT



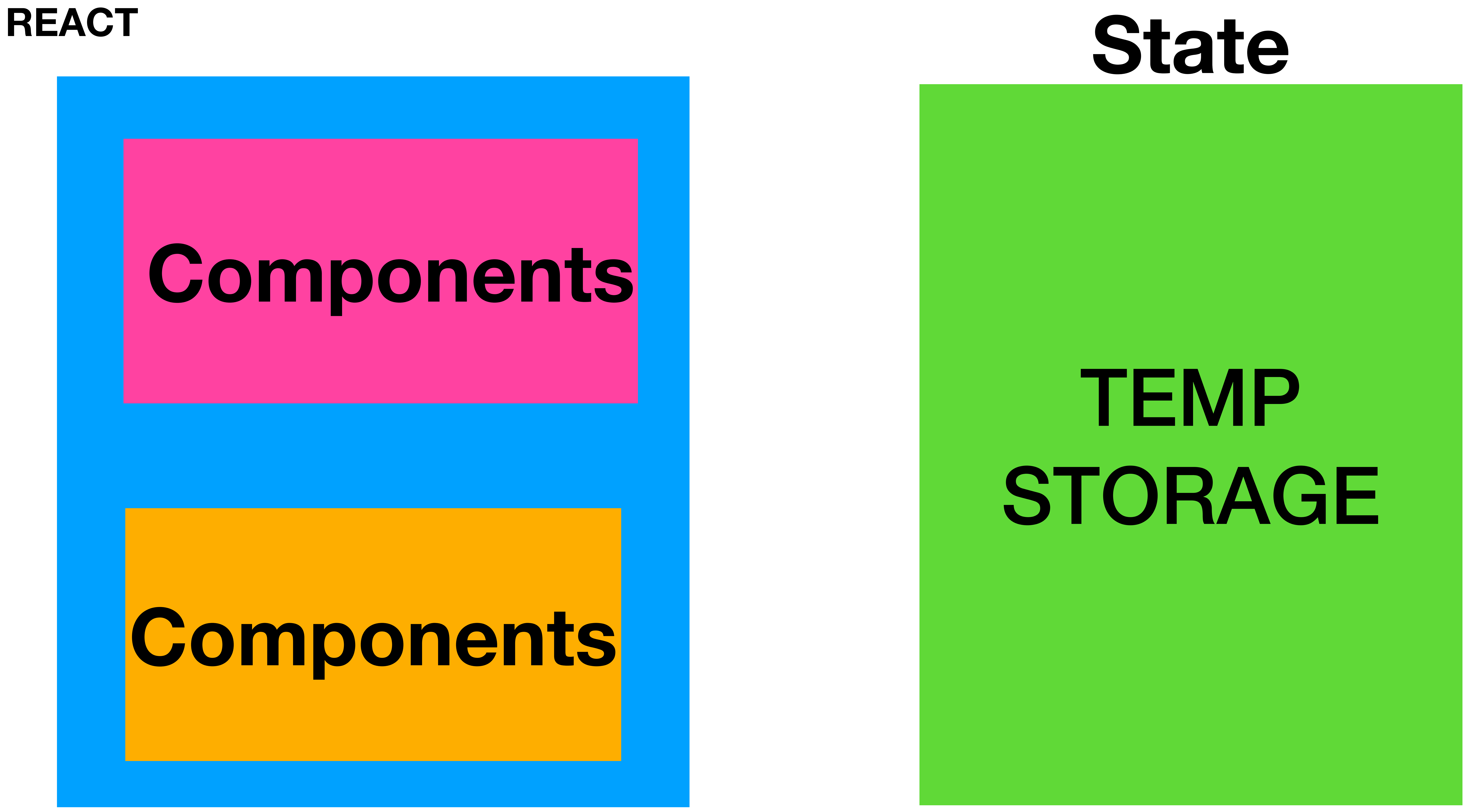

Function
Props
Components
JSX

REACT

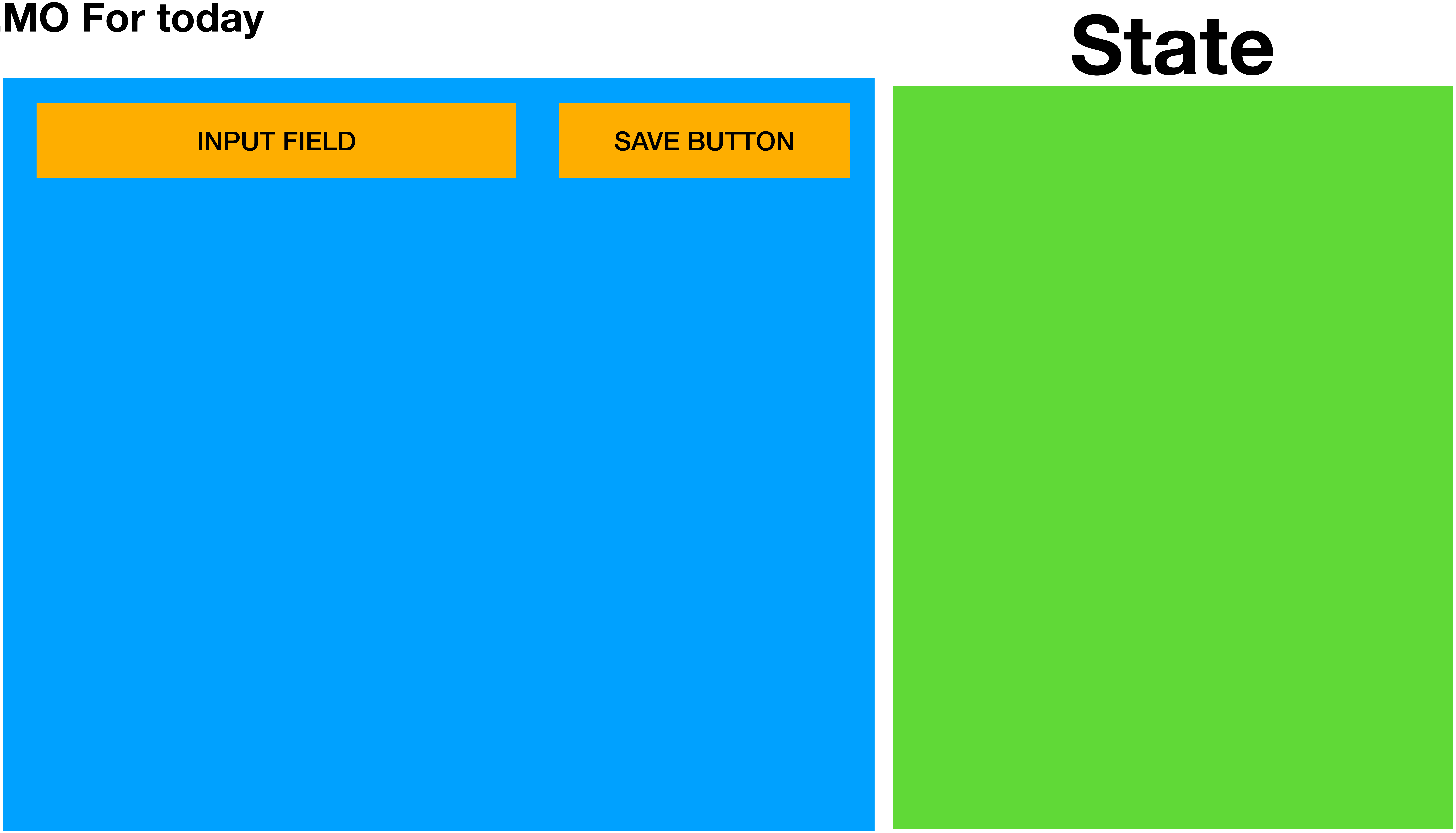

Function
Props
OBJECT
Objects
Arrays
Functions
Components
JSX
HTML
in javascript

Checkpoint

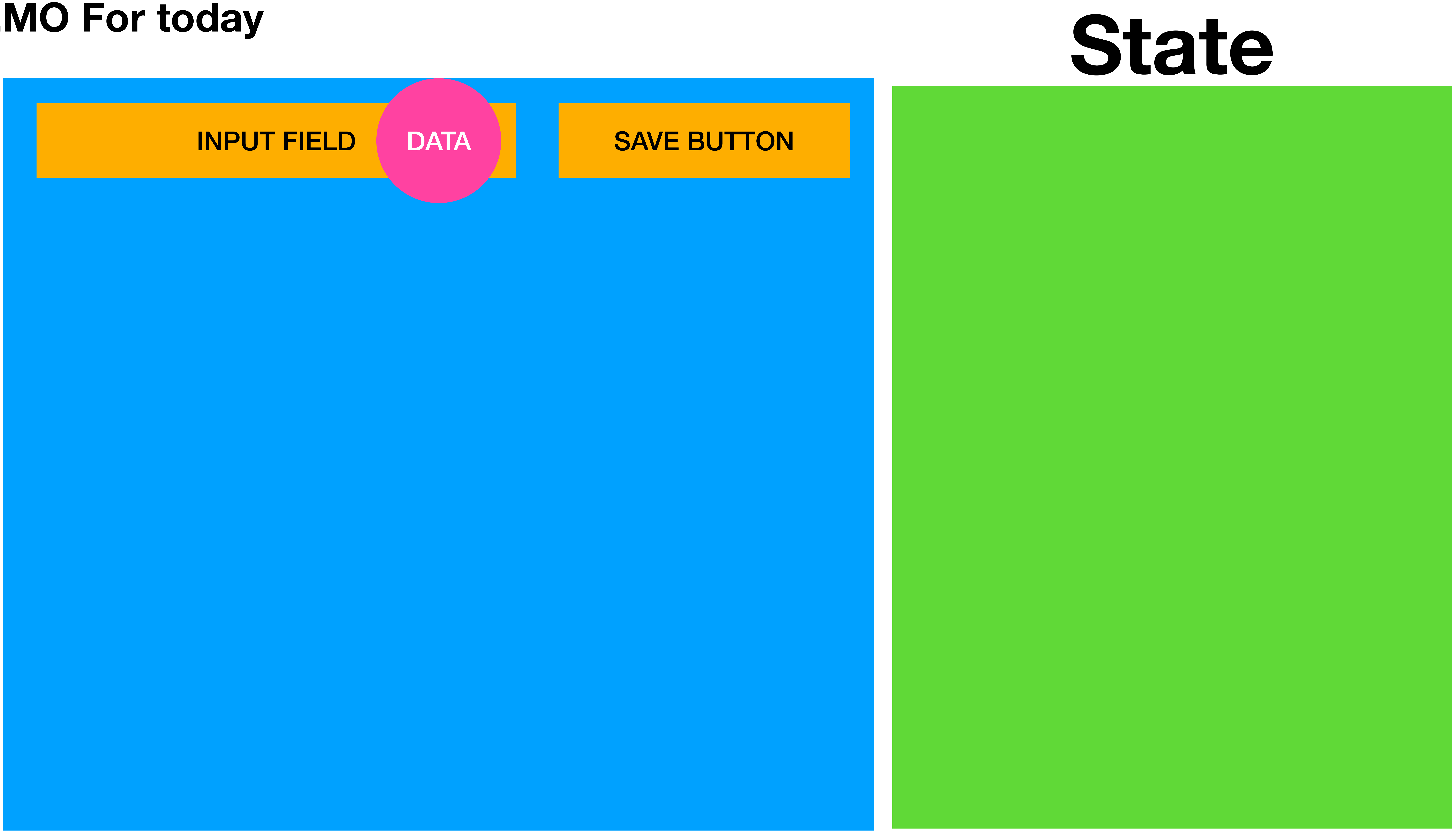
Questions?


REACT
State
Components
Components
TEMP STORAGE

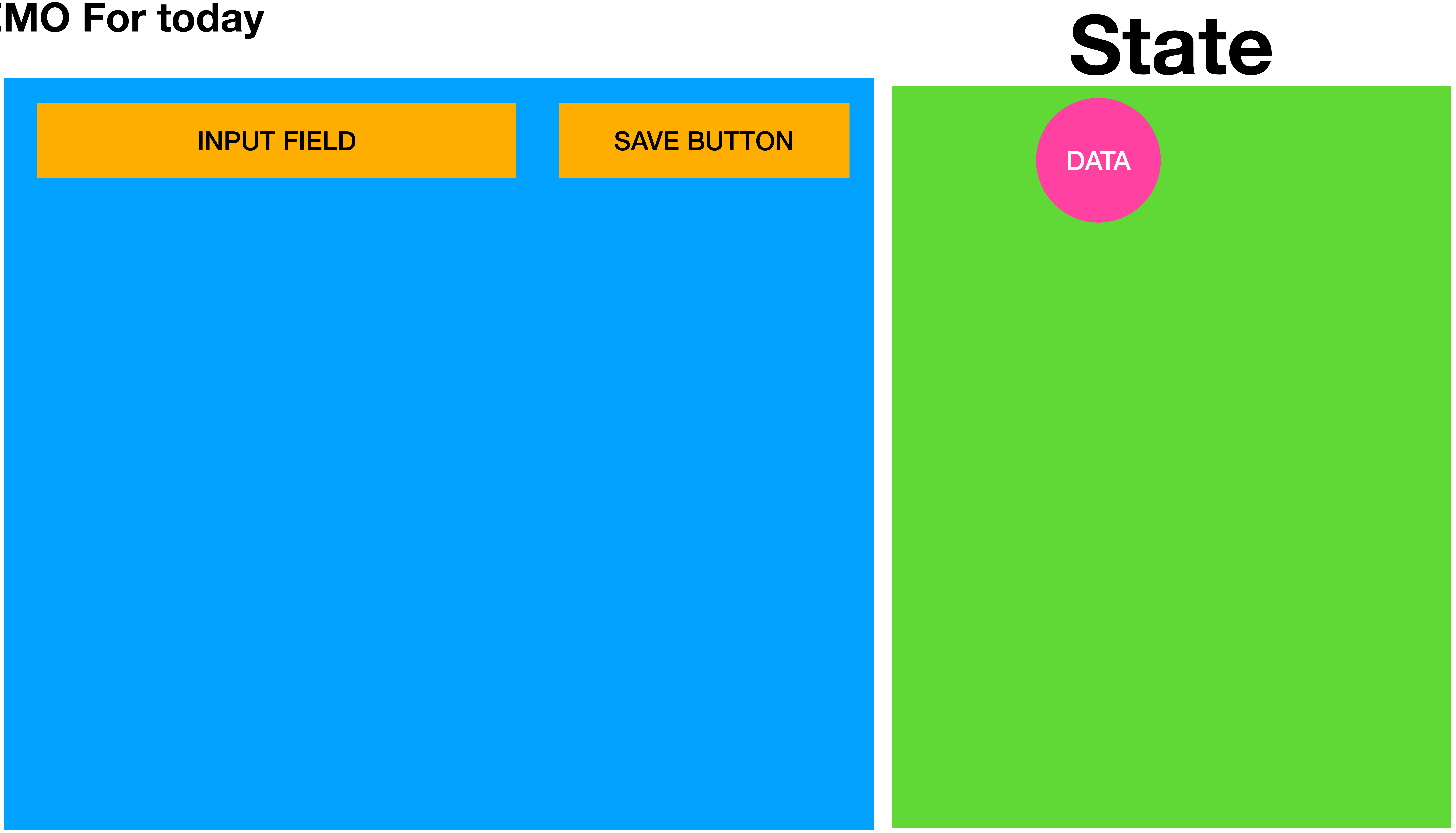
DEMO For today



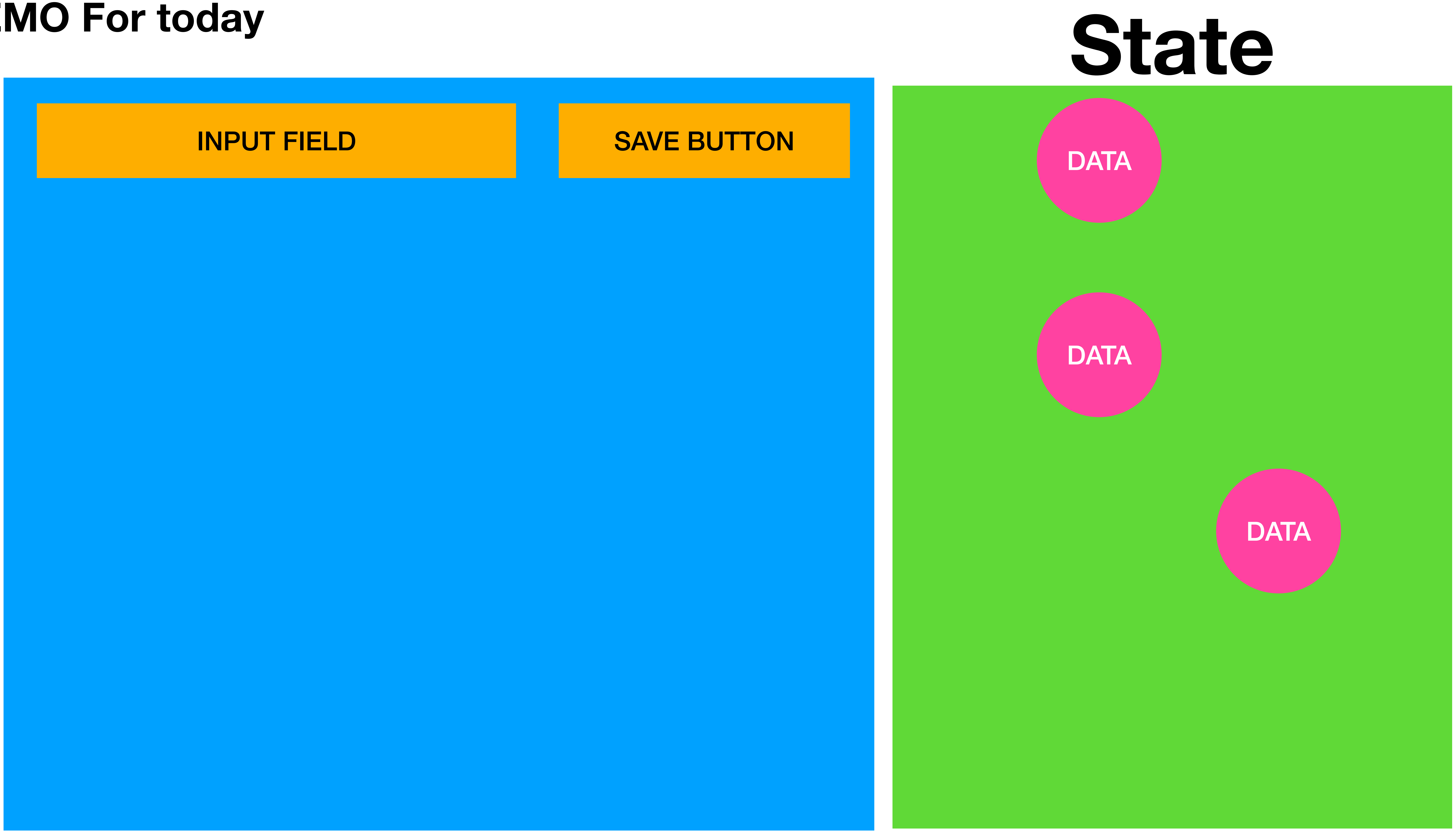
DEMO For today



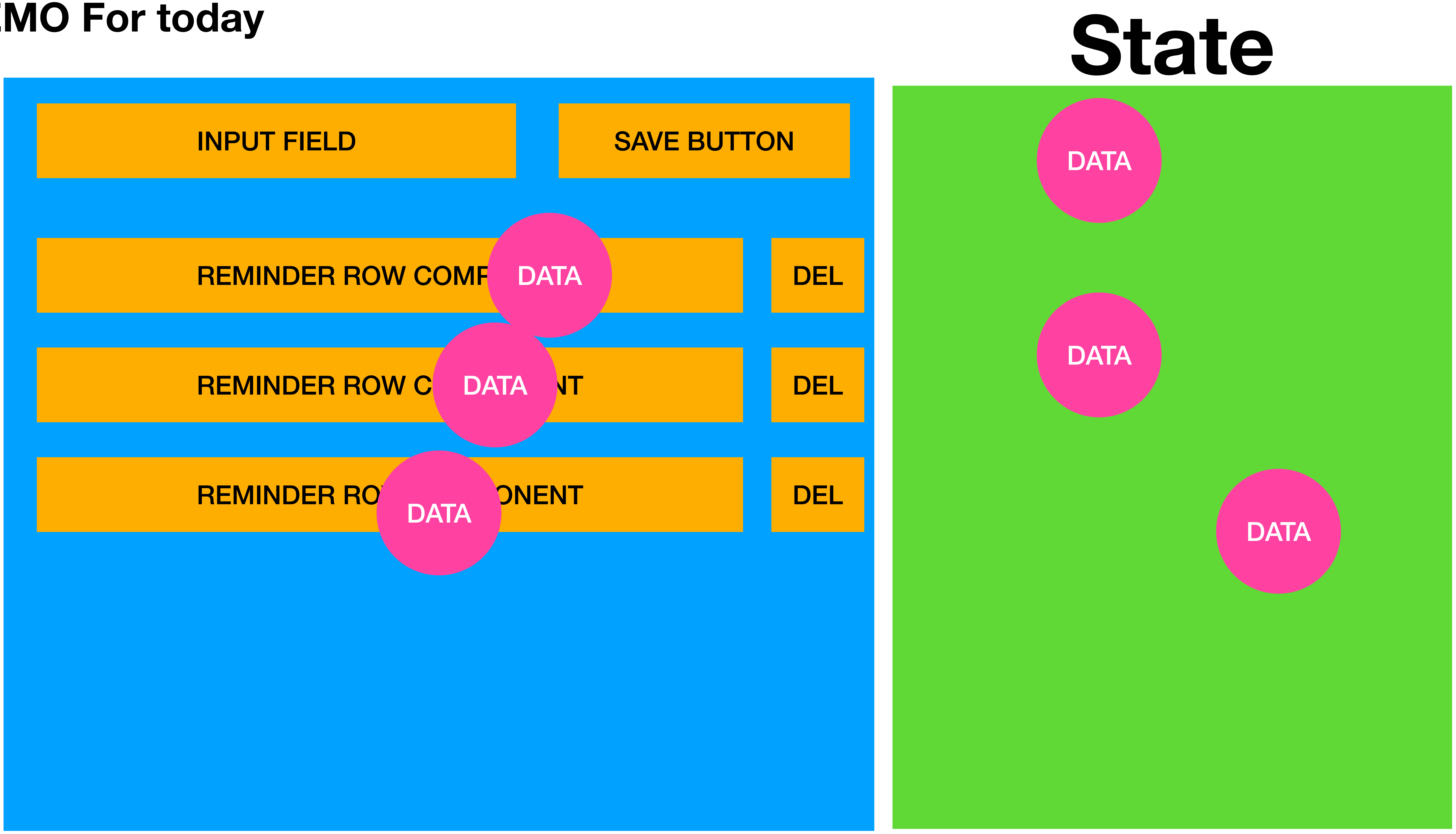
DEMO For today



DEMO For today

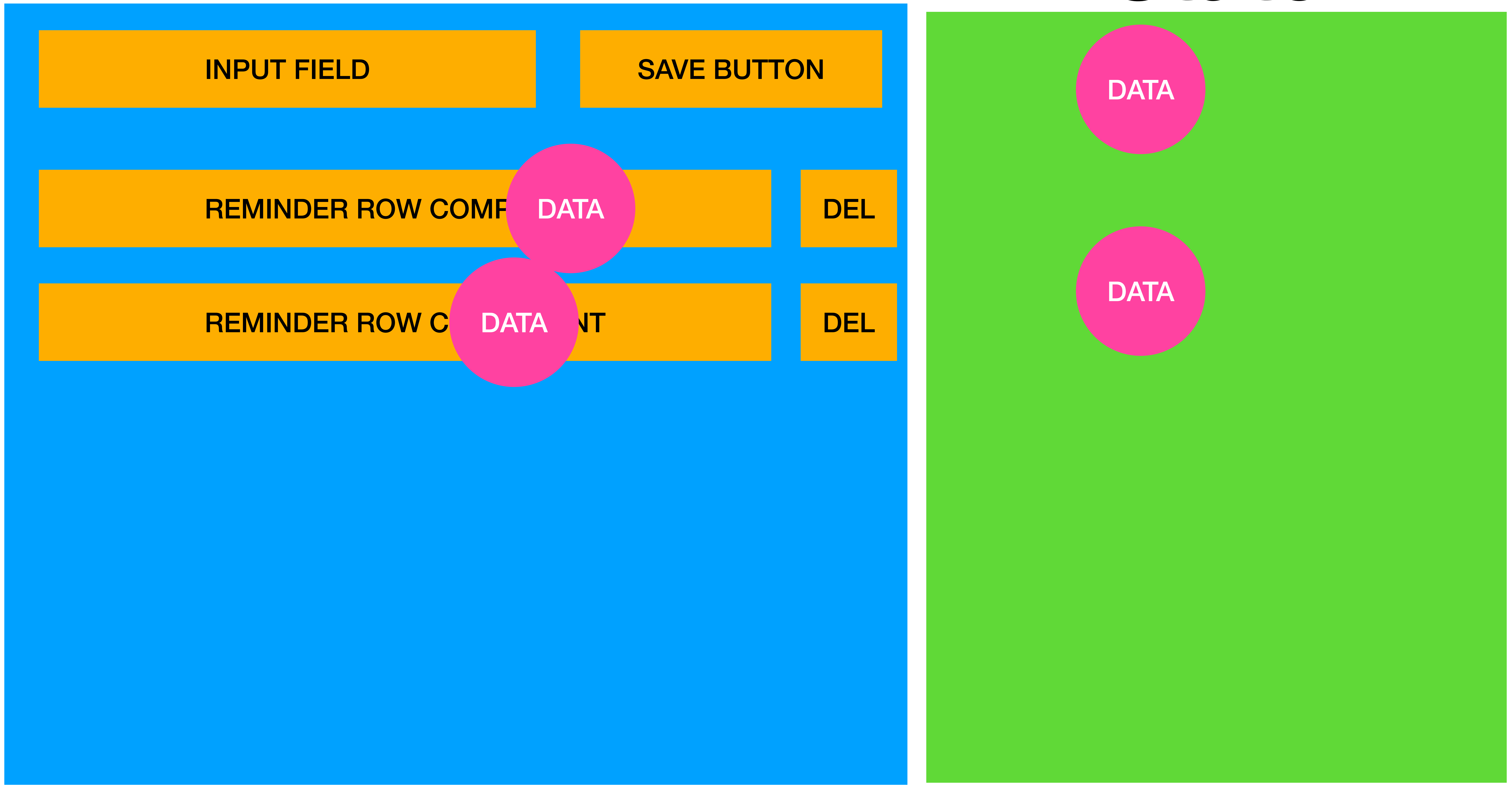


DEMO For today

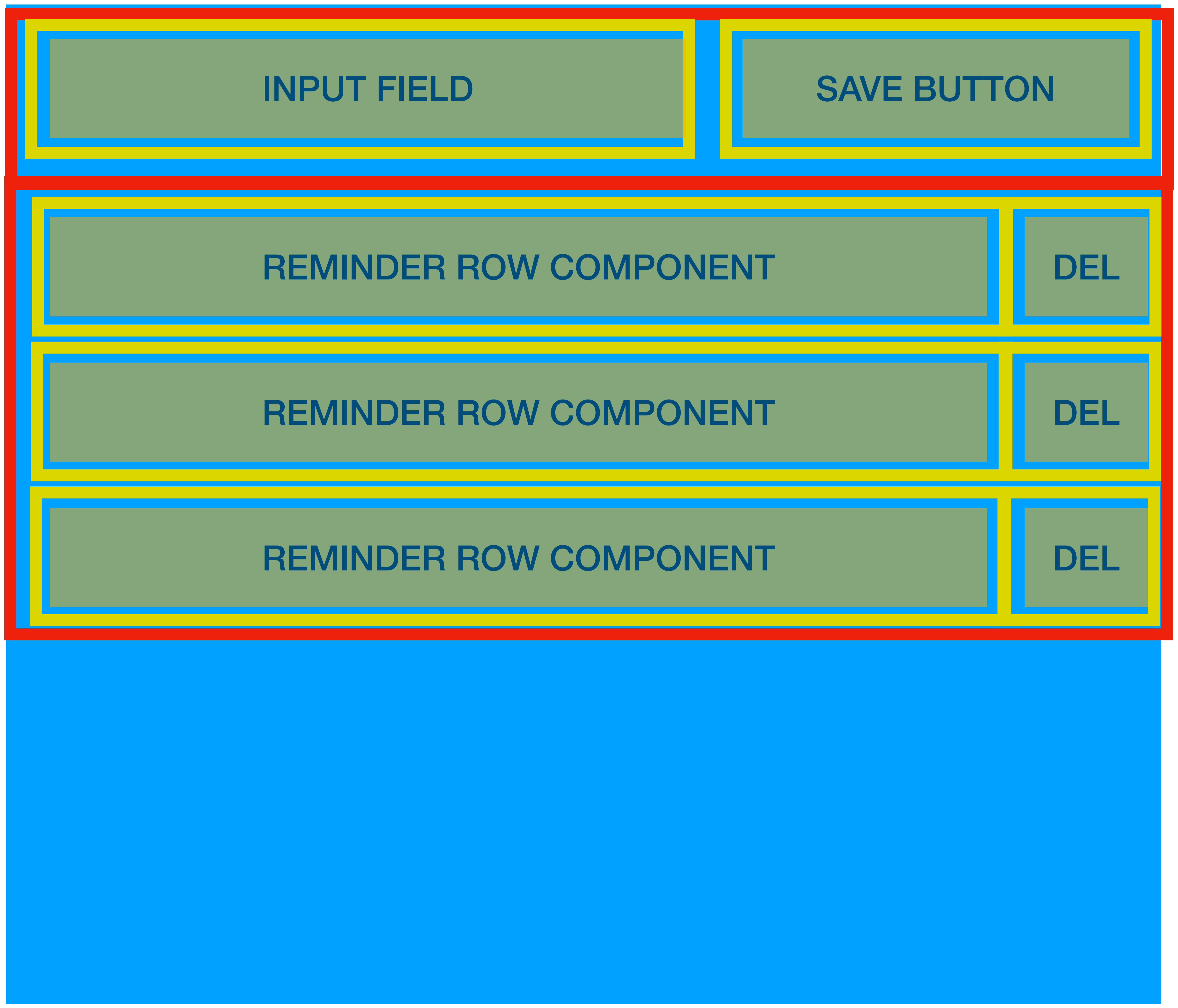


DEMO For today

State



DEMO For today



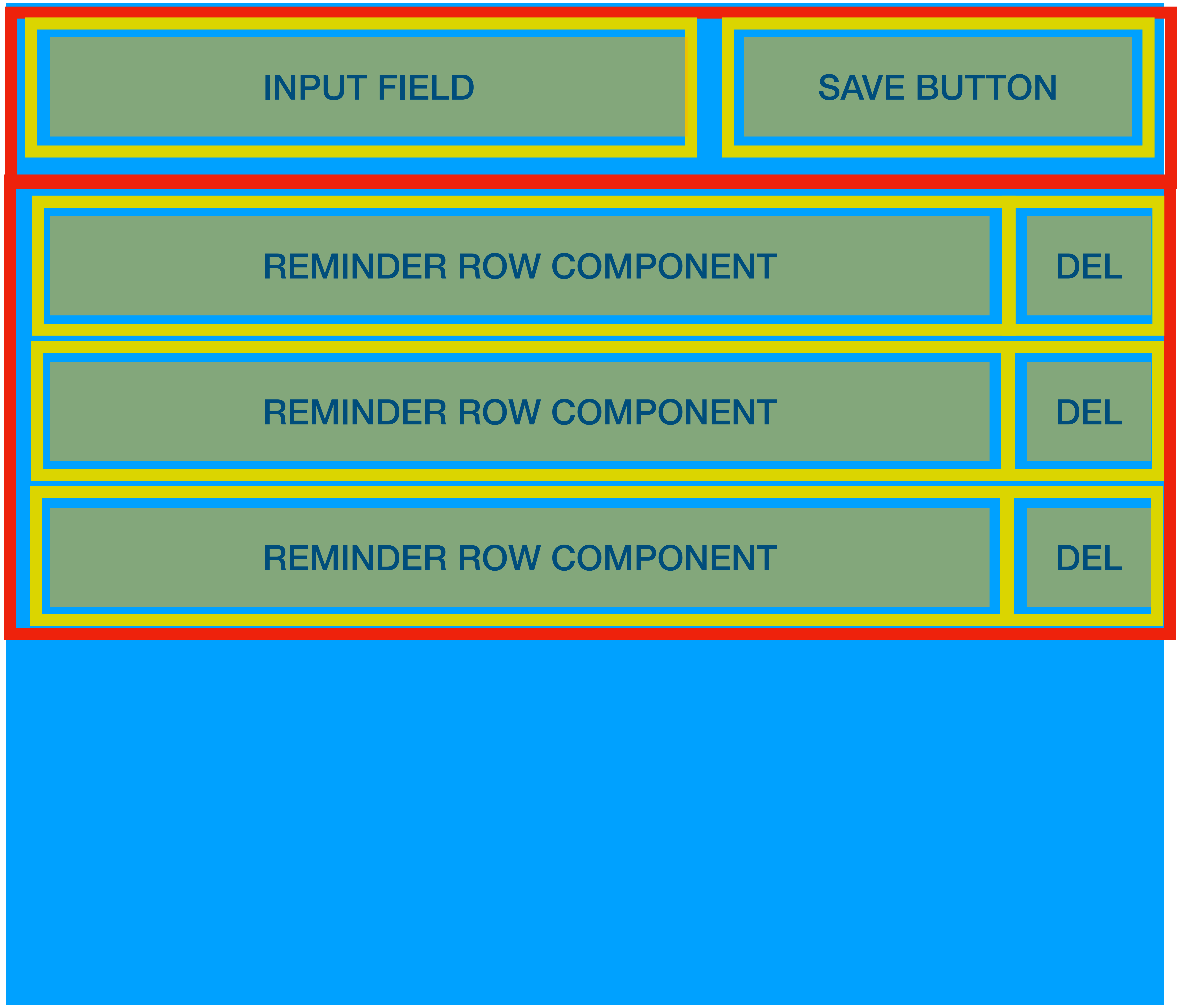
State

Checkpoint

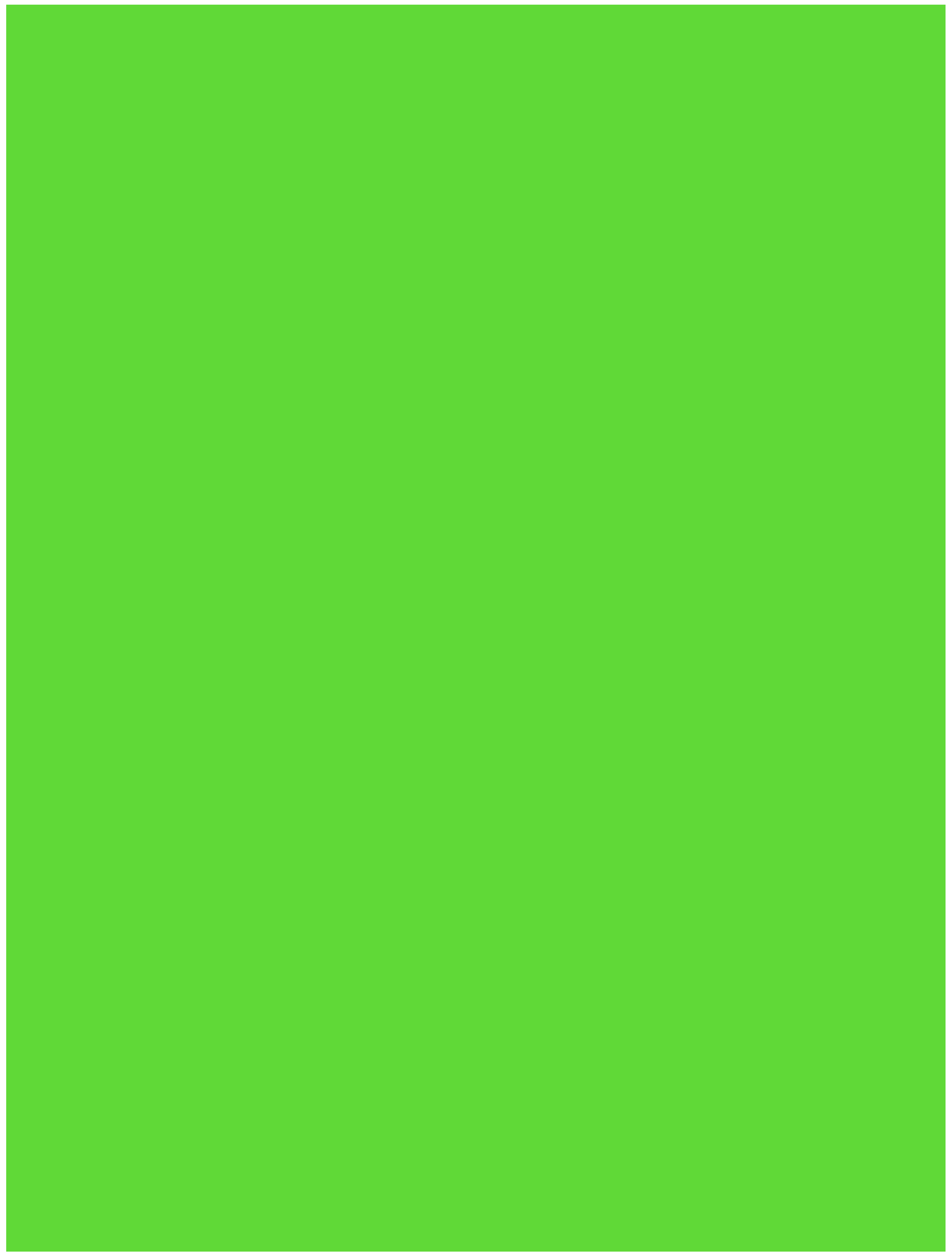
Questions?

DEMO For today

Flexbox

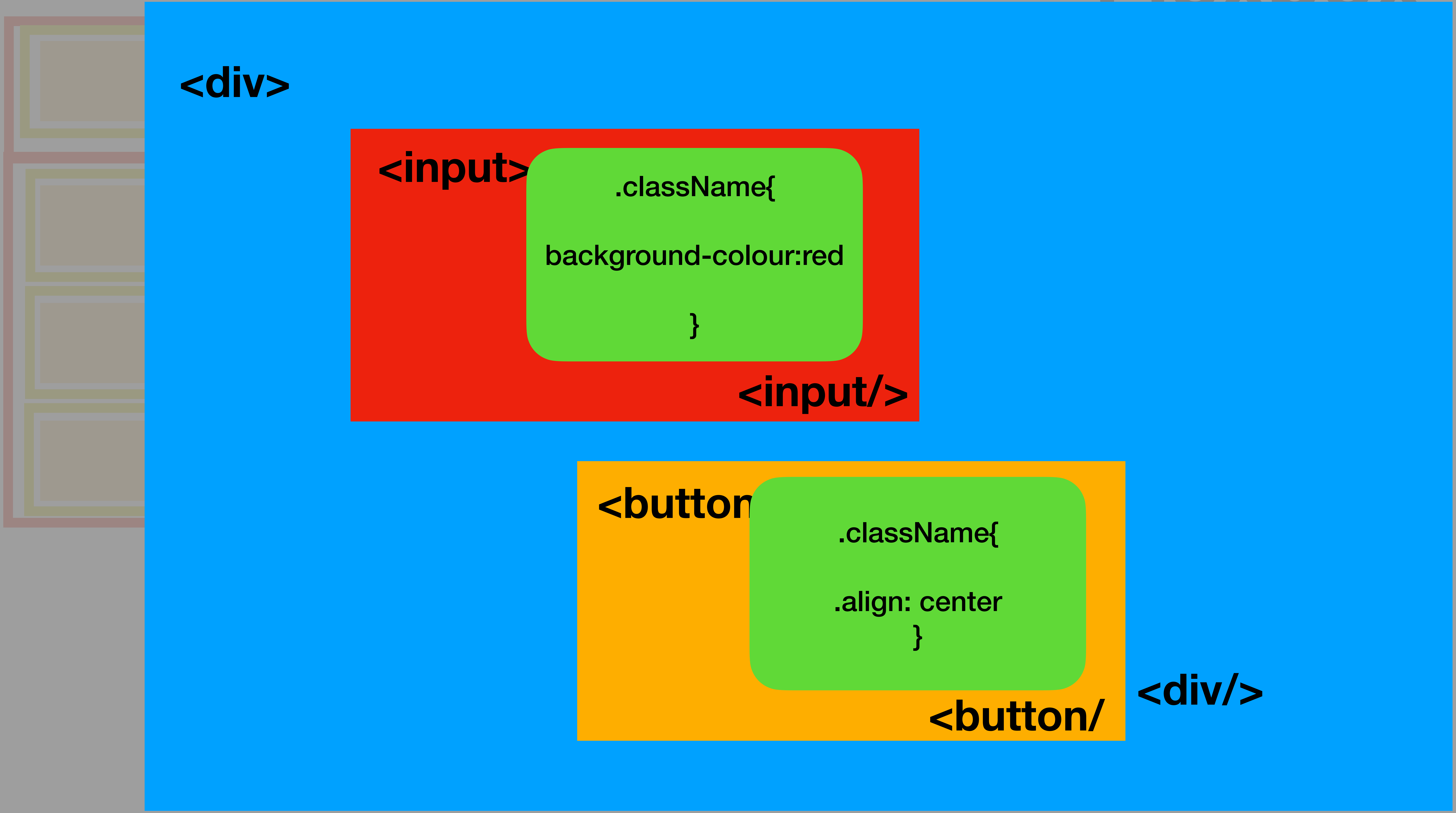


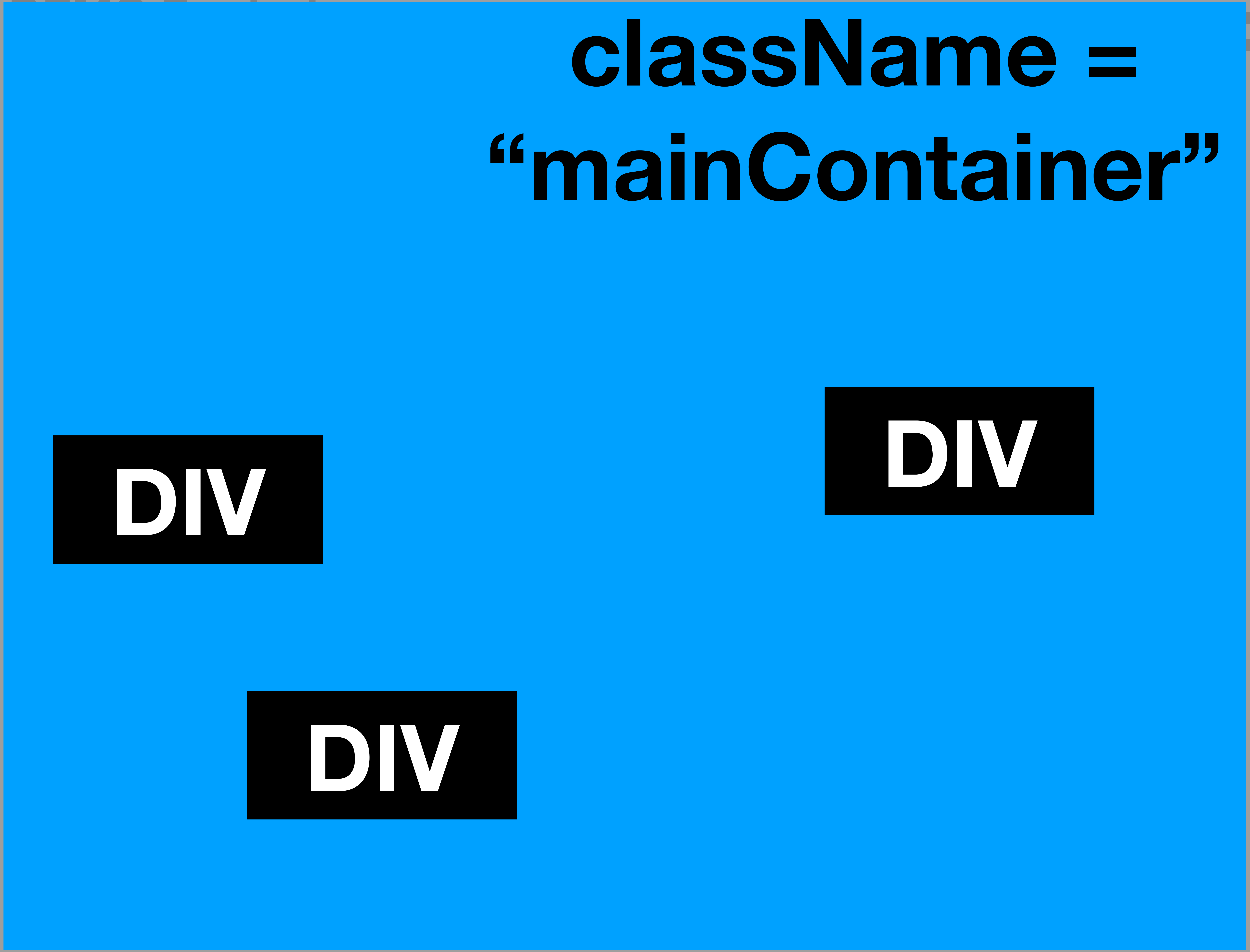
State



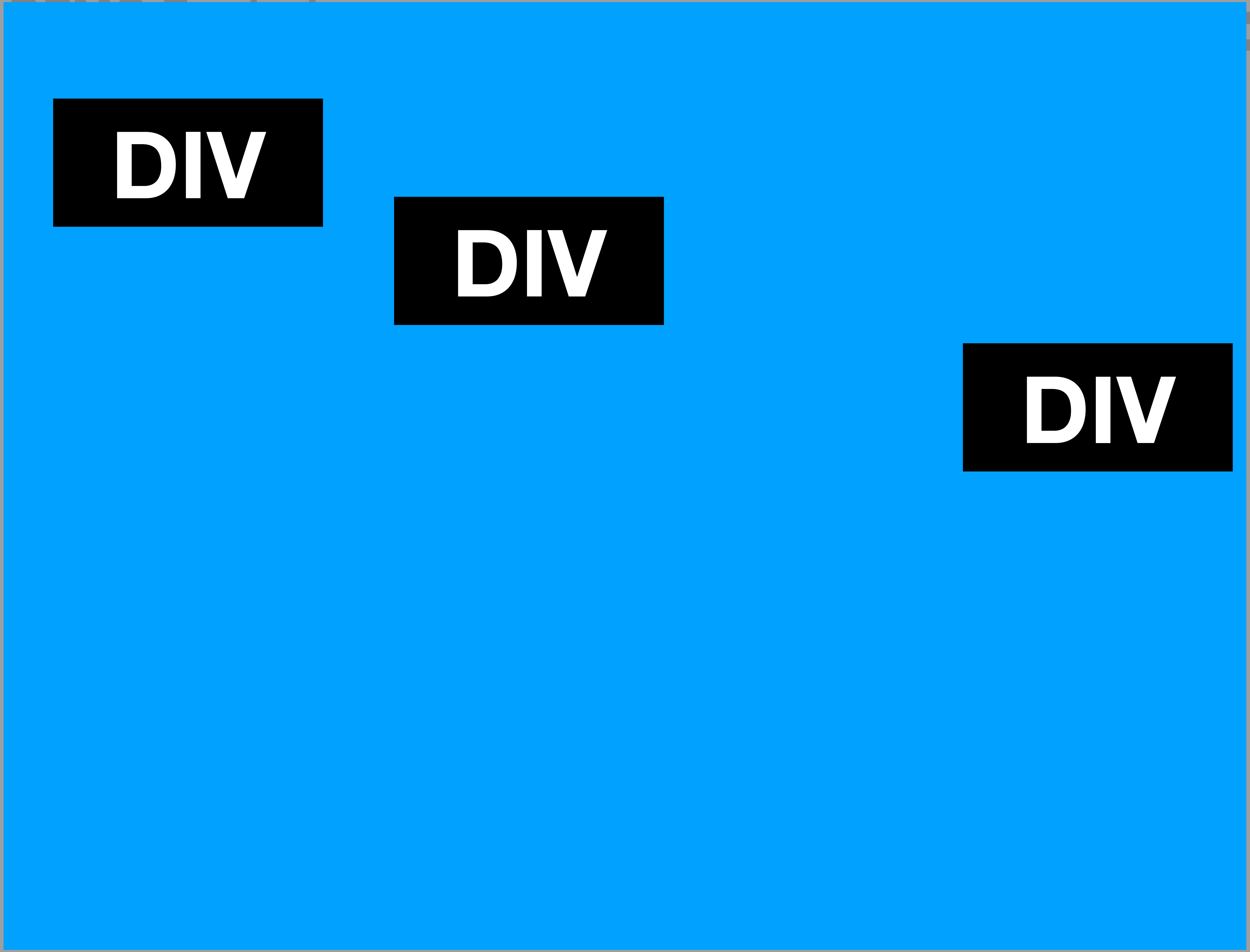
DEMO For today

Flexbox

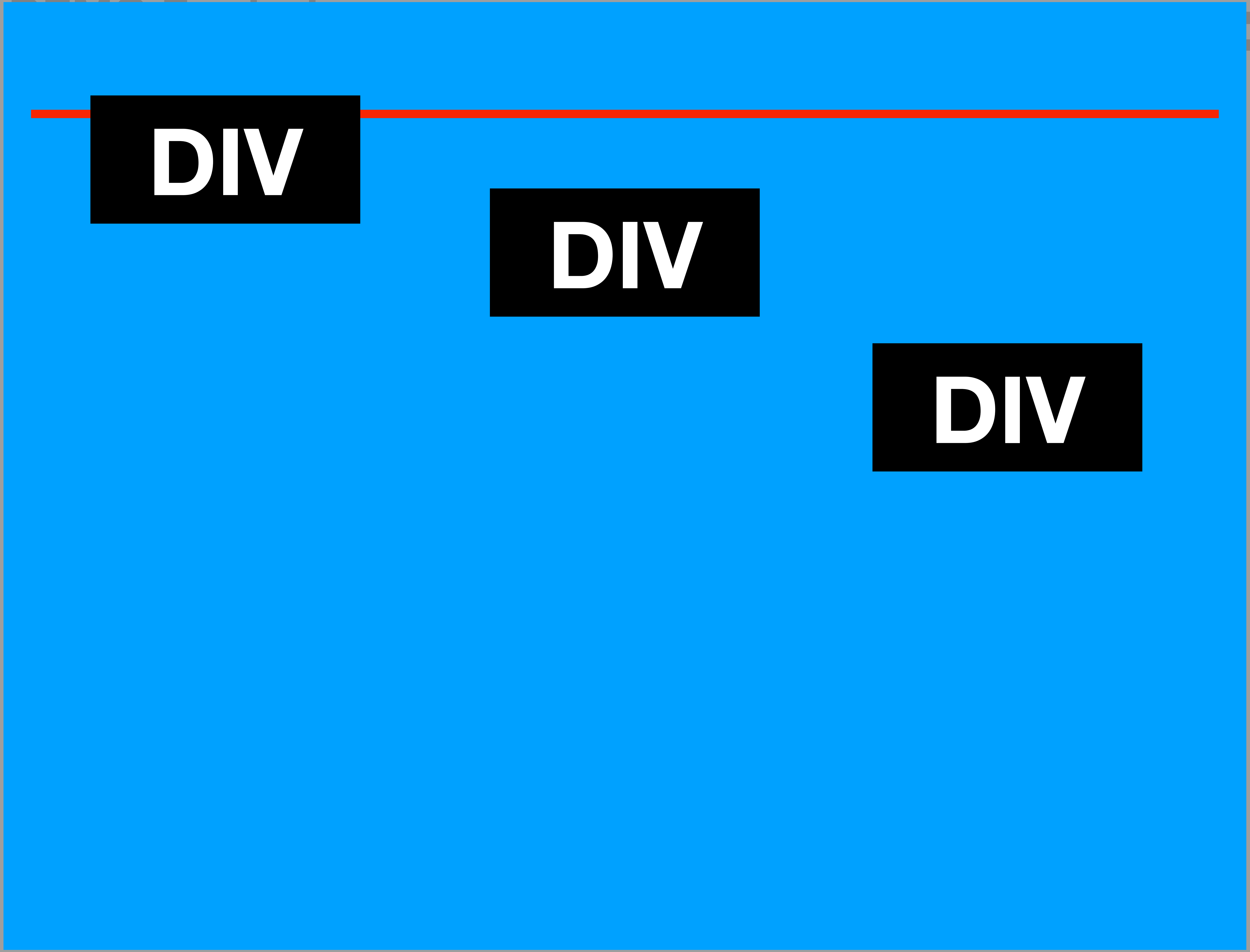




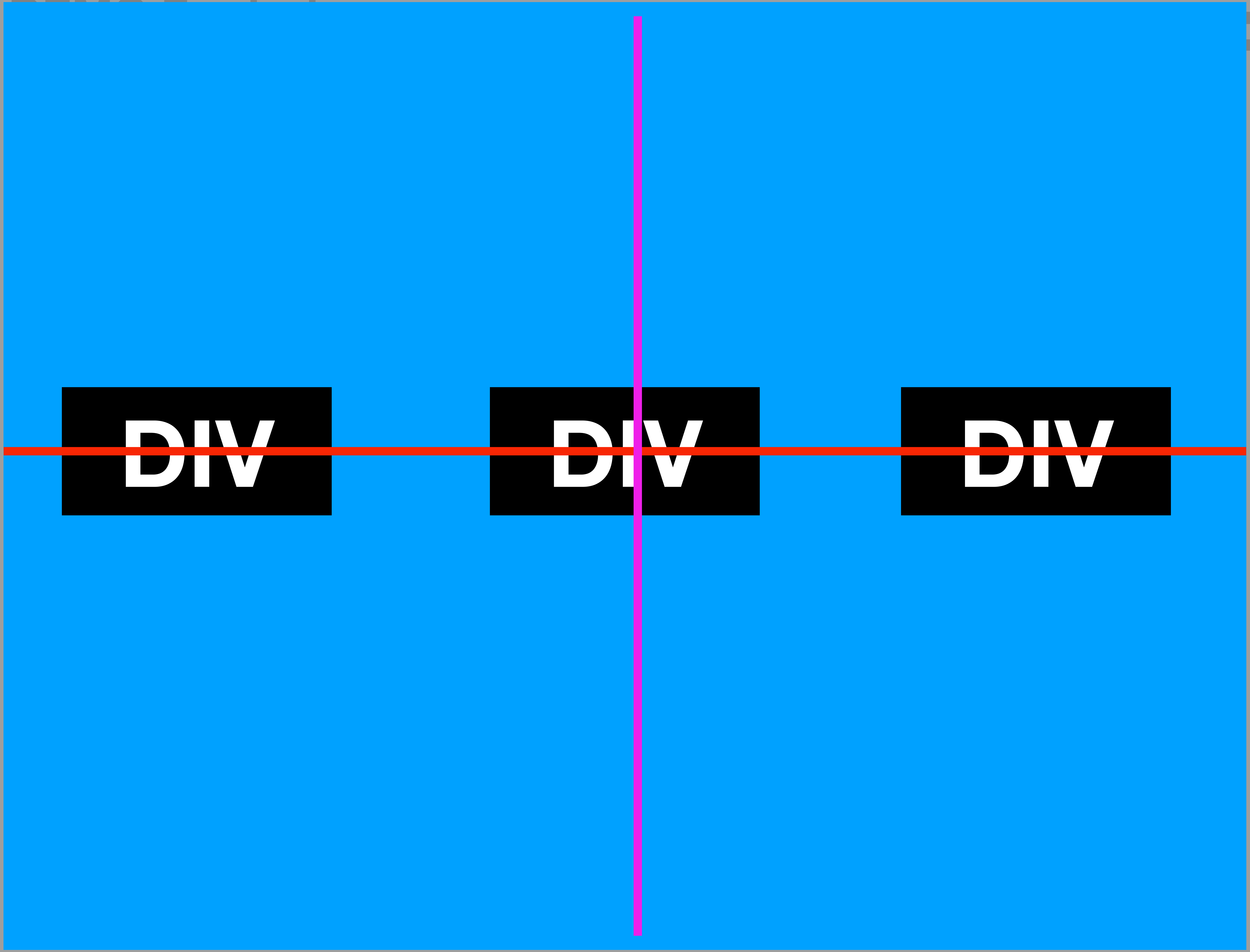
```
.mainContainer{

    display: flex;

}
```



```
.mainContainer{

    display: flex;
    flex-direction: row;

}
```



```
.mainContainer{

    display: flex;
    flex-direction: row;

    Justify-content:
    space-around;

    }
```



```
.mainContainer{

    display: flex;
    flex-direction: row;

    Justify-content:
    space-around;

    Align-content:
    Centre
    }
```

Checkpoint

Questions?

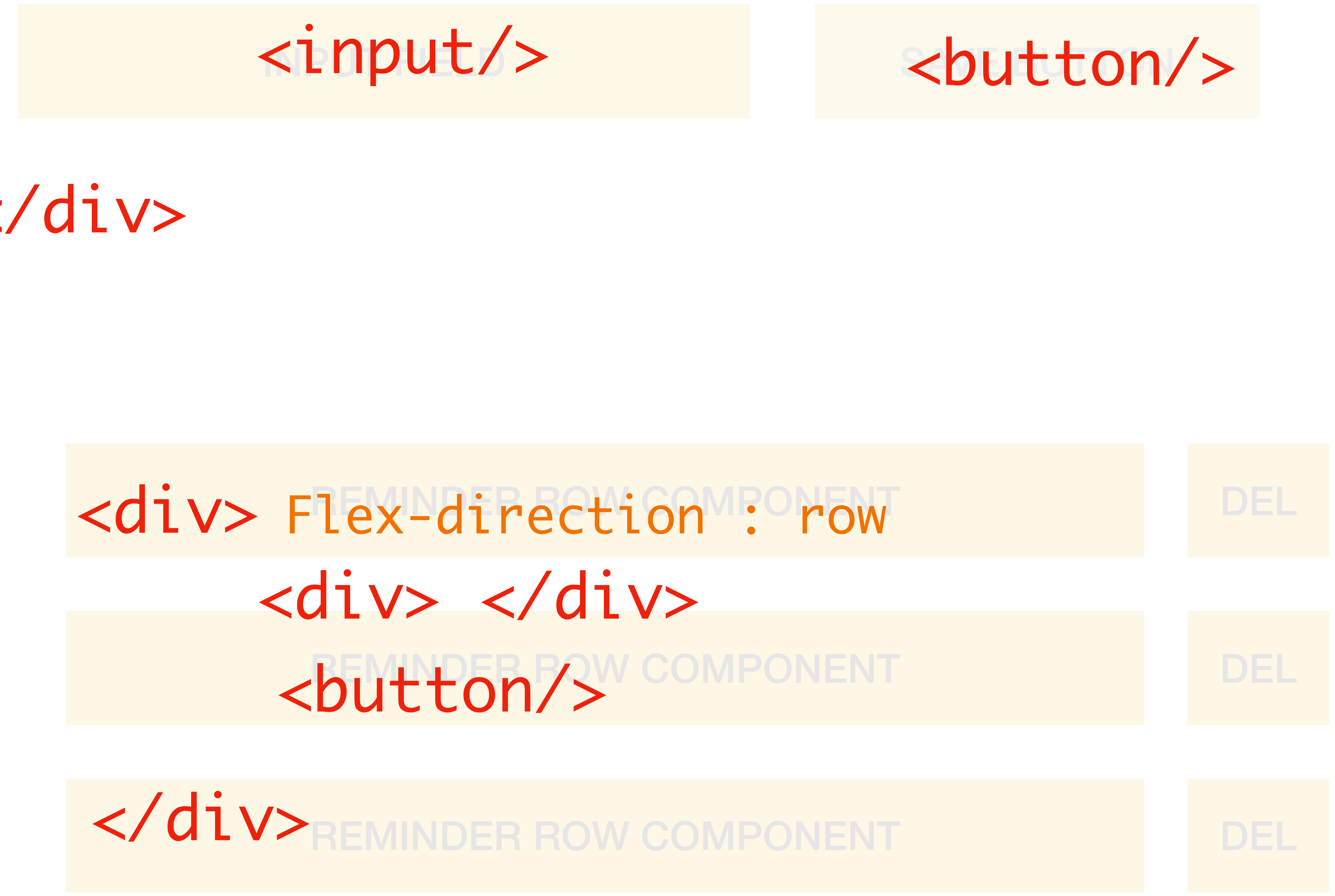
DEMO For today

```
<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



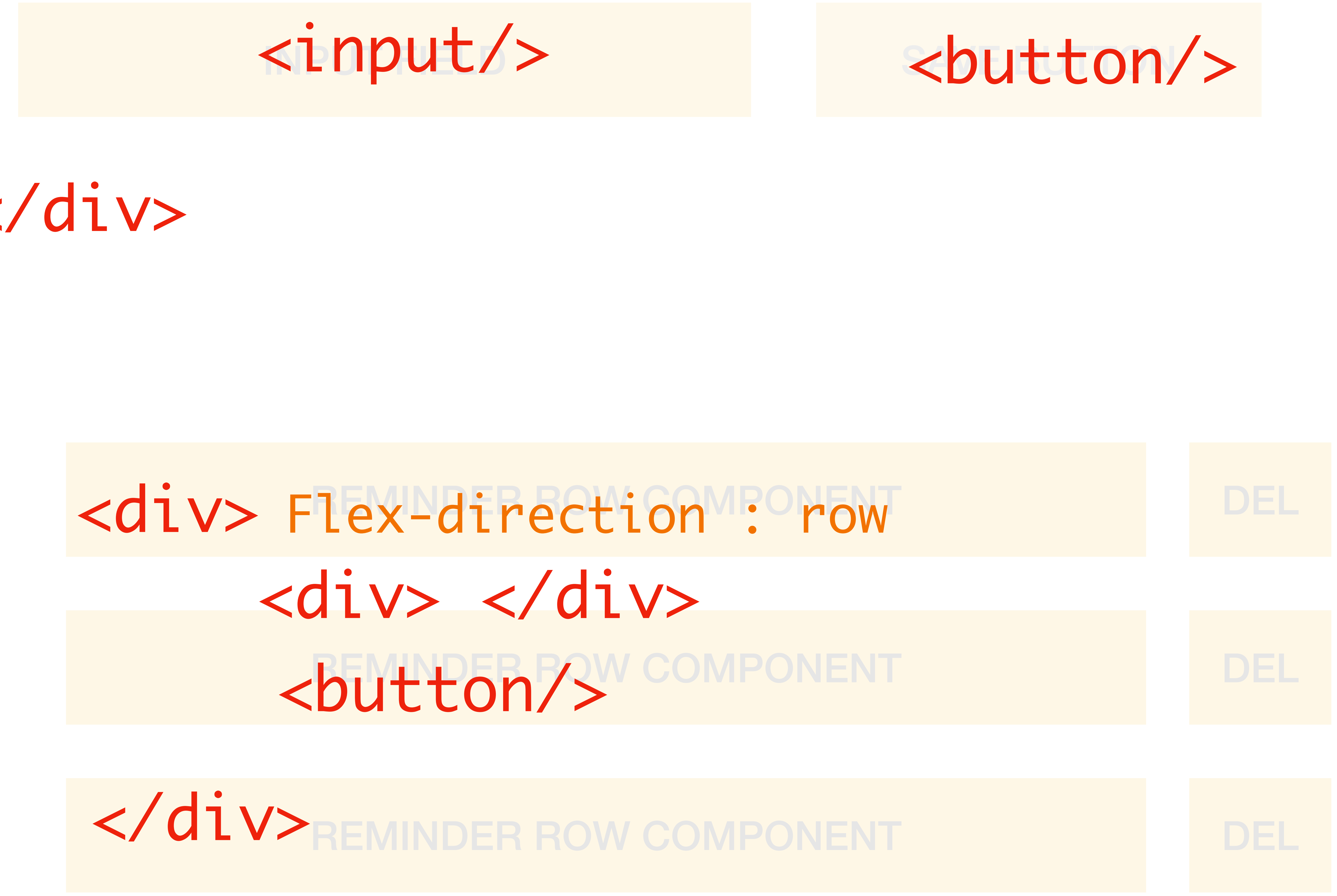
```
<div> Flex-direction : column
    <div> Flex-direction : row
          <input/>           <button/>

    </div>



      <div> Flex-direction : row
           <div> </div>
            <button/>

       </div>


</div>
```

Checkpoint

Questions?

DEMO For today

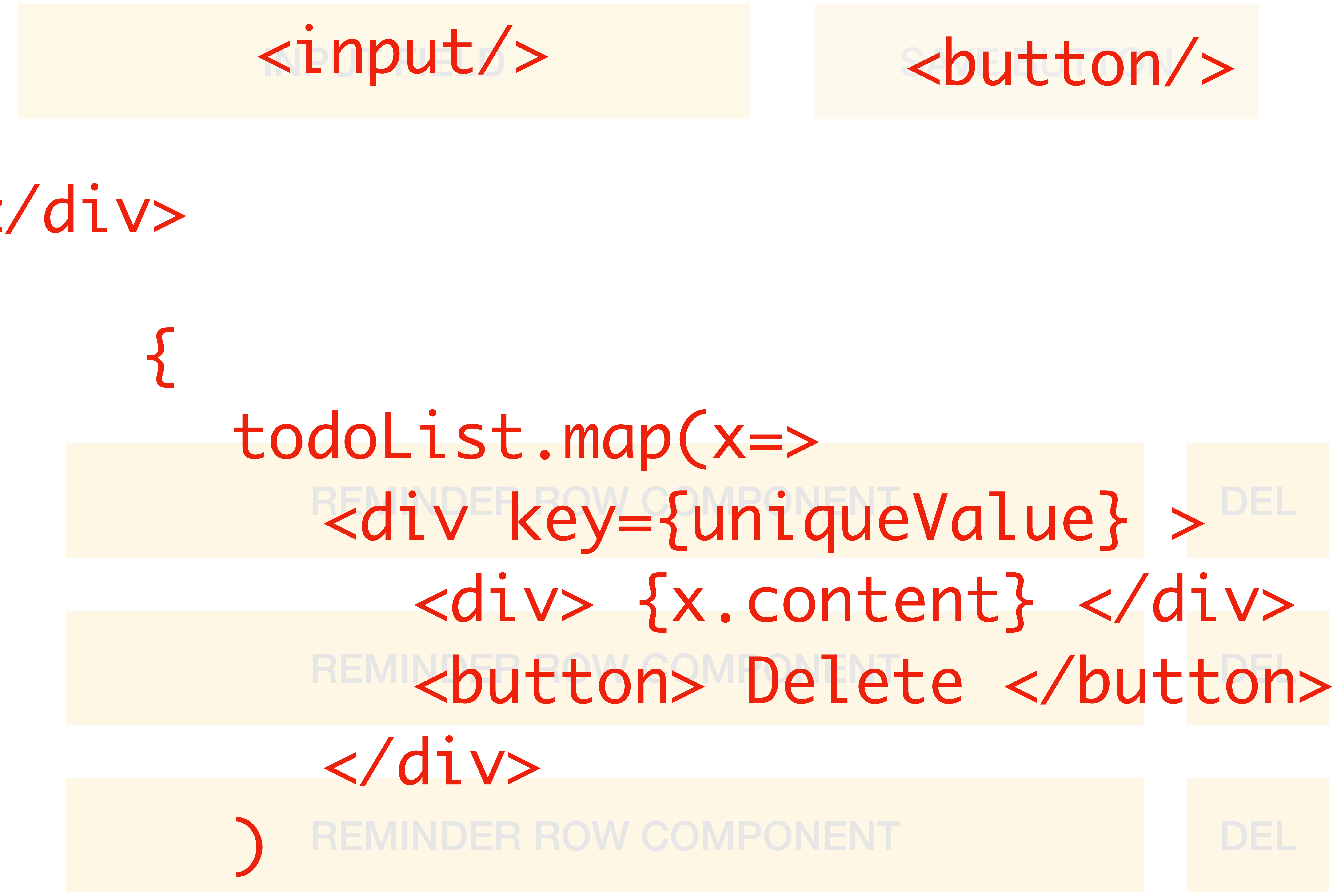
```
<div> Flex-direction : column
     <div> Flex-direction : row
            <input/>            <button/>

    </div>



       <div> Flex-direction : row
            <div> </div>
             <button/>

       </div>


</div>
```



State

```
inputValue=“”

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

DEMO For today



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

   </div>

        {
            todoList.map(x=>
               <div key={uniqueValue} >
                   <div> {x.content} </div>
                   <button> Delete </button>
               </div>
            )
        }
</div>
```

State

```
inputValue=“”

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

Checkpoint

Questions?

DEMO For today

```
inputValue=“”

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

Initialise state

```
[ inputValue, setInputValue ] = useState( “” )

[  todoList, setTodoList ] = useState( [ ] )
```

DEMO For today

```
inputValue=“”

todoList = [ ]
```

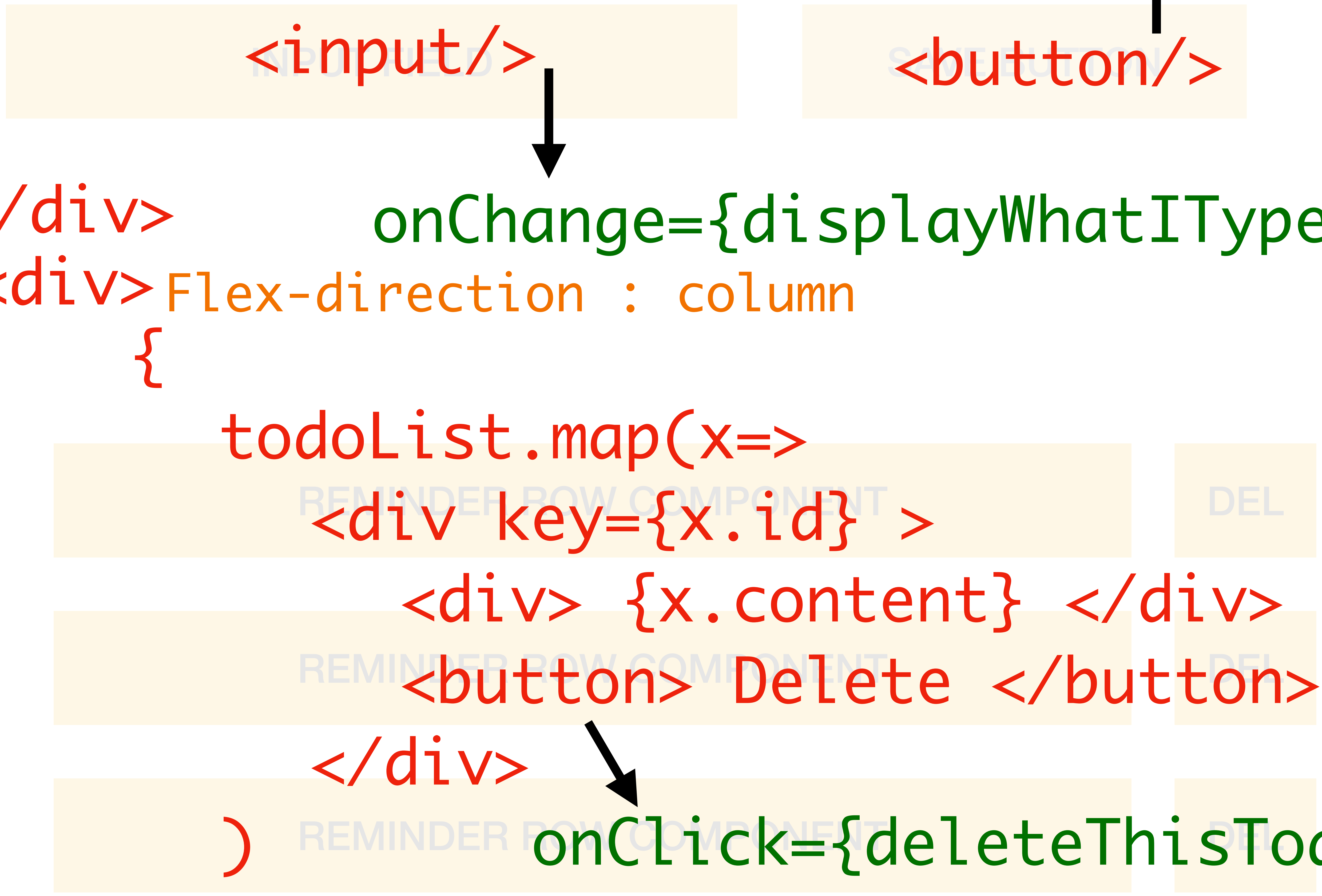
Initialise state

```
[ inputValue, setInputValue ] = useState( “” )

[  todoList, setTodoList ] = useState( [ ] )
```

Checkpoint

Questions?

DEMO For today



```
<div> Flex-direction : column          onClick={saveWhatIType}
    <div> Flex-direction : row
            <input/>                        <button/>

    </div>      onChange={displayWhatIType}
    <div>Flex-direction : column
        {
            todoList.map(x=>
                <div key={x.id} >
                    <div> {x.content} </div>
                    <button> Delete </button>
                </div>
            )          onClick={deleteThisTodo}
        }
    </div>
</div>
```

State

```
inputValue=“”

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

DEMO For today

```
[ inputValue, setInputValue ] = useState( “” )

[  todoList, setTodoList ] = useState( [ ] )
```

```
onChange={displayWhatIType}

onClick={saveWhatIType}

onClick={deleteThisTodo}
```

DEMO For today

```
onChange={displayWhatIType} setInputValue

onClick={saveWhatIType} setTodoList

onClick={deleteThisTodo}
```

DEMO For today

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

```
onClick={saveWhatIType}
onClick={deleteThisTodo}
```

DEMO For today

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

```
onChange={displayWhatIType}
onClick={deleteThisTodo}
```

DEMO For today

```
onClick={deleteThisTodo}
```

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

```
onClick={saveWhatIType}
onChange={displayWhatIType}
```

DEMO For today

```
<div> Flex-direction : column          onClick={saveWhatIType}
    <div> Flex-direction : row
        <input/>                        <button/>

   </div>          onChange={displayWhatIType}
   <div>Flex-direction : column
       {
          todoList.map(x=>
            <div key={x.id} >
               <div> {x.content} </div>
               <button> Delete </button>
            </div>
          ) onClick={ ()=>deleteThisTodo(id) }
       }
   </div>
</div>
```

INPUT FIELD SAVE BUTTON

REMINDER ROW COMPONENT DEL

REMINDER ROW COMPONENT DEL

REMINDER ROW COMPONENT DEL

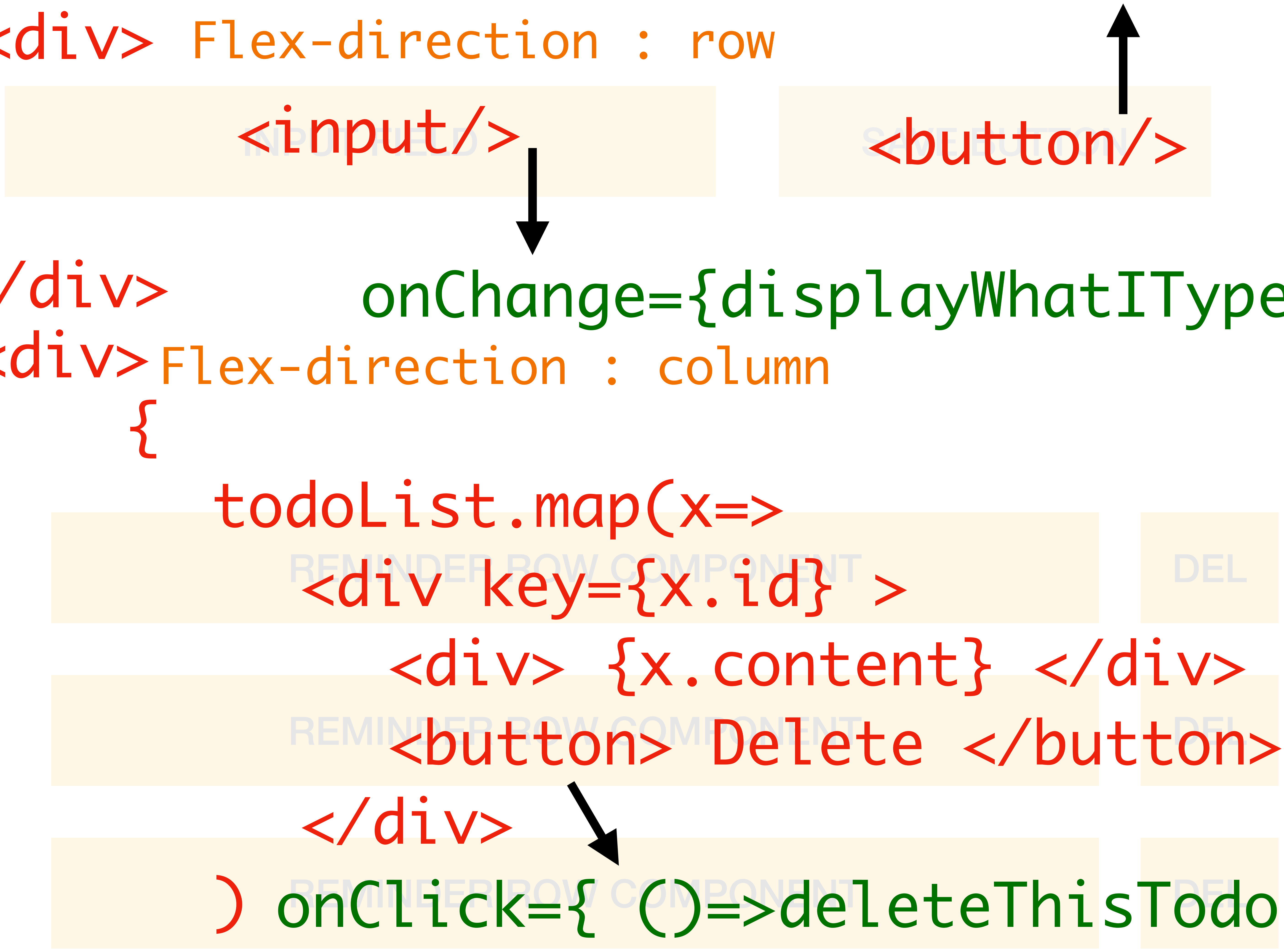
State

```
inputValue=“”

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

Checkpoint

Questions?

DEMO For today



```
<div> Flex-direction : column        onClick={saveWhatIType}
    <div> Flex-direction : row
            <input/>                      <button/>
    </div>        onChange={displayWhatIType}
    <div> Flex-direction : column
      {
         todoList.map(x=>
            <div key={x.id} >
               <div> {x.content} </div>
               <button> Delete </button>
            </div>
         ) onClick={ ()=>deleteThisTodo(id) }
      }
    </div>
</div>
```

State

```
inputValue=“”

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

DEMO For today

```
<div> Flex-direction : column
    {
        todoList.map(x=>
            <div key={x.id} >
                <div> {x.content} </div>
                <button> Delete </button>
            </div>
        ) onClick={ ()=>deleteThisTodo(id) }
    }
</div>
```

DEMO For today

```
 <div>Flex-direction : column
     {
        todoList.map(x=>
           <TodoRow/>
        )
     }
</div>
```

DEMO For today

```
 <div>Flex-direction : column
      {
         todoList.map(x=>
           <TodoRow data={x} deleteThisTodoProp={deleteThisTodo} />
         )
      }
</div>
```

```
 Const TodoRow = (props) =>{
    Let { data , deleteThisTodoProp } = props
       <div key={data.id} >
       <div> {data.content} </div>
       <button> Delete </button>
    </div>
 }
                          onClick={ ()=>deleteThisTodoProp(data.id) }
```

Checkpoint

Questions?

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

.
.
.

Thank You

http://bit.ly/3dc-web-dev

Bryce - telegram @brycerice