JAVASCRIPT

- Lesson 1 : ECMAScript 2015
- Lesson 2 : NodeJS
- Lesson 3 : CSS
- Lesson 4 : ReactJS

WHO IS SPEAKING?

CLAUDE DIOUDONNAT

Graduated from IUT (DUT + Lic Pro Mobile)

Work at ITNetwork

Worked at:

- Modis
- Université Blaise Pascal
- In My City

CLAUDE@DIOUDONNAT.FR

RULES

#1 RTFM

#2 Use your Brain

#3 Ask Questions

ECMASSCRIPT 2015

HISTORY

- LiveScript May 1995
- Brendan Eich
- Server

- LiveScript -> Javascript
- March 1996 Netscape Navigator 2.0
- JScript By Microsoft
- August 1996 Internet Explorer 3.0

ECMA-262

June 1997	ES1
June 1998	ES2
December 1999	ES3
\	ES4
December 2009	ES5
June 2015	ES6

*Under Conditions...

DATA TYPE

```
var foo;
typeof foo;
```

Type	Result
Undefined	"undefined"
Null / Object	"object"
Boolean	"boolean"
Number	"number"
String	"string"
Function object	"function"

```
123     // integer

12.345  // fractional

12.3e4  // 'scientific' notation: 12.3 * 10^4
```

```
12/0;  // Infinity

-12/0; // -Infinity
```

CONTEXT

```
var foo = "outside";
function test() {
    foo = "inside";
    console.log('test foo is "%s"', foo);
}

test();
```

```
var foo = "outside";
function test() {
    var foo = "inside";
    console.log('test foo is "%s"', foo);
}

test();
```

SYNTAX

```
if (typeof a === "undefined") {
    // a is undefined
} else if (a instanceof Number) {
    // a is a Number
} else {
    throw new Error('Hey, you missed the contract !')
}
```

```
switch(action) {
    case 'draw':
        drawit();
        break;
    case 'eat':
        eatit();
        break;
    default:
        donothing();
}
```

```
var obj = {
    foo: null
};

"foo" in obj;
```

```
try {
  foo.bar();
} catch (e) {
  if (e instanceof EvalError) {
    ...
  }
} finally {
    foo.baz();
}
```

- Error
- EvalError
- RangeError
- ReferenceError
- SyntaxError
- TypeError
- URIError

```
while (!is_empty(pancakes)) {
    ...
}
```

```
do {
    ...
} while (i < 5);
```

```
for (var i = 0; i < 9; i++) {
    ...
}
```

```
for (var prop in obj) {
    ...
}
```

OBJECT

```
class Animal {
  constructor(name) {
    this.name = name;
  }

  speak() {
    console.log(this.name + ' makes a noise.');
  }
}
```

```
class Dog extends Animal {
  speak() {
    console.log(this.name + ' barks.');
  }
}
```

```
function Animal(name) {
    this.name = name;
}

Animal.prototype.speak = function() {
    console.log(this.name + ' makes a noise.');
}
```

```
Dog.prototype = new Animal();

Dog.Dog.prototype.speak = function() {
    console.log(this.name + ' barks.');
}
```

```
var mini = new Dog('mini');
```

JSON

```
var obj = {}
```

PROMISE

```
new Promise(function(resolve, reject) {

    console.log(resolve);

    if(false) {
        resolve('Success!');
    }
    else {
        reject('Failure!');
    }
});
```

```
function readFile(path) {
    return new Promise(....);
}
const p = readFile('my-file');
```

```
p.then(function(s) {
    console.log(s);
    return s.other;
}).then(function(other) {
...
}).catch(function(e) {
    console.log(e);
})
```

AJAX - FETCH

```
var request = new Request('http://localhost:9312/all', {
    headers: new Headers({
        'Accept': 'application/json'
    })
});
```

```
fetch(request, {
    mode: 'no-cors',
    method: 'GET'
}).then(function(response) {
    return response.json();
}).then(function(j) {
    console.log(j)
}).catch(function(err) {
   console.log(err);
});
```

MODULE

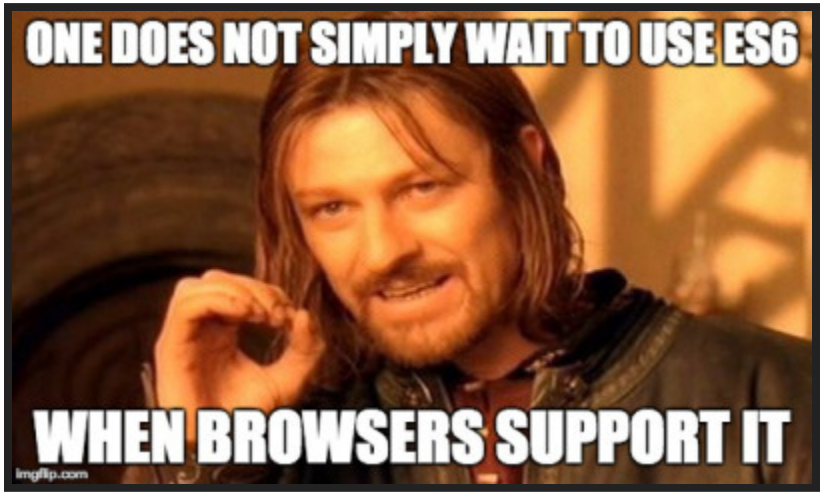
```
//include.js
export default function () { .... };
```

```
//main.js
import a from "./include";
```

```
//include.js
export a function () { .... }
export b function () { .... }
```

```
//main.js
import { a as z, b } from "./include";
```

BABELJS


ONE DOES NOT SIMPLY WAIT TO USE ES6
WHEN BROWSERS SUPPORT IT
imgflip.com

TRANSPILER

WEBPACK

NPM/YARN

- package manager
- https://www.npmjs.com/

```
{
  "name": "Pratice1",
  "description": "Pratice 1",
  "dependencies": {
    "exports-loader": "^0.6.2",
    "uglify-loader": "^1.3.0",
  }
}
```

```
$ npm install
$ npm install webpack
$ npm install -g webpack
```

More ....

NODEJS

HISTORY

MAY 2009

INTRODUCTION

- Asynchronous I/O Framework
- V8 engine
- Write in C++

MODULE

dal.js

```
class DAL {
    ...
}
module.exports = DAL
```

function.js

```
function a() { ... }
function b() { ... }
exports.a = a
exports.b = b
```

index.js

```
var DAL = require('./dal')
var utils = require('./function')
var url = require('url')
var dal_1 = new DAL();
utils.a()
utils.b()
url.parse(....)
```

REQUIRE('HTTP')

```
var http = require("http");
http.createServer(function(request, response) {
    response.write('Coucou');
    response.end();
}).listen(9999);
```

REQUEST

```
GET /thing?id=1 HTTP/1.1
```

```
console.log(request.method)
```

```
GET /thing?id=1 HTTP/1.1
```

```
var url = require('url');
var pathname = url.parse(request.url).pathname;
console.log(pathname)
```

```
GET /thing?id=1 HTTP/1.1
```

```
var url = require('url');
var parsedUrl = url.parse(request.url, true);
var queryAsObject = parsedUrl.query;
console.log(queryAsObject.id)
```

```
GET /thing?id=1 HTTP/1.1

{"status": "ok", "extended": true}
```

```
var body =  '';
request.on('data', function (data) {
    body += data;
});
request.on('end', function () {
   console.log(body);
})
```

RESPONSE

```
response.writeHead(201, {"Content-Type": "application/json"});
response.write('<html><body>Hello World</body></html>');
```

```
response.end()
```

- 1xx
- 2xx
- 3xx
- 4xx
- 5xx

CALLBACK

```
some_function('john', 'doe', function() {
return 9;
});
```

```
class Tools {
    do() { return 9}
}
var tool = new Tools();
some_function('john', 'doe', tool.do);
```

```
class Tools {
    constructor(number) { this.number = number}
    do() { return this.number }
}
var tool = new Tools(9);
some_function('john', 'doe', tool.do);
```

CLOSURE

```
function ajouteur(nombre) {
    function ajoute(valeur) {
        return valeur + nombre;
    }

    return ajoute;
}
```

```
var ajoute10 = ajouteur(10);
ajoute10(1); // retourne 11
```

```
var name = 'John'
ma_fonction('foo', function() {
    console.log(name);
})
```

```
class Tools {
    constructor(number) { this.number = number}
    do() { return this.number }
}
var tool = new Tools(9);
some_function('john', 'doe', function() { tool.do() });
```

REQUIRE('CHILD_PROCES')

```
#!/usr/bin/env bash
for count in `seq 1 100`; do
    echo $count
    sleep 0.1
done
```

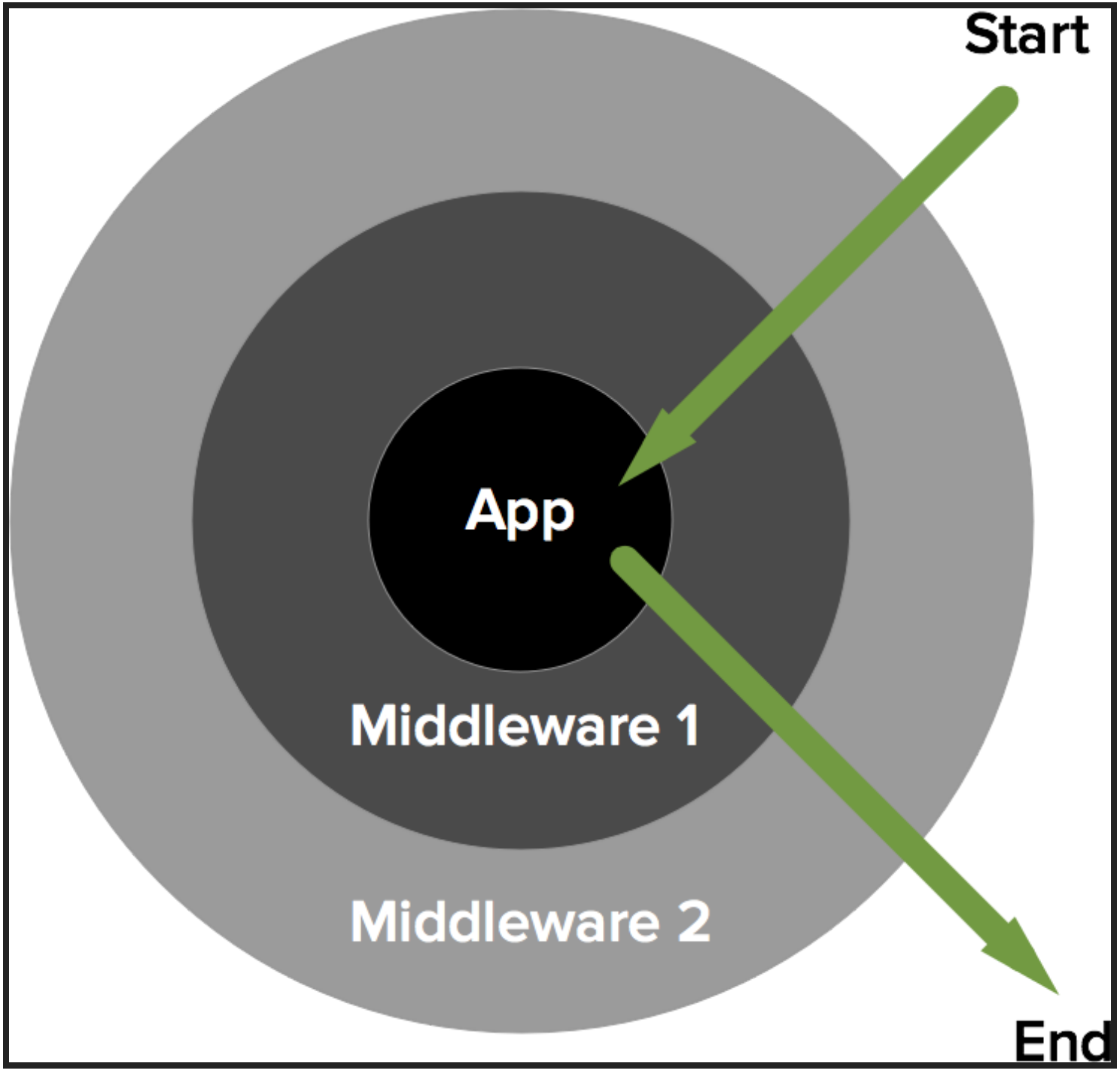
```
var spawn = require('child_proces').spawn()
var job = spawn('./script.sh');
job.stdout.on('data', function(tick) {

});
job.on('exit', function() {
    console.log('exit script.sh');
});
```

CONTROLLEUR

ROUTEUR

MIDDLEWARE


Start
App
Middleware 1
Middleware 2
End

BONUS

```
JSON.stringify(obj)
JSON.parse('{title: "foo", chapter: "bar"}')
```

EXPRESS

CSS


CSS
IS
AWESOME

HISTORY

- CSS 1 December 1996
- CSS 2 May 1998
- CSS 2.1 June 2011
- CSS 3 Release since 2001
- CSS 4 draft


HTML
HTML
+
CSS

SYNTAX

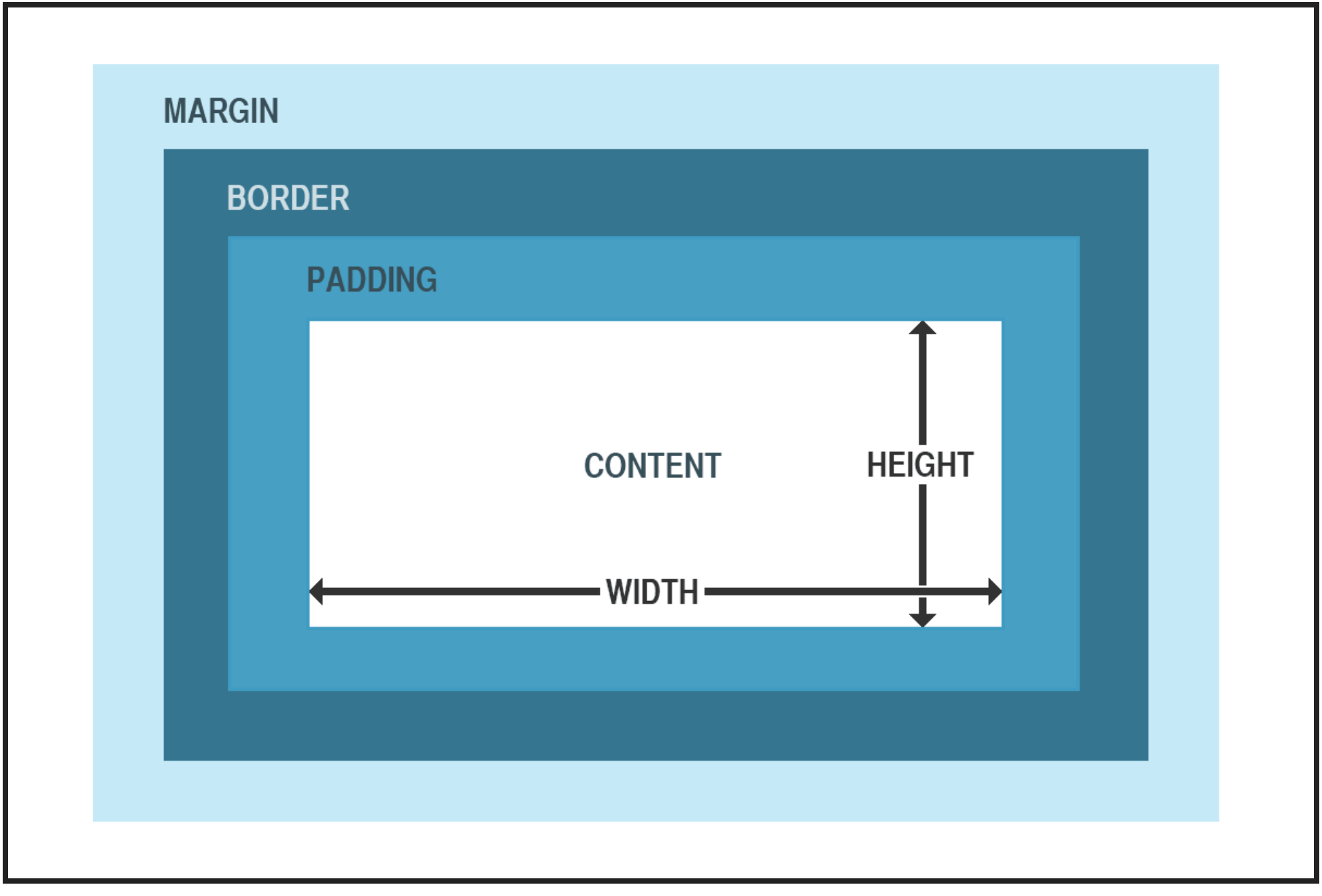
```
selector {
    porperty: value;
}
```

```
p.foo {
    color: #f00;
}

p {
    color: #ccc;
}
```

```
p {
    color: #ccc;
    color: rgba(255, 125, 125, .3);
    color: blabla;
}
```

BOX MODEL


MARGIN
BORDER
PADDING
CONTENT
HEIGHT
WIDTH

SIZE UNIT

EM

FONT

PX

resolution

%

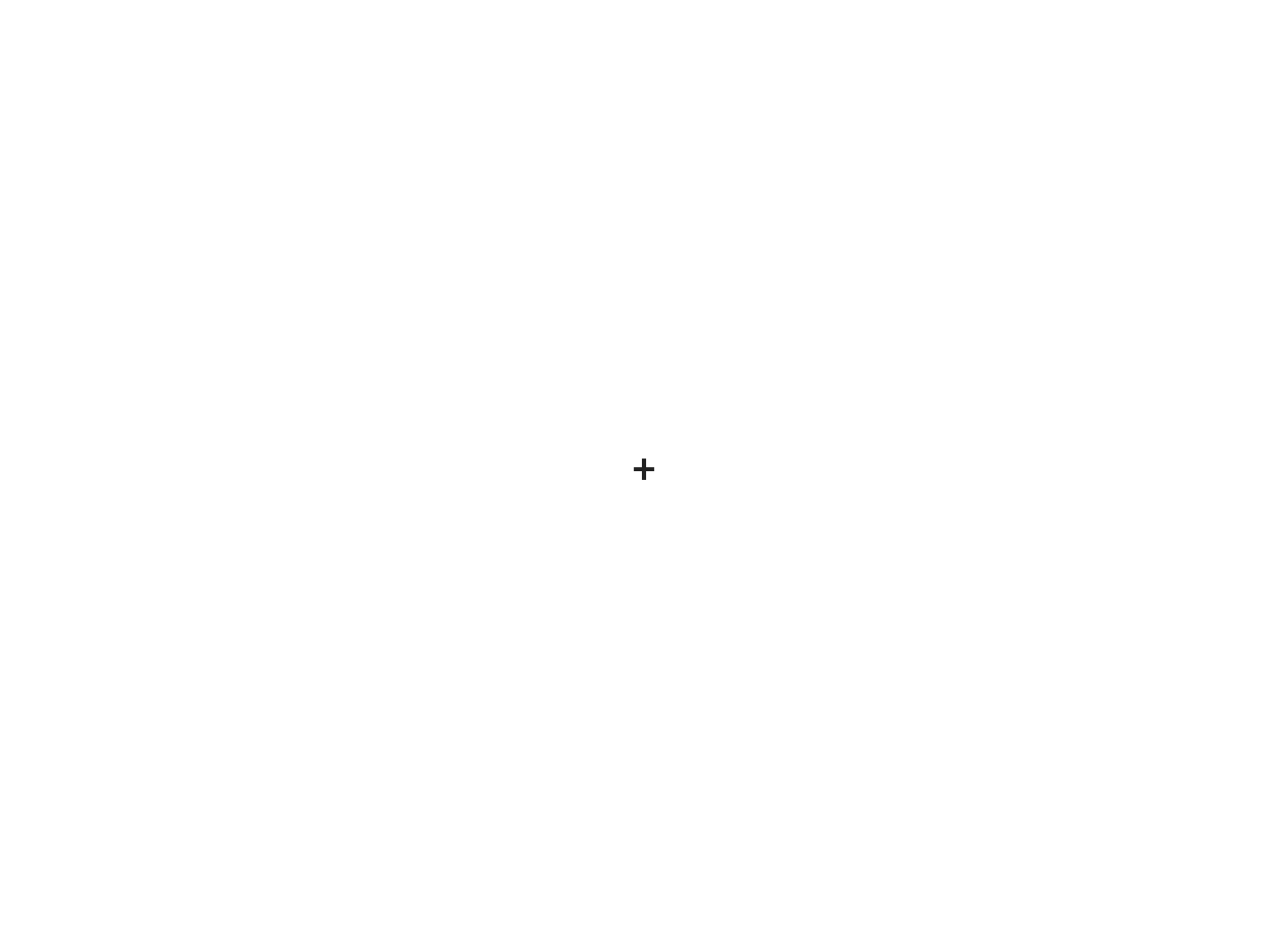
parent

DISPLAY

BLOCK

INLINE-BLOCK

INLINE



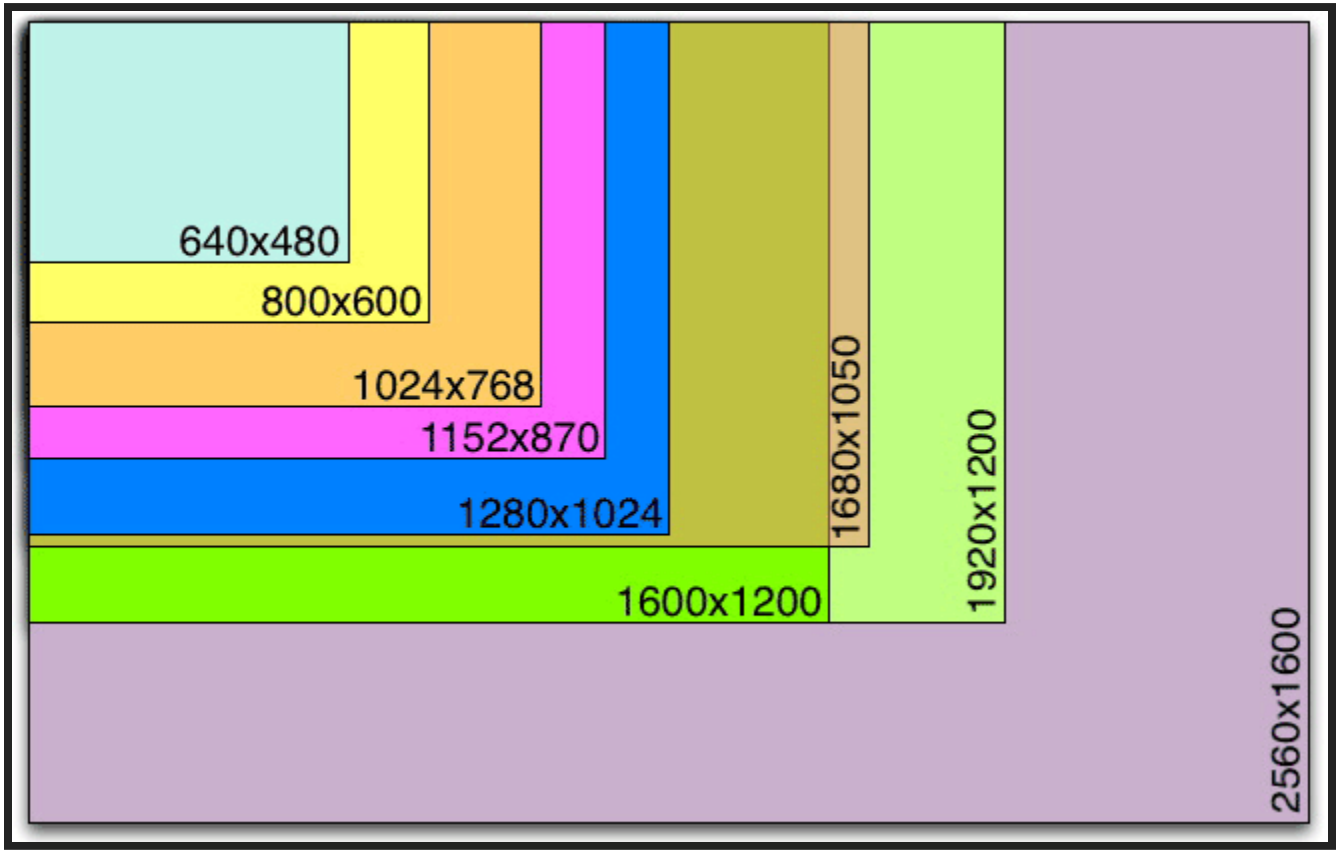
POSITION

STATIC

RELATIVE

ABSOLUTE

MEDIA QUERIES

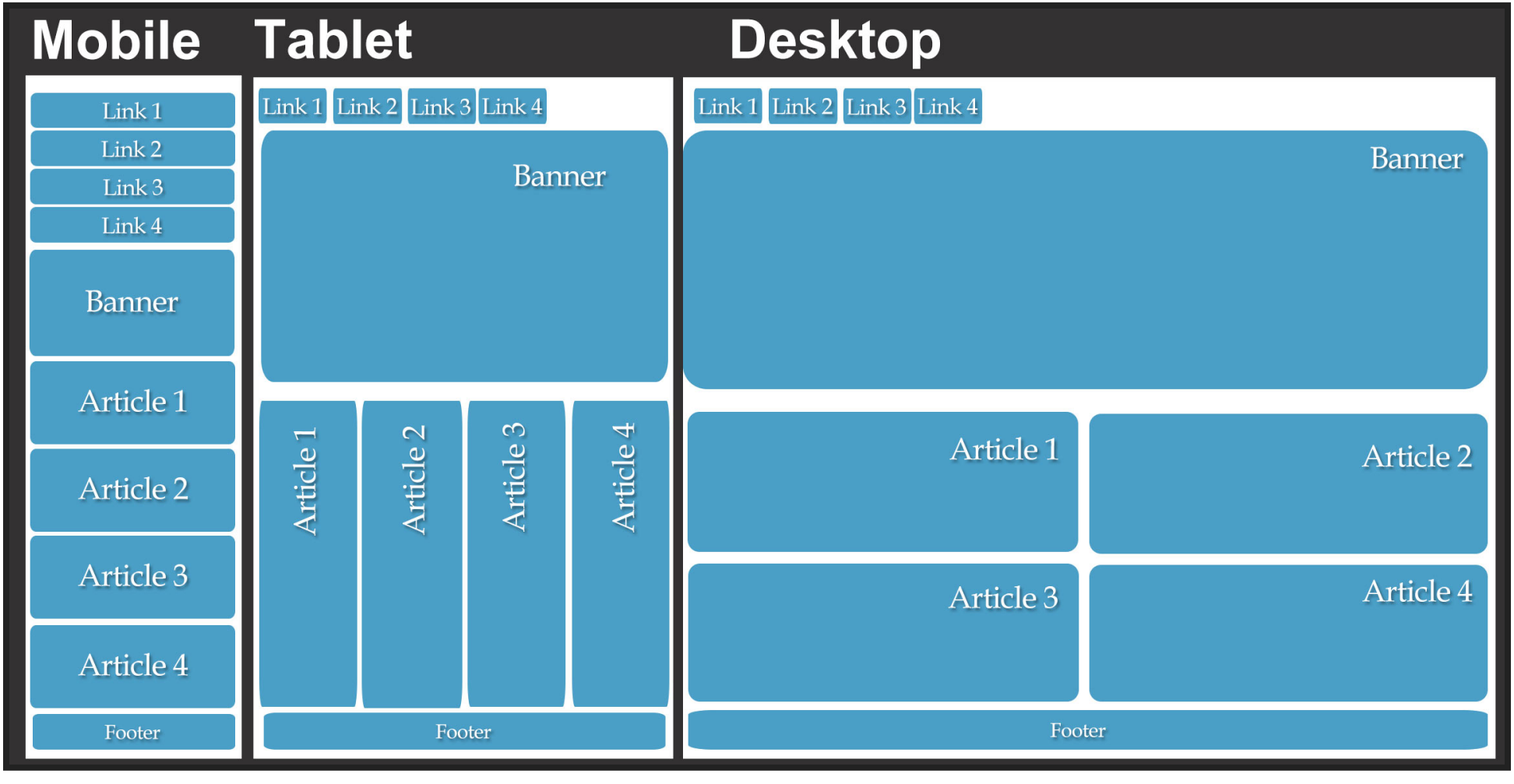

640x480
800x600
1024x768
1152x870
1280x1024
1600x1200
1680x1050
1920x1200
2560x1600

```
p {
    font-size: 30px;
}
```

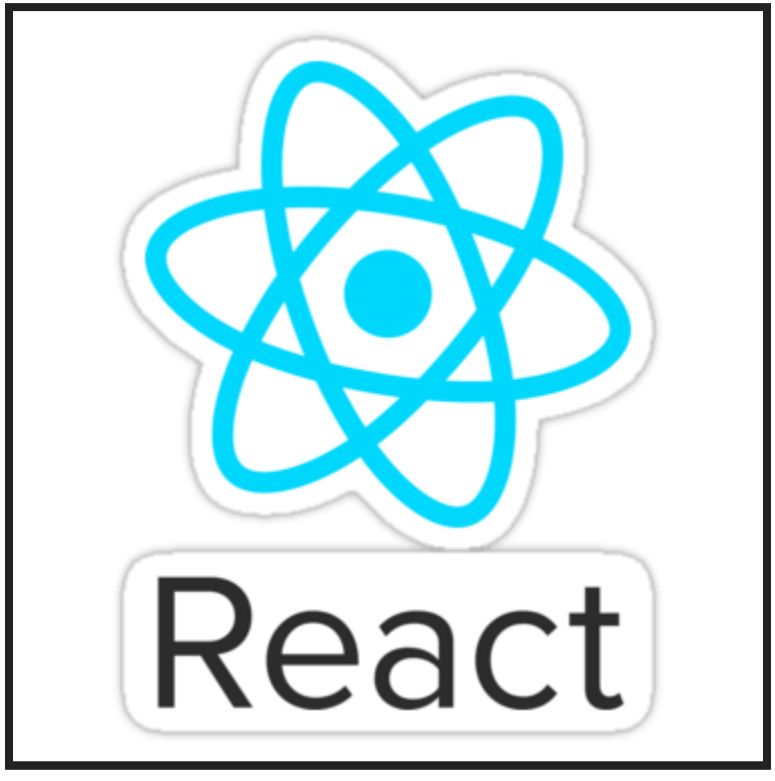
```
@media screen and (max-width: 980px) {
    p {
        font-size: 20px;
    }
}
```

```
@media screen and (max-width: 650px)
    p {
        font-size: 10px;
    }
}
```

RESPONSIVE DESIGN


Mobile
Tablet
Desktop
Link 1
Link 2
Link 3
Link 4
Banner
Article 1
Article 2
Article 3
Article 4
Footer
Link 1
Link 2
Link 3
Link 4
Banner
Article 1
Article 2
Article 3
Article 4
Footer
Link 1
Link 2
Link 3
Link 4
Banner
Article 1
Article 2
Article 3
Article 4
Footer

REACTJS



INTRODUCTION

- Open Source (MIT)
- Facebook
- library view oriented

COMPONENT

- No Controllers
- No Templates
- No Global Event Listeners
- No Models
- No View Models

JUST COMPONENTS

id	Serie	Season	Episode
agvgfj	The Big Bang Theory	1	1
hvgnik	Breaking Bad	5	13

Serie

Serie

Season

Season

Episode

Episode

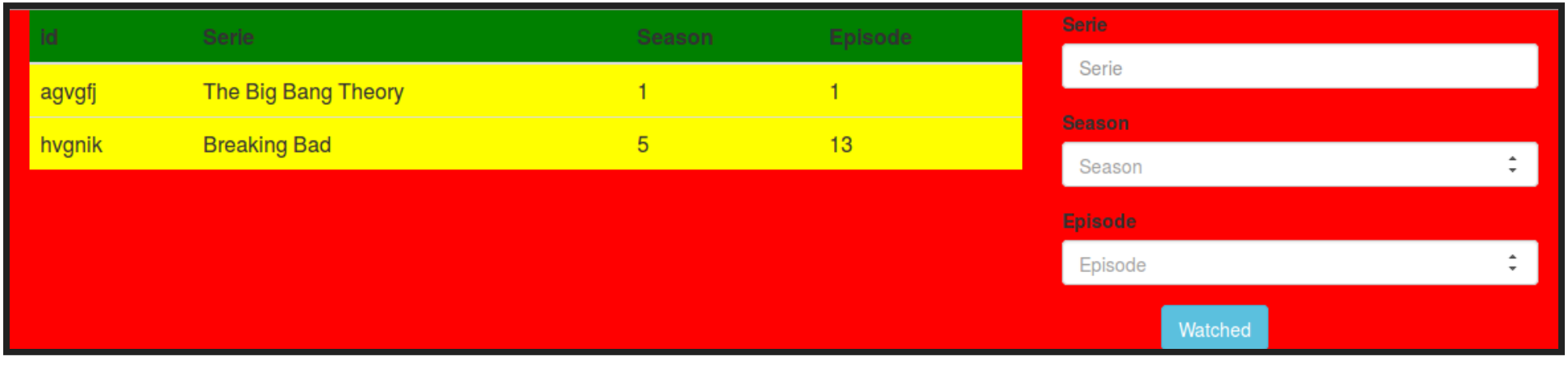
Watched

EPISODECOMPONENT

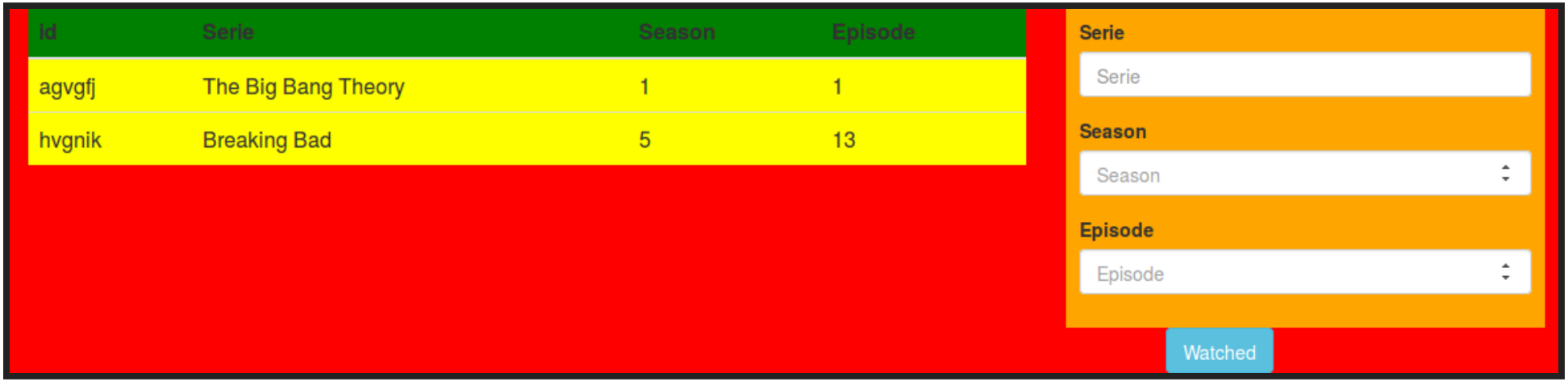
EPISODELISTCOMPONENT

EPISODEITEMCOMPONENT

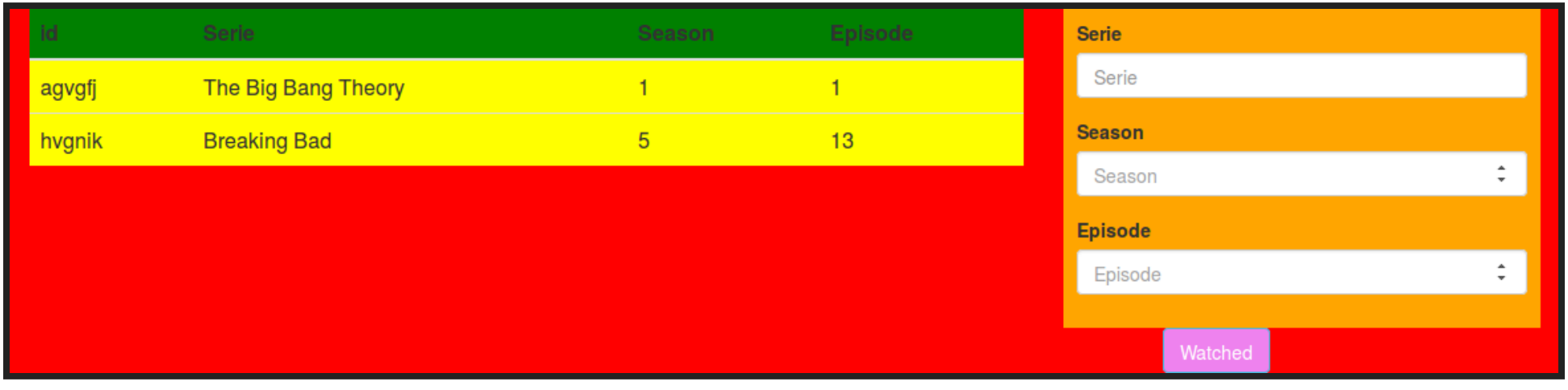
EPISODEITEMCOMPONENT



EPISODEFORMCOMPONENT



BUTTONCOMPONENT



- Composable
- Reusable
- Maintainable
- Testable

IT'S A IDEMPOTENT FUNCTION WHO DESCRIBE YOUR UI AT A SPECIFIC TIME AND IN A VERY PRECISELY STATE

VIRTUAL DOM

TOUCHING THE DOM IS EVIL

- It's Hard to test
- It's expensive
- It's Inconsistent

JSX

```
var ButtonComponent = React.createClass({
    getInitialState: function() {
        return {watched: false};
    },
    handleClick: function(event) {
      this.setState({watched: !this.state.watched});
    },
    render: function() {
        var text = this.state.watched ? 'Watched' : 'Unwatched';
        return (
            <button onclick="{this.handleClick}">
                {text}
            </button>
        );
    }
});
```

JSX = JAVASCRIPT XML

JSX

```
render: function() {
    var text = this.state.watched ? 'Watched' : 'Unwatched';
    return (
        <button onclick="{this.handleClick}">
            {text}
        </button>
    );
}
```

JSX COMPILED

```
render: function() {
    var text = this.state.watched ? 'Watched' : 'Unwatched';
    return (
        React.DOM.button({onclick:this.handleClick},
            text
        )
    )
}
```

ONCLICK IS A DOM EVENT

```
return(
    <div>
        <ButtonComponent key="watched" />
    </div>
)
```

- In-memory representation of the DOM
- render is call at every change
- React update the reald DOM
- It's Fast

```
componentDidMount: function() {
    var el = this.getDOMNode();
    this.setState({ ...});
}
```

- JSX is a DSL
- Must be compiled (babel)
- Very usefull ...
- ... but Optional

EACH DATA UPDATE

- Build a virtual tree
- Make a diff with the previous tree
- Store atomic operations in a queue
- Execute this operations

LIFECYCLE AND API

- componentWillMount
- componentDidMount
- componentWillUnmount

- render()
- getInitializeState() : {}
- setState({}) : trigger render

PROPS AND STATE

```
return(
    <div>
        <ButtonComponent key="watched" />
    </div>
)
```

```
var ButtonComponent = React.createClass({
    ...
        this.props.key
    ...
})
```

PROPS ARE IMMUTABLE

```
this.props.key = 'tutu' // Forbiden
```

STATE IS MUTABLE

```
this.setState({ active: true});
...
this.state.active
```

"MOUNT" IN REAL DOM

```
React.renderComponent(EpisodeComponent(),
    document.querySelector("body"));
```