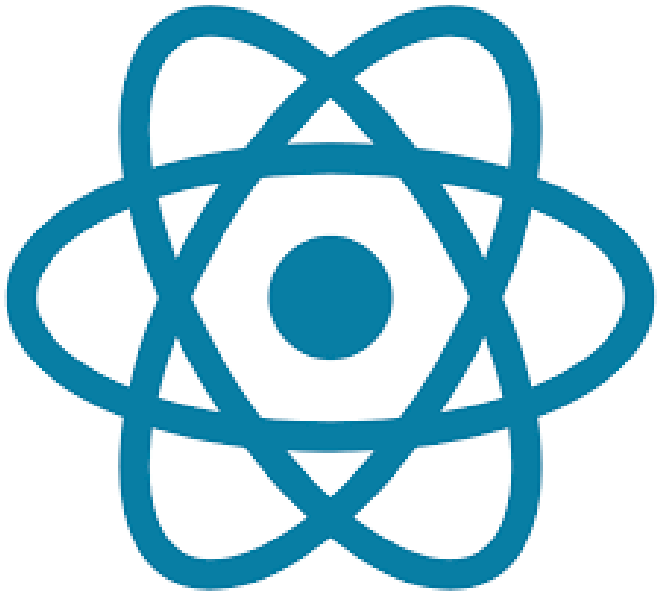
www.confucius.by



YAROSHEVICH
Andrey Olegovich

www.YAROSHEVICH.com
ao@dot.net.by
+375 29 254-07-92

IESED
Co-funded by the Erasmus+ Programme of the European Union

Browser on users computer

Internet

Web server

JS

HTML

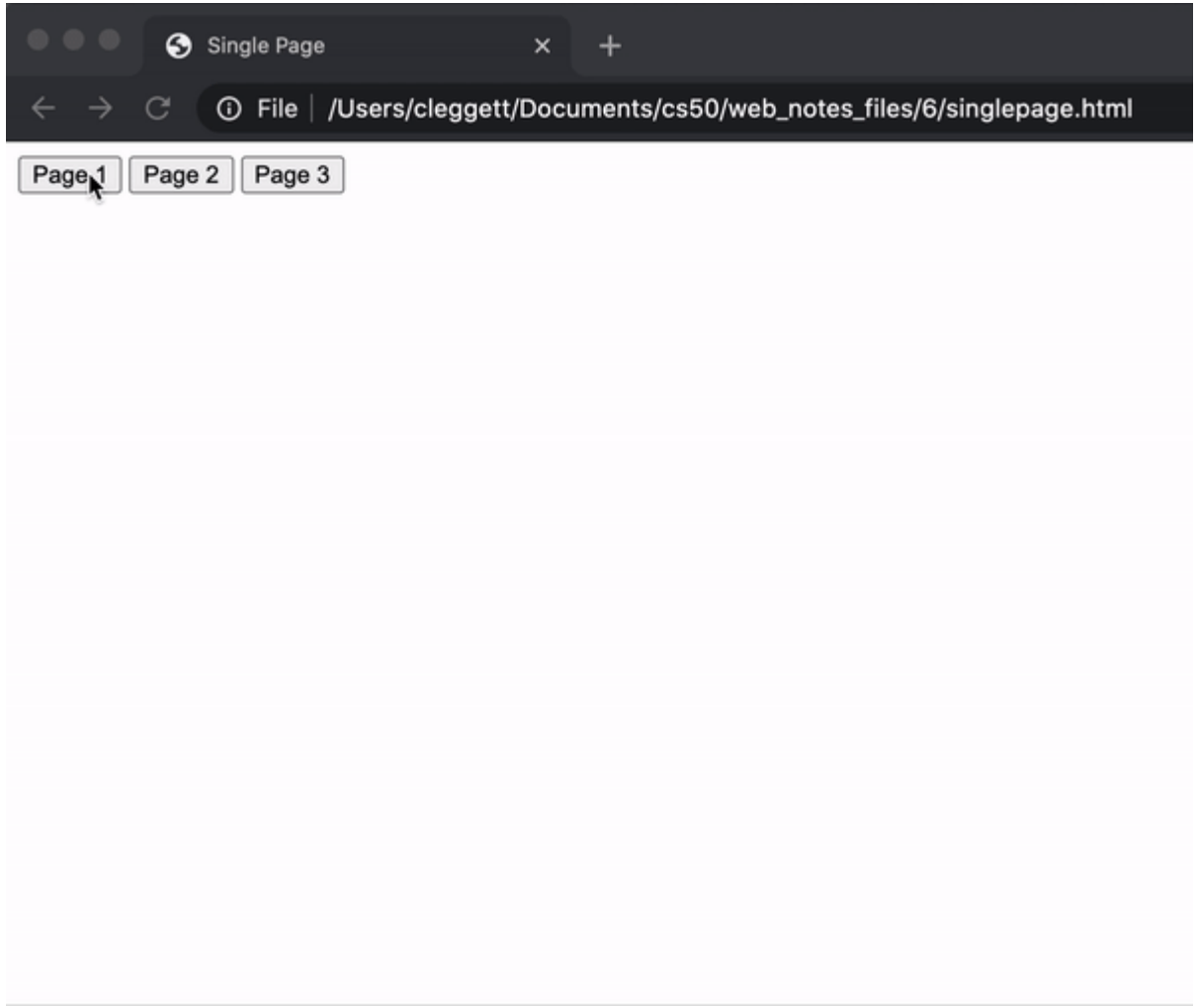
CSS

ASP.NET Webforms

Visual Studio

C# Programming

Single Page Applications



singlepage0.html

singlepage0.1.html

```html
<html lang="en">
    <head>
        <title>Single Page</title>
        <style>
          div {
                display: none;
            }
        </style>
    </head>
    <body>
        <button data-page="page1">Page 1</button>
        <button data-page="page2">Page 2</button>
        <button data-page="page3">Page 3</button>
        <div id="page1">
            <h1>This is page 1.</h1>
        </div>
        <div id="page2">
            <h1>This is page 2.</h1>
        </div>
        <div id="page3">
            <h1>This is page 3.</h1>
        </div>
    </body>
</html>
```

```html
<script>

function showPage(page) {
    // Hide all other pages
    document.querySelectorAll('div').forEach(div => {
        div.style.display = 'none';
    });

    // Show requested page
    document.querySelector(`#${page}`).style.display = 'block';
}

document.addEventListener('DOMContentLoaded', function() {
    document.querySelectorAll('button').forEach(button => {
        button.onclick = function() {
         showPage(this.dataset.page);
        };
    });
});
</script>
```

Imperative programming

View:

```
<h1>0</h1>
```

Logic:

```
let num = parseInt(.querySelector("h1").innerHTML);
num += 1;
document.querySelector("h1").innerHTML = num;
```

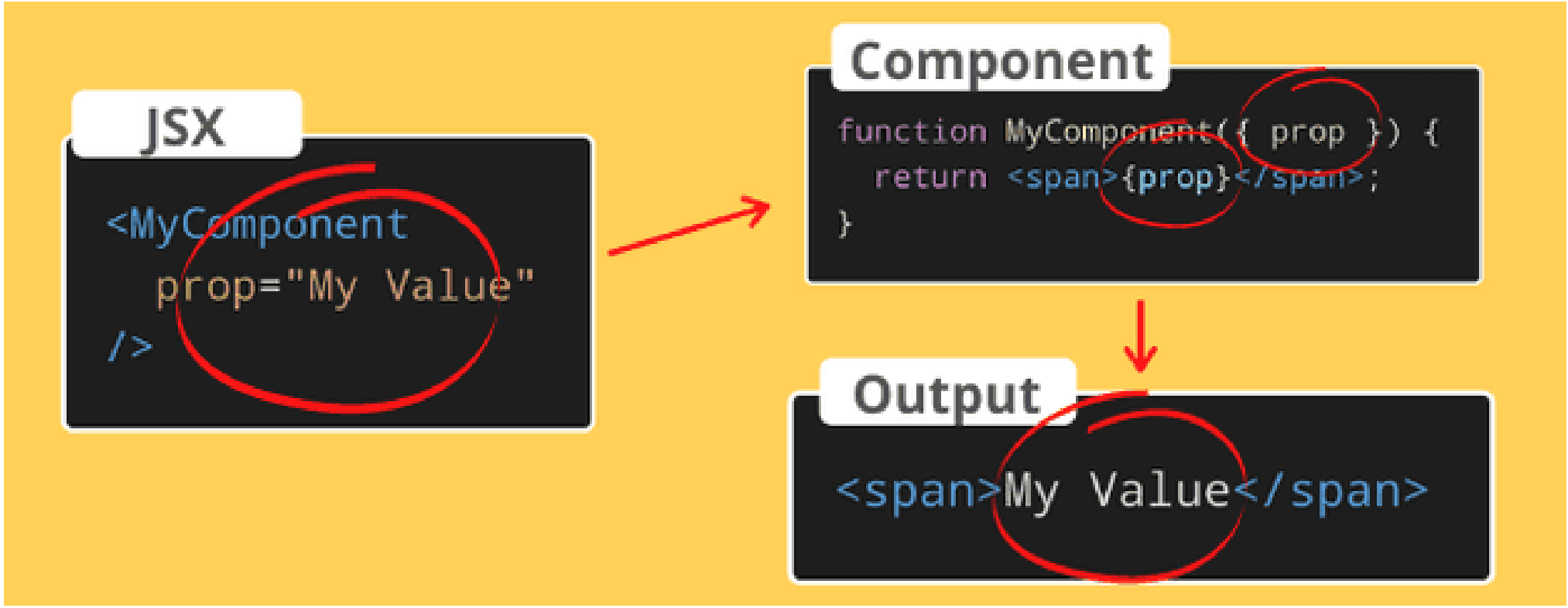
Declarative programming

View:

```
<h1>{num}</h1>
```

Logic:

```
num += 1;
```


JSX
<MyComponent
prop="My Value"
/>
Component
function MyComponent({ prop }) {
return {prop};
}
Output
My Value

To use **React**, we'll have to import three JavaScript Packages:

React: Defines components and their behavior

```
<script src= "https://unpkg.com/react@16/umd/react.production.min.js"></script>
```

ReactDOM: Takes React components and inserts them into the DOM

```
<script src= "https://unpkg.com/react-dom@16/umd/react-dom.production.min.js">
</script>
```

Babel: Translates from JSX, the language in which we'll write in React, to plain JavaScript that our browsers can interpret.

```
<script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
```

ReactDOM.render 01.htm

In the three lines above the title, we import the latest versions of React, ReactDOM, and Babel.

```
<html lang="en">
<title>Test React</title>
<script src= "https://unpkg.com/react@16/umd/react.production.min.js"></script>
<script src= "https://unpkg.com/react-dom@16/umd/react-dom.production.min.js">
</script>
<script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
```

In the body, we include a single div with an id of app.

```
<body>
<div id="id01">SBMT</div>
<script type="text/babel">
ReactDOM.render(
  <h1>School of Business</h1>,
  document.getElementById('id01'));
</script>
</body>
</html>
```

Component App 01.1.htm

In the three lines above the title, we import the latest versions of React, ReactDOM, and Babel.

```
<html lang="en">
<title>Test React</title>
<script src= "https://unpkg.com/react@16/umd/react.production.min.js"></script>
<script src= "https://unpkg.com/react-dom@16/umd/react-dom.production.min.js">
</script>
<script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
```

In the body, we include a single div with an id of app.

```
<body>
<div id="id01">SBMT</div>
```

We create a component called **App**. Components in React can be represented by JavaScript functions.

```
<script type="text/babel">
 function App() {
            return (
                  <h1>School of Business</h1>              );
        }
ReactDOM.render(  <App />,  document.getElementById('id01'));
</script>
</body>
</html>
```

```
function SB_CN(props) {
        return (
            <h1>白俄罗斯国立大学商学院</h1>
            );
    }
```

01.2.htm

```
function SB_EN(props) {
        return (
            <h1>School of Business BSU</h1>
            );
    }
```

```
function App() {
        return (
            <div>
                <SB_CN />
                <SB_EN />
            </div>

            );
    }
```

https://asp.net.by/React/01.3.htm

props in React

```
function SB(props) {
        return (
            <h1>{props.text}</h1>
           );
    }

    }
   function App() {
      return (
        <div>
          <SB text="白俄罗斯国立大学商学院" />
          <SB text="School of Business BSU:” />
        </div>
      );
   }
```

https://asp.net.by/React/01.5.htm

```
<div id="app"></div>

        <script type="text/babel">

            function App() {

                const [count, setCount] = React.useState(0);

                function updateCount() {
                    setCount(count + 1);
                }

                return (
                    <div>
                        <h1>{count}</h1>
                        <button onClick={updateCount}>Count</button>
                    </div>
                );
            }

            ReactDOM.render(<App />, document.querySelector("#app"));
        </script>
```

inside App component, we'll use React.useState hook to add state to our component.

updateCount function.

setCount function, can take as argument a new value for the state.

The argument to useState is the initial value of the state. which we'll set to 0.

Now, we can work on what the function will render, where we'll specify a header

and a button. We'll also add an event listener for when the button is clicked, which React handles using the onClick attribute:

Game “Binary Logarithm”

Link

```
const [state, setState] = React.useState({
                num1: 1,
                response: "",
                score: 0,
                incorrect: false
            });
```

The number
2 to the power of **X**

What the user has typed in like **X**

How many questions the user has answered correctly.

using the values in the state, we can render a basic user interface.

```
return (
 <div>
  <div className={state.incorrect ? "incorrect" : ""} id="problem">
    Lb( {state.num1})= <input onKeyPress={inputKeyPress} onChange={updateResponse} autoFocus={true}
                                                        value={state.response} />

</div>
```

Lb(1024)= 10

Score: 9

```
      <div>Score: {state.score}</div>
          </div>
        )
```

onChange attribute to the ir
element, and set it equal to
function called updateResponse

```
function updateResponse(event) {
   setState({
     ...state,
     response: event.target.value
                 });
              }
```

takes in the event that triggered the function

```
function updateResponse(event) {
    setState({
      ...state,
      response: event.target.value
                    });
                }
```

sets the response to the current value of the input.

check whether the ”Enter” key was pressed

check to see if the answer is correct

increase the score by 1

onKeyPress={**inputKe**

Lb(1024)= 10

Score:

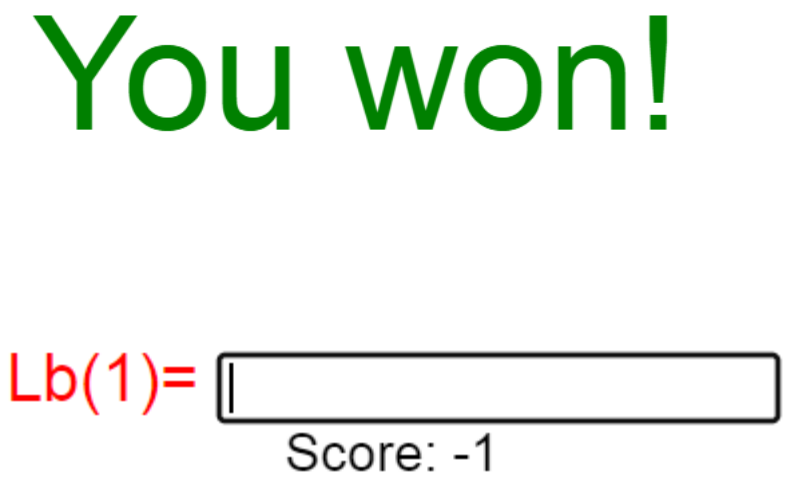
```
function inputKeyPress(event) {
    if (event.key === "Enter") {
        const answer = parseInt(state.response);
        if (answer === Math.log2( state.num1 )) {
            // User got question right
            setState({
                ...state,
                score: state.score + 1,
                response: "",
                num1: 2**(Math.ceil(Math.random() * 10)),
                incorrect: false
            });
        } else {// User got question wrong
              setState({
                ...state,
                score: state.score - 1,
                response: "",
                incorrect: true
            })
        }
    }
```

```
<style>
        #app {
            text-align: center;
            font-family: sans-serif;
        }

        #problem {
            font-size: 20px;
        }

        #winner {
            font-size: 72px;
            color: green;
        }

        .incorrect {
            color: red;
        }
    </style>
```



```
if (state.score === 10) {
        return renderWinScreen();
    } else {
        return renderProblem();
    }
```

```
function renderWinScreen() {
    return (
        <div id="winner">You won!</div>
    );
}
```

```
function renderProblem()
{
  return (
  <div>
    <div className={state.incorrect ? "incorrect" : ""} id="problem">
      Lb( {state.num1})= <input onKeyPress={inputKeyPress} onChange={updateResponse}
                                autoFocus={true} value={state.response} />
   </div>
  <div>Score: {state.score}</div>
        </div>
    }
```

Let’s take a look at our application!

http://www.confucius.by/React/Log2.htm

http://asp.net.by/React/02.htm

```
<div id="id01">SBMT</div>
<script type="text/babel">
const name='Business';
ReactDOM.render(
  <h1>School of {name}</h1>,
  document.getElementById('id01'));
</script>
```

http://asp.net.by/React/03.htm

```
<body>
<div id="root"></div>
<script type="text/babel">
ReactDOM.render(<h1>School of business</h1>,
document.getElementById('root'));
</script>

</body>
```

https://asp.net.by/React/04.htm

21:15:08

```
<div id="root"></div>

<script type="text/babel">

function tick() {
const element=(<i>{new
Date().toLocaleTimeString()}</i>);
ReactDOM.render(element,
                        document.getElementById('root'));
}

setInterval(tick, 3000);
</script>
```

React components = JavaScript functions

https://asp.net.by/React/05.htm

```
<div id="root">SBMT</div>
<script type="text/babel">
    function Business() {
        return <h1>School of Business!</h1>;
     }
ReactDOM.render(<Business />, document.getElementById('root'));
</script>
```

http://asp.net.by/Projects/1/1.3.DOMContentLoaded.htm

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Count</title>
        <script src="counter.js"></script>
    </head>
    <body>
        <h1>0</h1>
        <button>Count</button>
    </body>
</html>
```

counter.js

```js
let counter = 0;

function count() {
    counter++;
    document.querySelector('h1').innerHTML = counter;

    }
}

document.addEventListener('DOMContentLoaded',
function() {
    document.querySelector('button').onclick = count;
});
```

```
<body>
    <div id="app"></div>
     <script type="text/babel">
        function App() {
          const [count, setCount] = React.useState(0);
            function updateCount() {
              setCount(count + 1);
             }
            return (
              <div>
                <h1>{count}</h1>
                 <button onClick={updateCount}>Count</button>
              </div>
            );
        }
        ReactDOM.render(<App />, document.querySelector("#app"));
</script>
</body>
</html>
```