**Tạo máy tính đơn giản bằng ReactJS**

Trong bài này ta sẽ xây dựng một ứng dụng máy tính đơn giản bằng ReactJS.

Để hiểu được ví dụ này thì bạn phải tìm hiểu qua [JSX trong ReactJS](https://freetuts.net/jsx-trong-reactjs-2344.html) để xử lý sự kiện click, cách tạo một component và kết thừa từ lớp Component của ReactJS để xây dựng các thành phần giao diện của ứng dụng.

**Mục lục**

* [Cách tạo máy tính đơn giản bằng ReactJS](https://freetuts.net/tao-may-tinh-don-gian-bang-reactjs-2625.html#goto-h2-0)
  + [File src/App.js](https://freetuts.net/tao-may-tinh-don-gian-bang-reactjs-2625.html#goto-h3-0)
  + [File src/App.css](https://freetuts.net/tao-may-tinh-don-gian-bang-reactjs-2625.html#goto-h3-1)

**Cách tạo máy tính đơn giản bằng ReactJS**

Để xây dựng ứng dụng này thì trước tiên cần tạo mới một project và đặt tên cho nó là example4, bằng cách sử dụng lệnh dưới đây trong cửa sổ Command Prompt.

|  |  |
| --- | --- |
| 1 | create-react-app example4 |

Bây giờ ta sẽ quan tâm đến hai file chính, thứ nhất là file App.js và thứ hai là file App.css.

*Bài viết này được đăng tại [free tuts .net]*

Hãy mở hai file này lên và copy hai đoạn code dưới đây.

**File src/App.js**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148 | import React, { Component } from 'react';  import './App.css';    class App extends Component {    constructor(props){      super(props);      this.state = {        display: "0",        equation: ""      }      this.numInput = this.numInput.bind(this);      this.operInput = this.operInput.bind(this);      this.decInput = this.decInput.bind(this);      this.clearInput = this.clearInput.bind(this);      this.calculate = this.calculate.bind(this);    }      numInput(e){      if(this.state.equation.match(/[0-9\.]$/) && !this.state.equation.includes("=")){        if(this.state.equation.match(/[+\-\*\/]/) == null){          let val = this.state.equation + e.currentTarget.value;          this.setState({            display: val,            equation: val          });        } else {          this.setState({            display: this.state.display + e.currentTarget.value,            equation: this.state.equation + e.currentTarget.value          });        }      } else if(this.state.equation.match(/[+\-\*\/]$/)){        let val = this.state.equation + e.currentTarget.value;        this.setState({          display: e.currentTarget.value,          equation: val        });      } else if(this.state.display === "0" && e.currentTarget.value !== "0" || this.state.equation.includes("=")) {        this.setState({          display: e.currentTarget.value,          equation: e.currentTarget.value        });      }    }      operInput(e){      if(this.state.equation.includes("=")){        let val = this.state.display;        val += e.currentTarget.value;        this.setState({          equation: val        });      } else {        if(this.state.equation != "" && this.state.equation.match(/[\*\-\/+]$/) == null){          let val = this.state.equation;          val += e.currentTarget.value;          this.setState({            equation: val          });        } else if(this.state.equation.match(/[\*\-\/+]$/) != null){          let val = this.state.equation;          val = val.substring(0, (val.length-1));          val += e.currentTarget.value;          this.setState({            equation: val          });        }      }    }      decInput(e){      if(this.state.equation == "" || this.state.equation.includes("=")){        let val = '0.';        this.setState({          display: val,          equation: val        });      } else if(this.state.equation.match(/[+\-\*\/]$/)){        let val = '0.';        this.setState({          display: val,          equation: this.state.equation + val        });      } else if(!this.state.display.includes(".")){        this.setState({          display: this.state.display + e.currentTarget.value,          equation: this.state.equation + e.currentTarget.value        });      }    }      clearInput(){      this.setState({        display: "0",        equation: ""      });    }      calculate(){      if(this.state.equation.includes("=")){        let val = `${this.state.display} = ${this.state.display}`;        this.setState({          equation: val        });      } else if(this.state.equation != "" && this.state.equation.match(/[+\-\*\/]/) != null && this.state.equation.match(/[+\-\*\/]$/) == null) {        let result = Number.isInteger(eval(this.state.equation)) ? eval(this.state.equation) : parseFloat(eval(this.state.equation).toFixed(5));        let val = this.state.equation;        val += ` = ${result}`;        this.setState({          display: result,          equation: val        });      }    }      render() {      return (        <div className="container">          <Display equation={this.state.equation} display={this.state.display} />          <Button id="clear" value="clear" display="AC" class="row-3 col-1" click={this.clearInput} />          <Button id="sign" value="+/-" display="±" class="row-3 col-2" />          <Button id="percent" value="%" display="%" class="row-3 col-3" />          <Button id="divide" value="/" display="÷" class="oper row-3 col-4" click={this.operInput} />          <Button id="seven" value="7" display="7" class="num row-4 col-1" click={this.numInput} />          <Button id="eight" value="8" display="8" class="num row-4 col-2" click={this.numInput} />          <Button id="nine" value="9" display="9" class="num row-4 col-3" click={this.numInput} />          <Button id="multiply" value="\*" display="×" class="oper row-4 col-4" click={this.operInput} />          <Button id="four" value="4" display="4" class="num row-5 col-1" click={this.numInput} />          <Button id="five" value="5" display="5" class="num row-5 col-2" click={this.numInput} />          <Button id="six" value="6" display="6" class="num row-5 col-3" click={this.numInput} />          <Button id="subtract" value="-" display="−" class="oper row-5 col-4" click={this.operInput} />          <Button id="one" value="1" display="1" class="num row-6 col-1" click={this.numInput} />          <Button id="two" value="2" display="2" class="num row-6 col-2" click={this.numInput} />          <Button id="three" value="3" display="3" class="num row-6 col-3" click={this.numInput} />          <Button id="add" value="+" display="+" class="oper row-6 col-4" click={this.operInput} />          <Button id="zero" value="0" display="0" class="num row-7 col-1-2" click={this.numInput} />          <Button id="decimal" value="." display="." class="num row-7 col-3" click={this.decInput} />          <Button id="equals" value="=" display="=" class="oper row-7 col-4" click={this.calculate} />        </div>      );    }  }    const Display = props => <div id="calc-display" className="row-1-2 col-1-4"><span id="eq">{props.equation}</span><span id="dis">{props.display}</span></div>;    const Button = props => <button type="button" id={props.id} value={props.value} className={props.class} onClick={props.click}>{props.display}</button>;    export default App; |

Trong hàm khởi tạo của component App ta có truyền một tham số tên là props. Tai jhàm khởi tạo constructor này ta tiếp tục truyền tham số đó vào phương thức super: super(props)

**File src/App.css**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85 | @font-face {    font-family: "Digital";    src: url("//db.onlinewebfonts.com/t/8e22783d707ad140bffe18b2a3812529.eot");    src: url("//db.onlinewebfonts.com/t/8e22783d707ad140bffe18b2a3812529.eot?#iefix") format("embedded-opentype"), url("//db.onlinewebfonts.com/t/8e22783d707ad140bffe18b2a3812529.woff2") format("woff2"), url("//db.onlinewebfonts.com/t/8e22783d707ad140bffe18b2a3812529.woff") format("woff"), url("//db.onlinewebfonts.com/t/8e22783d707ad140bffe18b2a3812529.ttf") format("truetype"), url("//db.onlinewebfonts.com/t/8e22783d707ad140bffe18b2a3812529.svg#Digital-7") format("svg");  }    body {    height: 95vh;    width: 95vw;    background-color:#696969;  }  main { height: 100%; }  .container {    height: 100%;    display: grid;    grid-template-columns: 30% repeat(4, 10%) 30%;    grid-template-rows: 12.5% 5% 20% repeat(5, 10%) 12.5%;  }    /\* Grid Classes - Columns \*/  .col-1 { grid-column: 2 / 3; }  .col-2 { grid-column: 3 / 4; }  .col-3 { grid-column: 4 / 5; }  .col-4 { grid-column: 5 / 6; }  .col-1-2 { grid-column: 2 / 4; }  .col-1-4 { grid-column: 2 / 6; }  /\* Grid Classes - Rows \*/  .row-1 { grid-row: 2 / 3; }  .row-2 { grid-row: 3 / 4; }  .row-3 { grid-row: 4 / 5; }  .row-4 { grid-row: 5 / 6; }  .row-5 { grid-row: 6 / 7; }  .row-6 { grid-row: 7 / 8; }  .row-7 { grid-row: 8 / 9; }  .row-1-2 { grid-row: 2 / 4; }    /\* Calculator Display \*/  #calc-display {    width: 100%;    height: 100%;    background-color: black;    color: white;    display: flex;    flex-direction: column;    justify-content: center;    align-items: flex-end;    font-family: digital;  }  #dis, #eq { padding-right: 10%; }  #dis {    font-size: 4vw;    margin-top: 20px;  }  #eq { font-size: 1.5vw; }    /\* Calculator Buttons \*/  button {    -webkit-appearance: none;    -moz-appearance: none;    appearance: none;    display: inline-block;    margin: 0;    text-align: center;    text-decoration: none;    border: 1px solid black;    cursor: grab;    font-size: 2.5em;  }  button:hover {    opacity: 0.7;    border: 1.5px solid white;  }  button:focus { outline: 0; }  .num { background-color: #A9A9A9; }  .oper { background-color: orange; }    /\* Responsive for Small Screens \*/  @media only screen and (max-width: 767px) {    .container{      grid-template-columns: 6% repeat(4, 22%) 6%;      grid-template-rows: 5% 5% 20% repeat(5, 13%) 5%;    }    #dis { font-size: 12vw; }    #eq { font-size: 4vw; }  } |

Hãy chạy demo để xem thành quả của các bạn nhé. Hoặc click vào link demo mà mình đã cung cấp ở trên cũng được.