1)calculator

App.js

```
=====================================
import React, { useState } from "react";
import './App.css'
const App = () => {
   const[result,setResult] = useState("");
   const handleClick = (e) =>{
      setResult(result.concat(e.target.name));
   }
   const clear = () =>{
      setResult("");
   }
   const calculate = () =>{
      try{
         setResult(eval(result).toString());
      }catch(err){
         setResult("Error");
      }
   }
   const backspace = () =>{
      setResult(result.slice(0,-1));
   }
   return(
      <>
         <div className="container">
            <form>
               <input type="text" value={result} />
            </form>
            <div className="keypad">
               <button className="highlight" onClick={clear}
id="clear">Clear</button>
               <button className="highlight" onClick={backspace}
id="backspace" >C</button>
               <button className="highlight" onClick={handleClick}
name="/">&divide;</button>
               <button onClick={handleClick} name="7">7</button>
               <button onClick={handleClick} name="8">8</button>
               <button onClick={handleClick} name="9">9</button>
               <button className="highlight" onClick={handleClick}
name="*">x</button>
```

```
                <button onClick={handleClick} name="4">4</button>
                <button onClick={handleClick} name="5">5</button>
                <button onClick={handleClick} name="6">6</button>
                <button className="highlight" onClick={handleClick} name="-">-
</button>
                <button onClick={handleClick} name="1">1</button>
                <button onClick={handleClick} name="2">2</button>
                <button onClick={handleClick} name="3">3</button>
                <button className="highlight" onClick={handleClick}
name="+">+</button>
                <button onClick={handleClick} name="0">0</button>
                <button onClick={handleClick} name=".">.</button>
                <button className="highlight" onClick={calculate}
id="result">=</button>
            </div>
        </div>
    </>
  );
}

export default App;
```

2)Life cycle

App.js

=========

```
import React, { Component } from 'react';
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {hello: "JavaTpoint"};
    this.changeState = this.changeState.bind(this)
  }
  render() {
    return (
      <div>
        <h1>ReactJS component's Lifecycle</h1>
        <h3>Hello {this.state.hello}</h3>
        <button onClick = {this.changeState}>Click Here!</button>
      </div>
    );
  }
  componentWillMount() { console.log('Component Will MOUNT!')  }
  componentDidMount() {  console.log('Component Did MOUNT!')  }
  changeState(){
    this.setState({hello:"All!!- Its a great reactjs tutorial."});
  }
  componentWillReceiveProps(newProps) {
    console.log('Component Will Recieve Props!')
  }
  shouldComponentUpdate(newProps, newState) {
    return true;
  }
  componentWillUpdate(nextProps, nextState) {
    console.log('Component Will UPDATE!');
  }
  componentDidUpdate(prevProps, prevState) {
    console.log('Component Did UPDATE!')
  }
  componentWillUnmount() {
    console.log('Component Will UNMOUNT!')
  }
} export default App;
```

3)react router

App.js

==========

```
import React from 'react'
import { BrowserRouter,  Route, Routes } from 'react-router-dom'
import About from './About'
import Dashboard from './Dashboard'
import Home from './Home'
import Navbar from './Navbar'

export default function App() {
   return (
      <div >
         <BrowserRouter>

            <Navbar />

            <Routes>
               <Route path="/" exact element={<Home />} />
               <Route path='/dashboard' exact element={<Dashboard />} />
               <Route path="/about" exact element={<About />}/>
            </Routes>
         </BrowserRouter>
      </div>
   )
}
```

Navbar.js

==========

```
import React from 'react'
import { Link } from 'react-router-dom'
import './App.css'
export default function Navbar() {
   return (
      <div >
         <ul>
            <Link to="/"><li>Home</li></Link>
            <Link to="/dashboard"><li>Dashboard</li></Link>
            <Link to="/about"><li>About</li></Link>
         </ul>
      </div>
```

```
    )
}
```

Home.js
=========

```
import React ,{useState} from 'react'
import {Navigate} from 'react-router-dom'

export default function Home() {
    const [data,setData]=useState(false);
    if(data){
        return <Navigate to='/dashboard' />
    }
    return (
        <div>
            <h1>This is home page</h1>
            <button onClick={()=>setData(true)}>Click me</button>
        </div>
    )
}
```

Dashboard.js
==========

```
import React from 'react'

export default function Dashboard() {
    return (
        <div>
            <h1>Dashboard page</h1>
        </div>
    )
}
```

About.js
==========

```
import React from 'react'

export default function About() {
    return (
        <div>
            <h1>About page</h1>
        </div>
    )
}
```

5)Data IO

buffer_write.js

==========

```
buf256 = new Buffer(256);
buf256.fill(0);
buf256.write("add some text");
console.log(buf256.toString());
buf256.write("more text", 9, 9);
console.log(buf256.toString());
buf256[18] = 43;
console.log(buf256.toString());
```

output:

```
C:\books\node\ch05>node buffer_write.js
add some text
add some more text
add some more text+
```

---

buffer_read.js

==========

```
bufUTF8 = new Buffer("Some UTF8 Text \u00b6 \u30c6 \u20ac", 'utf8');
console.log(bufUTF8.toString());
console.log(bufUTF8.toString('utf8', 5, 9));
var StringDecoder = require('string_decoder').StringDecoder;
var decoder = new StringDecoder('utf8');
console.log(decoder.write(bufUTF8));
```

output:

```
C:\books\node\ch05>node buffer_read.js
Some UTF8 Text ¶ テ €
UTF8
Some UTF8 Text ¶ テ €
e3
e3838620
```

stream_read.js

```
================
var stream = require('stream');
var util = require('util');
util.inherits(Answers, stream.Readable);
function Answers(opt) {
   stream.Readable.call(this, opt);
   this.quotes = ["yes", "no", "maybe"];
   this._index = 0;
 }
 Answers.prototype._read = function() {
   if (this._index > this.quotes.length){
   this.push(null);
 } else {
   this.push(this.quotes[this._index]);
   this._index += 1;
 }
 };
 var r = new Answers();
 console.log("Direct read: " + r.read().toString());
 r.on('data', function(data){
 console.log("Callback read: " + data.toString());
 });
 r.on('end', function(data){
  console.log("No more answers.");
 });
```

output:

```
C:\books\node\ch05>node stream_read.js
Direct read: yes
Callback read: no
Callback read: maybe
No more answers.
```

stream_write.js:

==========

```
var stream = require('stream');
var util = require('util');
util.inherits(Writer, stream.Writable);
function Writer(opt) {
stream.Writable.call(this, opt);
this.data = new Array();
 }
Writer.prototype._write = function(data, encoding, callback) {
this.data.push(data.toString('utf8'));
console.log("Adding: " + data);
callback();
 };
var w = new Writer();
 for (var i=1; i<=5; i++){
w.write("Item" + i, 'utf8');
}
w.end("ItemLast");
 console.log(w.data);
```

output:

```
C:\books\node\ch05>node stream_write.js
Adding: Item1
Adding: Item2
Adding: Item3
Adding: Item4
Adding: Item5
Adding: ItemLast
[ 'Item1', 'Item2', 'Item3', 'Item4', 'Item5', 'ItemLast' ]
```

zlib_buffers.js [Compressing/decompressing buffers using the Zlib module]

```
================
var zlib = require("zlib");
var input = '.............text.............';
zlib.deflate(input, function(err, buffer) {
  if (!err) {
    console.log("deflate (%s): ", buffer.length, buffer.toString('base64'));
    zlib.inflate(buffer, function(err, buffer) {
      if (!err) {
        console.log("inflate (%s): ", buffer.length, buffer.toString());
      }
    });
    zlib.unzip(buffer, function(err, buffer) {
      if (!err) {
        console.log("unzip deflate (%s): ", buffer.length, buffer.toString());
      }      });    }      });
zlib.gzip(input, function(err, buffer) {
if (!err) {
 console.log("gzip (%s): ", buffer.length, buffer.toString('base64'));
zlib.gunzip(buffer, function(err, buffer) {
 if (!err) {
 console.log("gunzip (%s): ", buffer.length, buffer.toString());
 }
 });
 zlib.unzip(buffer, function(err, buffer) {
if (!err) {
console.log("unzip gzip (%s): ", buffer.length, buffer.toString());
}      });    }
});
```

output:

```
C:\books\node\ch05>node zlib_buffers.js
deflate (18): eJzT00MBJakVJagiegB9Zgcq
deflateRaw (12): 09NDASWpFSWoInoA
gzip (30): H4sIAAAAAAAAAC9PTQwElqRUlqCJ6AIq+x+AiAAAA
inflate (34): .............text.............
unzip deflate (34): .............text.............
inflateRaw (34): .............text.............
gunzip (34): .............text.............
unzip gzip (34): .............text.............
```

6)File System Module

file_write.js: Writing a JSON string to a file

==========

```
var fs = require('fs');
var config = {
maxFiles: 20,
maxConnections: 15,
rootPath: "/webroot"
};
var configTxt = JSON.stringify(config);
var options = {encoding:'utf8', flag:'w'};
fs.writeFile('config.txt', configTxt, options, function(err){
 if (err){
 console.log("Config Write Failed.");
 } else {
 console.log("Config Saved.");
 }
 });
```

output:

```
C:\books\node\ch06\writing>node file_write.js
Config Saved.
```

---

file_write_stream.js: [Implementing a Writable stream to allow streaming writes to a file]

==========

```
 var fs = require('fs');
  var grains = ['wheat', 'rice', 'oats'];
 var options = { encoding: 'utf8', flag: 'w' };
  var fileWriteStream =
 fs.createWriteStream("grains.txt", options);
  fileWriteStream.on("close", function(){
 console.log("File Closed.");
 });
 while (grains.length){
 var data = grains.pop() + " ";
 fileWriteStream.write(data);
 console.log("Wrote: %s", data);
  }
  fileWriteStream.end();
```

output:

```
Wrote: oats
Wrote: rice
Wrote: wheat
File Closed.
```

file_read.js:

==========

```
var fs = require('fs');
var options = {encoding:'utf8', flag:'r'};
fs.readFile('config.txt', options, function(err, data){
if (err){
console.log("Failed to open Config File.");
} else {
console.log("Config Loaded.");
var config = JSON.parse(data);
console.log("Max Files: " + config.maxFiles);
console.log("Max Connections: " + config.maxConnections);
console.log("Root Path: " + config.rootPath);
}
});
```

output:

```
C:\books\node\ch06\reading>node file_read.js
Config Loaded.
Max Files: 20
Max Connections: 15
Root Path: /webroot
```

---

file_read_stream.js: [Implementing a Readable stream to allow streaming reads from a file]

==========

```
var fs = require('fs');
var options = { encoding: 'utf8', flag: 'r' };
var fileReadStream = fs.createReadStream("grains.txt", options);
fileReadStream.on('data', function(chunk) {
console.log('Grains: %s', chunk);
console.log('Read %d bytes of data.', chunk.length);
});
fileReadStream.on("close", function(){
console.log("File Closed.");
});
```

output:

```
Grains: oats rice wheat
Read 16 bytes of data.
File Closed.
```

file_stats.js: [Implementing an fs.stats() call to retrieve information about a file]
==========
```
var fs = require('fs');
fs.stat('file_stats.js', function (err, stats) {
if (!err){
console.log('stats: ' + JSON.stringify(stats, null, ' '));
console.log(stats.isFile() ? "Is a File" : "Is not a File");
console.log(stats.isDirectory() ? "Is a Folder" : "Is not a Folder");
console.log(stats.isSocket() ? "Is a Socket" : "Is not a Socket");
stats.isDirectory();
stats.isBlockDevice();
 stats.isCharacterDevice();
 //stats.isSymbolicLink(); //only lstat
 stats.isFIFO();
 stats.isSocket();
 }
 });
```
output:
```
C:\books\node\ch06>node file_stats.js
stats: {
"dev": 818973644,
"mode": 33206,
"nlink": 1,
"uid": 0,
"gid": 0,
"rdev": 0,
"ino": 1970324837052284,
"size": 535,
"atime": "2016-09-14T18:03:26.572Z",
"mtime": "2013-11-26T21:51:51.148Z",
"ctime": "2014-12-18T17:30:43.340Z",
"birthtime": "2016-09-14T18:03:26.572Z"
}
Is a File
Is not a Folder
Is not a Socket
```

```
fs.unlink("new.txt", function(err){
console.log(err ? "File Delete Failed" : "File Deleted");
});
fs.truncate("new.txt", function(err){
console.log(err ? "File Truncate Failed" : "File Truncated");
});
fs.mkdir("./data/folderA", function(err){
   console.log(err ? "Directory ceration Failed" : "Directory ceration");
})
```

```
fs.rmdir("./data/folderA/folderB/folderC", function(err){
   console.log(err ? "Directory deletion Failed" : "Directory deletion");
});
```

```
fs.rename("old.txt", "new.txt", function(err){
console.log(err ? "Rename Failed" : "File Renamed");
});
```

File watch.js

```
const fs = require('fs');
fs.watch("ex_dir", (eventType, filename) => {
  console.log("\nThe file", filename, "was modified!");
  console.log("The type of change was:", eventType);
});
setTimeout(
  () => fs.writeFileSync("ex_dir/ex1.txt",
  "The file is modified"), 1000
);
setTimeout(
  () => fs.renameSync("ex_dir/ex2.txt",
  "ex_dir/new_ex2.txt"), 2000
);
==========
output:
The file ex1.txt was modified!
The type of change was: change

The file ex2.txt was modified!
The type of change was: rename

The file new_ex2.txt was modified!
The type of change was: rename
Hello from a Static File
```

7) Http Client Server

http_client_static.js: [Basic web client retrieving static files]

==========

```
var http = require('http');
var options = {
hostname: 'localhost',
port: '8080',
path: '/hello.html'
};
function handleResponse(response) {
var serverData = '';
response.on('data', function (chunk) {
 serverData += chunk;
});
 response.on('end', function () {
 console.log(serverData);
 });
 }
 http.request(options, function(response){
 handleResponse(response);
 }).end();
```

---

http_server_static.js: [Implementing a basic static file webserver]

```
var fs = require('fs');
var http = require('http');
var url = require('url');
var ROOT_DIR = "html/";
http.createServer(function (req, res) {
var urlObj = url.parse(req.url, true, false);
fs.readFile(ROOT_DIR + urlObj.pathname, function (err,data) {
if (err) {
 res.writeHead(404);
  res.end(JSON.stringify(err));
 return;
  }
  res.writeHead(200);
  res.end(data);
  });
  }).listen(8080);
```

Output:

```
/>node http_server_static.js
<html>
<head>
<title>Static Example</title>
</head>
<body>
<h1>Hello from a Static File</h1>
</body>
</html>
```