

Eight Ball: Solution



[Download our solution](#)

eightball/src/EightBall.js

```
import React, { useState } from "react";
import "./EightBall.css";

function choice(arr) {
  const randomIdx = Math.floor(Math.random() * arr.length);
  return arr[randomIdx];
}

function EightBall(props) {
  const [msg, setMsg] = useState("Think of a Question.");
  const [color, setColor] = useState("black");

  function handleClick() {
    const { msg, color } = choice(props.answers);
    setMsg(msg);
    setColor(color);
  }

  return (
    <div
      className="EightBall"
      onClick={handleClick}
      style={{ backgroundColor: color }}
    >
      <b>{msg}</b>
    </div>
  );
}

EightBall.defaultProps = {
  answers: [
    { msg: "It is certain.", color: "green" },
    { msg: "It is decidedly so.", color: "green" },
    { msg: "Without a doubt.", color: "green" },
    { msg: "Yes – definitely.", color: "green" },
    { msg: "You may rely on it.", color: "green" },
    { msg: "As I see it, yes.", color: "green" },
    { msg: "Most likely.", color: "green" },
    { msg: "Outlook good.", color: "green" },
    { msg: "Yes.", color: "green" },
    { msg: "Signs point to yes.", color: "goldenrod" },
    { msg: "Reply hazy, try again.", color: "goldenrod" },
    { msg: "Ask again later.", color: "goldenrod" },
    { msg: "Better not tell you now.", color: "goldenrod" },
    { msg: "Cannot predict now.", color: "goldenrod" },
    { msg: "Concentrate and ask again.", color: "goldenrod" },
    { msg: "Don't count on it.", color: "red" },
    { msg: "My reply is no.", color: "red" },
    { msg: "My sources say no.", color: "red" },
    { msg: "Outlook not so good.", color: "red" },
    { msg: "Very doubtful.", color: "red" }
  ]
};

export default EightBall;
```

eightball/src/EightBall.css

```
.EightBall {
  box-shadow: 5px 5px 10px #678;
  margin: 50px;
  border-radius: 50%;
  width: 150px;
  height: 150px;
  display: flex;
  justify-content: center;
  align-items: center;
  color: white;
  padding: 10px;
  text-align: center;
  cursor: pointer;
}
```

Color Boxes

color-boxes/src/BoxesContainer.js

```
import React, { useState } from "react";
import Box from "./Box";
import "./BoxesContainer.css";
import { choice } from "./helpers";

function BoxesContainer(props) {
  const [boxes, setBoxes] = useState(
    Array.from({ length: props.numBoxes }, () => choice(props.allColors))
  );

  /** On button click: pick random box, change to random color. */
  const handleClick = evt => {
    let idx = Math.floor(Math.random() * props.numBoxes);

    setBoxes(boxes => {
      let boxCopy = [...boxes];
      boxCopy[idx] = choice(props.allColors);
      return boxCopy;
    });
  };

  const boxComponents = boxes.map((color, i) => <Box key={i} color={color} />);

  return (
    <div>
      <section className="BoxesContainer">{boxComponents}</section>
      <button onClick={handleClick}>Change a Box</button>
    </div>
  );
}

BoxesContainer.defaultProps = {
  numBoxes: 16,
  allColors: [
    "Aqua",
    "Black",
    "BlanchedAlmond",
    "Blue",
    "Chocolate",
    "DodgerBlue",
    "Lavender",
    "LawnGreen",
    "Peru",
    "Plum",
    "SpringGreen",
    "SteelBlue",
    "Tan",
    "Teal",
    "Thistle",
    "Tomato",
    "Turquoise",
    "Violet",
    "Yellow",
    "YellowGreen"
  ]
};

export default BoxesContainer;
```

color-boxes/src/Box.js

```
import React from "react";
import "./Box.css";

/** Individual colored box. */
function Box(props) {
  return (
    <div
      className="Box"
      style={{backgroundColor: props.color}}
    />
  );
}

export default Box;
```