Rubik's Cube Program:

```
<!DOCTYPE html>
<html>
<head>
 <title>Rubik's Cube Solver (Simplified)</title>
 <style>
  body { font-family: sans-serif; padding: 20px; }
  #cubeContainer { font-size: 20px; margin-top: 20px; white-space: pre-wrap; }
 </style>
</head>
<body>
 <h1>Rubik's Cube Demo</h1>
 <button onclick="scrambleAndDisplay()">Scramble Cube</button>
 <button onclick="solveAndDisplay()">Solve Cube</button>

 <div id="cubeContainer"></div>

 <script>
  class RubiksCube {
   constructor() {
    this.faces = {
     U: Array(9).fill('w'),
     D: Array(9).fill('y'),
     F: Array(9).fill('g'),
     B: Array(9).fill('b'),
     L: Array(9).fill('o'),
     R: Array(9).fill('r')
    };
```

```
}

rotateFaceClockwise(face) {
  const f = this.faces[face];
  this.faces[face] = [
    f[6], f[3], f[0],
    f[7], f[4], f[1],
    f[8], f[5], f[2]
  ];
}

rotateF() {
  this.rotateFaceClockwise('F');
  const { U, D, L, R } = this.faces;
  const tmp = [U[6], U[7], U[8]];
  [U[6], U[7], U[8]] = [L[8], L[5], L[2]];
  [L[2], L[5], L[8]] = [D[2], D[1], D[0]];
  [D[0], D[1], D[2]] = [R[0], R[3], R[6]];
  [R[0], R[3], R[6]] = tmp;
}

toColorString() {
  return (
    this.faces['U'].join('') +
    this.faces['R'].join('') +
    this.faces['F'].join('') +
    this.faces['D'].join('') +
    this.faces['L'].join('') +
```

```
    this.faces['B'].join('')
   );
  }

  scramble(moves = 1) {
   const actions = ['F'];
   for (let i = 0; i < moves; i++) {
    const move = actions[Math.floor(Math.random() * actions.length)];
    this["rotate" + move]();
   }
  }

  solve() {
   this.rotateF();
   this.rotateF();
   this.rotateF();
  }
}

function getCubeSvg(cubeString) {
  document.getElementById('cubeContainer').textContent = cubeString;
}

const cube = new RubiksCube();

function scrambleAndDisplay() {
  cube.scramble(1);
  getCubeSvg(cube.toColorString());
```

```
  }

  function solveAndDisplay() {
    cube.solve();
    getCubeSvg(cube.toColorString());
  }

  getCubeSvg(cube.toColorString());
 </script>
</body>
</html>
```

expected Output: