

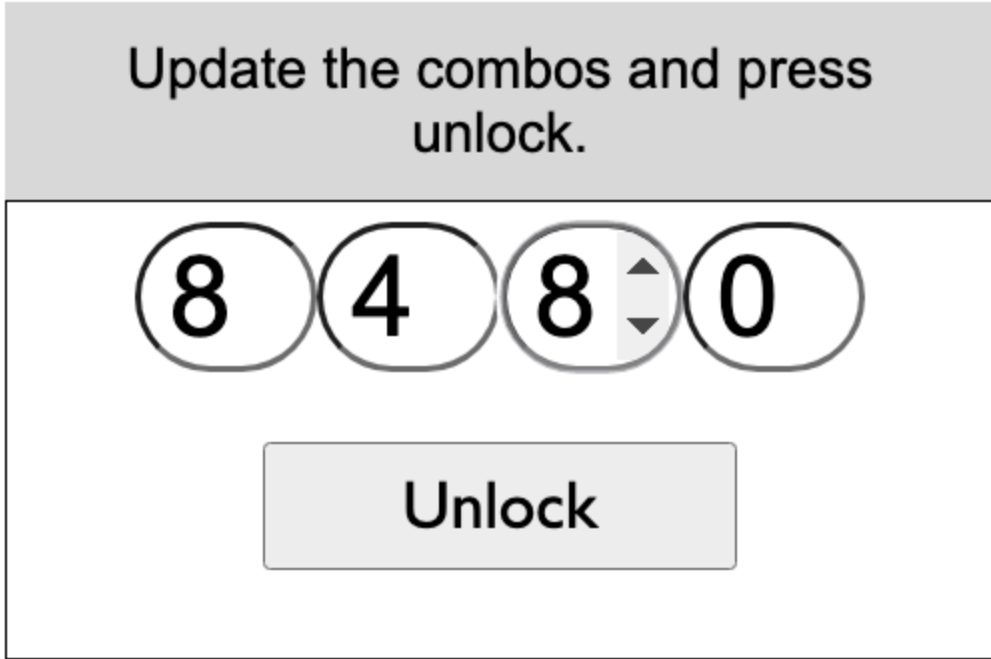


Lock Combo Guessing Game

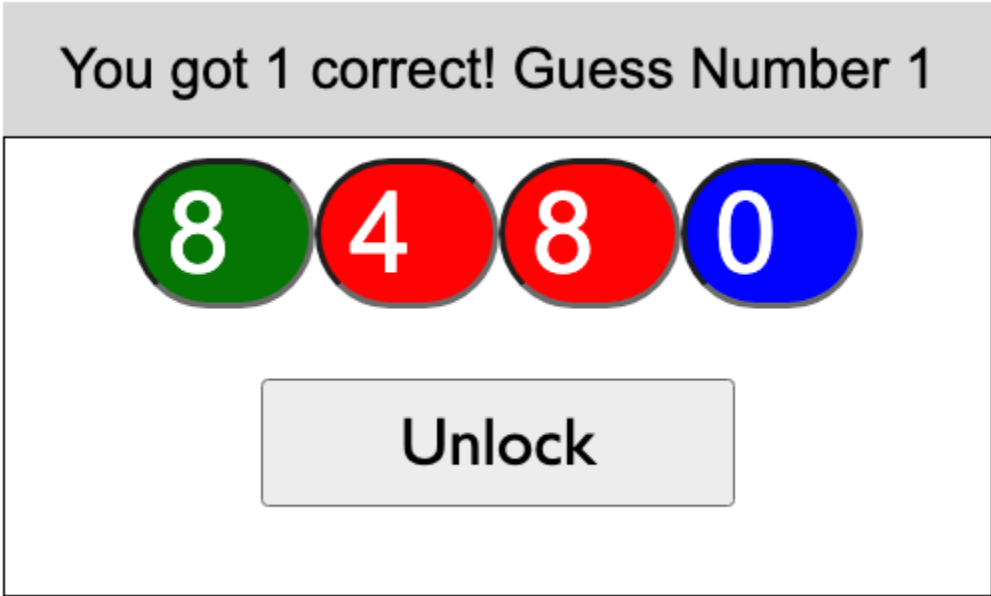
Dynamic number of lock slots, guess the correct combination to open the lock. When the guess is incorrect the background of the lock slot will be blue for too low and red for too high. Use the hints to select and guess the correct values for each slot. Once all the lock slots are selected correctly, matching the random numbers then the game is over. Scoring is done by the number of tries it takes to unlock the combo.

Press Start to Begin

Start Game



Provides a color coded response as hints to the actual combination.



Once the correct value is provided the game shows the number of guesses made and allows for a new round of game play.,

The Lock opens you got them correct. It took 5 guesses.

Start Game



```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript</title>
  <style>
    .message{
      padding:10px;
      text-align:center;
      font-size:1em;
      background-color:#ddd;
      font-family:Arial, Helvetica, sans-serif;
    }
    .gameArea{
      border:1px solid black;
      padding:5px;
      text-align:center;
```

```
}
.combo{
  border-radius:25px;
  text-align:center;
  font-size:2em;
}
.btn{
  display:block;
  margin:20px auto;
  padding:5px;
  font-size:1.2em;
  width:50%;
  font-family:'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS',
sans-serif;
}
</style>
</head>
<body>
  <div class="output"></div>
  <script src="app5.js"></script>
</body>
</html>
```

```
const output = document.querySelector('.output');
const message = maker('div',output,'Press Start to Begin','message');
const btn = maker('button',output,'Start Game','btn');
btn.onclick = startGame;
btn.style.display = 'block';
```

```

const gameArea = maker('div',output,"','gameArea');
const game = {
  combos : 4, arr:[]
};

function startGame(){
  console.log('start');
  btn.style.display = 'none';
  game.guesses = 0;
  gameArea.innerHTML = "";
  message.textContent = 'Update the combos and press unlock.'
  setUpGameBoard();
}

function setUpGameBoard(){
  for(let i=0;i<game.combos;i++){
    const ele = maker('input',gameArea,"','combo');
    ele.setAttribute('type','number');
    ele.max = 9;
    ele.min = 0;
    ele.value = Math.floor(Math.random()*10);
    const val = Math.floor(Math.random()*10);
    game.arr.push(val);
  }
  const btn1 = maker('button',gameArea,'Unlock','btn');
  btn1.onclick = checkCombo;
}

```

```

function checkCombo(){
  const ins = document.querySelectorAll('.combo');
  let counter = 0;
  game.guesses++;
  ins.forEach((el,ind)=>{
    el.style.color = 'white';
    if(el.value==game.arr[ind]){
      el.style.backgroundColor = 'green';
      counter++;
    }else{
      if(el.value > game.arr[ind]){
        el.style.backgroundColor = 'red';
      }else{
        el.style.backgroundColor = 'blue';
      }
    }

  });
})
if(counter >= ins.length){
  message.textContent = `The Lock opens you got them correct. It took
${game.guesses} guesses.`;
  endgame();
}else{
  message.textContent = `You got ${counter} correct! Guess Number
${game.guesses}`;
}
}

```

```
function endgame(){
  const btn1 = document.querySelector('.gameArea button');
  btn1.style.display = 'none';
  btn.style.display = 'block';
}
```

```
function maker(eleTag,parent,html,cla){
  const el = document.createElement(eleTag);
  el.innerHTML = html;
  el.classList.add(cla);
  return parent.appendChild(el);
}
```