

REACT TOGGLE PATTERN MASTERY GUIDE



THE UNIVERSAL TOGGLE SKELETON

jsx

```
import React, { useState } from 'react';

function ToggleSkeleton() {
  // 🎯 STEP 1: CHOOSE YOUR STATE NAME
  const [isOn, setIsOn] = useState(false);

  // 🎯 STEP 2: BUTTON TEXT (Change these)
  const buttonText = isOn ? 'TURN OFF' : 'TURN ON';
  const statusText = isOn ? 'ON' : 'OFF';

  // 🎯 STEP 3: CONTENT TO SHOW/HIDE (Change this)
  const contentToShow = isOn ? <p>Content is visible!</p> : null;

  return (
    <div>
      <button onClick={() => setIsOn(!isOn)}>
        {buttonText}
      </button>
      <p>Status: {statusText}</p>
      {contentToShow}
    </div>
  );
}
```



QUICK TRANSFORMATION EXAMPLES

Show/Hide Message

jsx

```
const [isVisible, setIsVisible] = useState(false);
const buttonText = isVisible ? 'HIDE' : 'SHOW';
const contentToShow = isVisible ? <p>Secret message!</p> : null;
```

Dark/Light Mode

jsx

```
const [isDark, setIsDark] = useState(false);
const buttonText = isDark ? 'LIGHT' : 'DARK';
const contentToShow = <p>Mode: {isDark ? 'Dark' : 'Light'}</p>;
```

Favorite Button

jsx

```
const [isFavorited, setIsFavorited] = useState(false);
const buttonText = isFavorited ? '❤️' : '💔';
const contentToShow = <p>{isFavorited ? 'Favorited!' : 'Not favorited'}</p>;
```

INTERVIEW CHEAT SHEET

Problem	State Name	Button Text	Content
ON/OFF	<code>isOn</code>	ON/OFF	Status text
Show/Hide	<code>isVisible</code>	SHOW/HIDE	Your content
Dark/Light	<code>isDark</code>	DARK/LIGHT	Mode indicator
Expand	<code>isExpanded</code>	EXPAND/COLLAPSE	Detailed content
Favorite	<code>isFavorited</code>	❤️ / 🤍	Favorite status

⚡ 2 WAYS TO WRITE TOGGLE FUNCTIONS

Method 1: Inline (Your Skeleton)

jsx

```
<button onClick={() => setIsOn(!isOn)}>
  {isOn ? 'ON' : 'OFF'}
</button>
```

Method 2: Separate Function (Their Version)

```
jsx

function handleClick() {
  setIsOn(!isOn);
}

<button onClick={handleClick}>
  {isOn ? 'ON' : 'OFF'}
</button>
```

BOTH METHODS WORK THE SAME WAY!



COMMON ERRORS & FIXES

× State not updating?

```
jsx

// WRONG: Direct mutation
isOn = true;

// RIGHT: Use setter function
setIsOn(true);
```

× Button not working?

```
jsx

// WRONG: Missing function call
<button onClick={handleClick}>

// RIGHT: Add parentheses or use arrow function
<button onClick={() => handleClick()}>
```

× Content not showing?

jsx

```
// WRONG: Using if/else in JSX
{if (isOn) return <div>Content</div>}

// RIGHT: Use && or ternary
{isOn && <div>Content</div>}
```

🎯 BATTLE PLAN FOR INTERVIEWS

1. **IDENTIFY:** "Is this switching between two states?" → TOGGLE PATTERN
2. **CHOOSE STATE NAME:** Match to problem (isVisible, isDark, isOn, etc.)
3. **COPY SKELETON:** Use the universal skeleton
4. **CHANGE 3 THINGS:**
 - State variable name
 - Button text
 - Content to show
5. **TEST:** Click button to verify toggle works
6. **SUBMIT:** Done!

💡 PRO TIPS

- **Start simple** - get basic toggle working first
- **Use descriptive names** - `isModalOpen` vs `isOpen`
- **Keep returns clean** - put logic above return statement
- **Talk through your code** - explain your thinking
- **Both function styles are valid** - adapt to what they give you

FINAL CHECKLIST

- Component imports React and useState
 - State variable properly declared
 - Button has onClick handler
 - Button text changes based on state
 - Conditional content shows/hides correctly
 - No console errors
 - Basic functionality works
-

REMEMBER:

You only need to change 3 things:

1. **State variable name**
2. **Button text**
3. **Content to show**

The return statement stays exactly the same!

You've mastered the toggle pattern! Now you can solve ANY toggle question! 🎉