

Cheat Sheet: Understanding Function Components with Array and DOM Manipulation

Function Component with Array and DOM	Description	Code Example
		<pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10 11. 11</pre>
Function Component with function keyword	Function component starts with function keyword along with name of the component and includes html tags within return. It also exports component name by default	<pre>1. import React from 'react' 2. 3. function Extra() { 4. return (5. <> 6. <p>This is paragraph</p> 7. </> 8.) 9. } 10. 11. export default Extra</pre> <div>Copied!</div>
		<pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9</pre>
Function Component with arrow function	Function component starts with variable type along with name of the component and includes html tags within return. It also exports component name by default	<pre>1. import React from 'react' 2. 3. const Extra = () => { 4. return (5. <div>Extra</div> 6.) 7. } 8. 9. export default Extra</pre> <div>Copied!</div>
		<pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10 11. 11 12. 12</pre>
Props in function component	Props can be sent from parent component as attribute along with child component	<pre>1. import React from 'react' 2. import ChildComponent from './ChildComponent' 3. 4. function ParentComponent () { 5. let title='Project Manager'; 6. return (7. <></pre>

Access Props within child function component

Props can be accessed easily within the child function component using `props.variable_name`

Event handling in class component

Events such as click event can be performed by calling function which is declared before return of function component

State management in function component

State management can be done easily with `useState()` hook

about:blank

```
8.   <ChildComponent title={title}/>
9.   </>
10. )
11. }
12. export default ParentComponent
```

Copied!

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
```

```
1. import React from 'react'
2.
3. const ChildComponent = (props) => {
4.   return (
5.     <>
6.       <p>The title is {props.title}</p>
7.     </>
8.   )
9. }
10.
11. export default ChildComponent
```

Copied!

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
```

```
1. import React from 'react'
2. const Extra = (props) => {
3.   function show(){
4.     console.log('Show function');
5.   }
6.   return (
7.     <>
8.       <p>The title is {props.title}</p>
9.       <button onClick={()=>show()}>Click Here</button>
10.     </>
11.   )
12. }
13. export default Extra
```

Copied!

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
```

Array Declaration

Array can be declared in square brackets

Stateful Array

Array can be declared using useState

Dynamically Constructed Arrays

Arrays can be constructed dynamically based on application logic or received data

Array map() method

The map() method is commonly used to iterate over each element of an array and return a new array of React elements

for...of Loop

You can use the for...of loop to iterate over the elements of an array:

Rendering a List of Items

You can render a list of items by mapping over an array and returning a JSX element for each item

about:blank

12. 12

```
1. import React, { useState } from 'react'
2.
3. const StateManagement = () => {
4.   const [name, setName]=useState('John');
5.   return (
6.     <>
7.     <h1>State Management using useState</h1>
8.     <p>The name is {name}</p>
9.     </>
10.  )
11. }
12. export default StateManagement
```

Copied!

1. 1

```
1. const names = ['Alice', 'Bob', 'Charlie'];
```

Copied!

1. 1

```
1. const [todos, setTodos] = useState(['Learn React', 'Build Project']);
```

Copied!

1. 1
2. 2
3. 3
4. 4

```
1. const numbers = [];
2. for (let i = 0; i < 10; i++) {
3.   numbers.push(i);
4. }
```

Copied!

1. 1
2. 2
3. 3

```
1. const items = ['Apple', 'Banana', 'Orange'];
2. const itemList = items.map((item, index) => <li key={index}>{item}</li>);
3. return <ul>{itemList}</ul>;
```

Copied!

1. 1
2. 2
3. 3
4. 4
5. 5

```
1. const items = ['Apple', 'Banana', 'Orange'];
2. for (const item of items) {
3.   console.log(item);
4. }
5.
```

Copied!

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12

Adding and removing items in array You can add or remove items from an array using state and React's setState method

```
13. 13
14. 14
15. 15
16. 16
17. 17

1. import React from 'react';
2. function ArrayComponent() {
3.   const items = ['Autumn', 'Spring', 'Summer', 'Winter'];
4.   return (
5.     <div>
6.       <h1> Seasons Names</h1>
7.       <ul>
8.         {items.map((item, index) => (
9.           <li key={index}>{item}</li>
10.        ))}
11.       </ul>
12.     </div>
13.   );
14. }
15. export default ArrayComponent;
16.
17.

Copied!

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
26. 26
27. 27
28. 28
29. 29
30. 30
31. 31
32. 32
33. 33
34. 34
35. 35
36. 36

1. import React, { useState } from 'react';
2. function MyComponent() {
3.   const [items, setItems] = useState(['Autumn', 'Spring', 'Winter', 'Summer']);
4.   const [inputValue, setInputValue] = useState('');
5.
6.   const addItem = () => {
7.     setItems([...items, inputValue]);
8.     setInputValue('');
9.   };
10.
11.   const removeItem = (index) => {
```

Conditional rendering using ternary operator

You can conditionally render components based on the content of an array

inline style in react

Inline style can be applied within the tag as an attribute within double curly braces

about:blank

```
12.   const newItems = [...items];
13.   newItems.splice(index, 1);
14.   setItems(newItems);
15. };
16.
17. return (
18.   <div>
19.     <h1>Fruits</h1>
20.     <ul>
21.       {items.map((item, index) => (
22.         <li key={index}>
23.           {item}
24.           <button onClick={() => removeItem(index)}>Remove</button>
25.         </li>
26.       ))}
27.     </ul>
28.     <input
29.       type="text"
30.       value={inputValue}
31.       onChange={(e) => setInputValue(e.target.value)}
32.     />
33.     <button onClick={addItem}>Add</button>
34.   </div>
35. );
36. }
```

Copied!

- 1. 1
- 2. 2
- 3. 3
- 4. 4
- 5. 5
- 6. 6
- 7. 7
- 8. 8
- 9. 9
- 10. 10
- 11. 11
- 12. 12
- 13. 13
- 14. 14
- 15. 15
- 16. 16
- 17. 17
- 18. 18
- 19. 19
- 20. 20

```
1. import React, { useState } from 'react';
2.
3. function ArrayComponent() {
4.   const [items, setItems] = useState(['React JS', 'Vue JS', 'Angular JS', 'Vanilla JS']);
5.   return (
6.     <div>
7.       <h1>Front End Languages</h1>
8.       {items.length > 0 ? (
9.         <ul>
10.          {items.map((item, index) => (
11.            <li key={index}>{item}</li>
12.          ))}
13.        </ul>
14.      ) : (
15.        <p>No Front End language is available</p>
16.      )}
17.     </div>
18.   );
19. }
20. export default ArrayComponent;
```

Copied!

- 1. 1
- 2. 2
- 3. 3

Style using object

Style can be applied as an object like inline style

about:blank

```
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11

1.
2. import React from 'react';
3. function MyComponent() {
4.   return (
5.     <div style={{ backgroundColor: 'lightblue', padding: '20px', borderRadius: '5px' }}>
6.       <p style={{ color: 'white', fontSize: '18px' }}>This is a paragraph with inline styles.</p>
7.     </div>
8.   );
9. }
10. export default MyComponent;
11.
```

Copied!

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
```

```
1.
2. import React, { useState } from 'react';
3. function ToggleMessage() {
4.   const [isVisible, setIsVisible] = useState(true);
5.   const toggleVisibility = () => {
6.     setIsVisible(!isVisible);
7.   };
8.   const messageStyle = {
9.     display: isVisible ? 'block' : 'none',
10.    color: 'green',
11.    fontSize: '18px',
12.    marginTop: '10px'
13.  };
14.   return (
15.     <div>
16.       <h2>Toggle Message</h2>
17.       <button onClick={toggleVisibility}>
18.         {isVisible ? 'Hide Message' : 'Show Message'}
19.       </button>
20.       <p style={messageStyle}>This is a hidden message.</p>
21.     </div>
22.   );
23. }
24.
```

Copied!



Skills Network