

Quiz Questions | Answers

Module 7 | Lesson 7.3 – The useEffect Hook

1. What are the two arguments the **useEffect** Hook can accept?

- A:** A callback effect function and a dependencies list that limits when the effect is fired to certain conditions.
- B:** A callback effect function and a single state value that limits the effect to fire only when said state value has changed.
- C:** A callback effect function and a cleanup function that is run before the intended effect function.
- D:** The `useEffect` hook only accepts a single callback effect function.

Answer: A - A callback effect function and a dependencies list that limits when the effect is fired to certain conditions.

2. When does an effect callback run when no dependency is provided in the **useEffect** Hook dependencies list?

- A:** Only when the component mounts.
- B:** On every completed render.
- C:** Only when the component un-mounts.
- D:** When the component mounts and when any component state value is changed.

Answer: B - On every completed render.

3. When do we expect to see the “Message 2” statement in our console, in the following **useEffect** example?

```
useEffect(() => {  
  console.log("Message 1");  
  
  return () => {  
    console.log("Message 2");  
  };  
});
```

- A:** Only when the component un-mounts.
- B:** Only when the component mounts.
- C:** When the component mounts and un-mounts.
- D:** Before the intended effect is run (i.e. “Message 1” statement is logged) and when the component un-mounts.

Answer: D - Before the intended effect is run (i.e. “Message 1” statement is logged) and when the component un-mounts.

4. If an effect depends on a component value that is prone to change (e.g. state, props), the value should be provided in the effect’s dependencies list to prevent the effect from referencing stale values from previous renders.

- A:** True
- B:** False

Answer: A - True