

# New React Hooks

Created by :  
Sangeeta Joshi

# What are new hooks(16.8)

- Hooks are a new addition in React 16.8.
- They let you use state and other React features without writing a class.
- Hooks are backwards-compatible

# What is a Hook?

- Hooks are functions that let you “*hook into*”
  - React state
  - and lifecycle featuresfrom *function components*.
- Hooks don’t work inside classes — they let you use React without classes.
- React provides a few built-in Hooks
- We can also create our own Hooks to reuse stateful behavior between different components.

# State Hook: useState()

- useState is a hook
- to add some local state to function component
- useState returns a pair:
  - the current state value
  - and a function that lets us update it.
  - We can call this function from an event handler or somewhere else.
- The only argument to useState is the initial state
- The initial state argument is only used during the first render.

# Effect Hook : useEffect()

## Side Effects :

- Some operations cause side effects
  - Data fetching
  - Subscriptions
  - Manually changing the dom from react component
- Such operations can not be done during rendering as they may affect other components.
- These operations are called as side effects or just effect.

# Effect Hook : useEffect()

- The Effect Hook, useEffect :
  - adds the ability to perform side effects from a function component.
  - It serves the same purpose as
    - componentDidMount
    - componentDidUpdate,
    - componentWillUnmountin React classes, but unified into a single API.

# Effect Hook : `useEffect()`

- `useEffect` :
  - We are telling React to run “effect” function after flushing changes to the DOM.
  - Effects are declared inside the component so they have access to its props and state.
  - By default, React runs the effects after every render including the first render.
  - Effects may also optionally specify how to “clean up” after them by returning a function