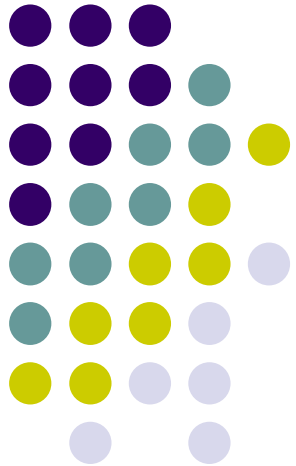
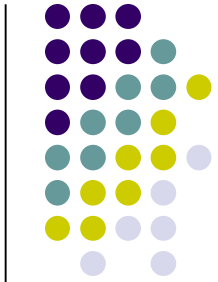
React



REACT ES6

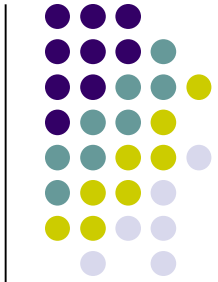


- ES6 stands for ECMA Script 6 version
- ECMAScript is the standardization of Javascript programming language.
- Use of ES6 features we Write Less and Do More.

Some of the new features like:

1. Classes
2. Arrow Functions
3. Variables (let, const, var)

Features of ES6



• CLASS - A class is used to bind data as well as methods together as a single unit. Object acts like a variable of the class.

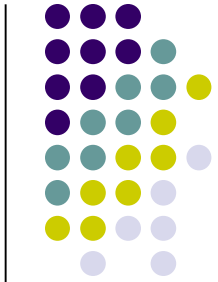
• ARROW- To write a shorter syntax for the function we use ARROW.

• VAR – If we use outside the function its called global variable. If we use inside the function its called local variable.

• CONST - Once we assign a variable as constant we never change it.

• LET - If you use var inside a for loop or any other block, the variable also available outside of that block or loop, So we use "LET" for overcoming from this problem.

If we use variable as let in block or loop, then the value only available inside the block only 2.



WHAT IS JSX?

- JSX stands for JavaScript XML.
- JSX allows us to write HTML in React.
- JSX makes it easier to write and add HTML in React.

```
const element = <h1>Hello, world!</h1>;
```

- This funny tag syntax is neither a string nor HTML.
- It is called JSX, and it is a syntax extension to JavaScript.
- JSX may remind you of a template language, but it comes with the full power of JavaScript.
- JSX produces React "elements".

CODING JSX

JSX allows us to write HTML elements in JavaScript and place them in the DOM without any createElement() and/or appendChild() methods.

JSX converts HTML tags into react elements.

WITH JSX

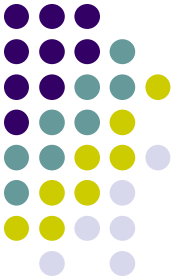
```
const myelement = <h1>I Love JSX!</h1>;

ReactDOM.render(myelement, document.getElementById('root'));
```

WITHOUT JSX

```
const myelement = React.createElement('h1', {}, 'I do not use JSX!');

ReactDOM.render(myelement, document.getElementById('root'));
```



EXPRESSIONS IN JSX

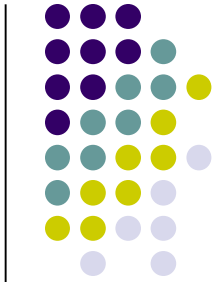
With JSX you can write expressions inside curly braces { }.

The expression can be a React variable, or property, or any other valid JavaScript expression.

JSX will execute the expression and return the result:

```
const myelement = <h1>React is {5 + 5} times better with JSX</h1>;
```

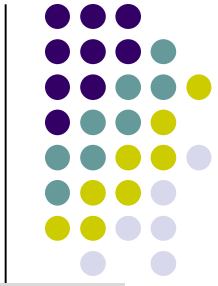
React Components



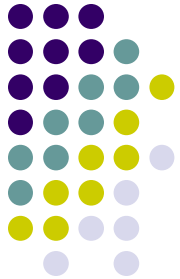
- Every application you will develop in React will be made up of pieces called components.
- Components make the task of building UI's much easier. We have lot of individual components like a single web page contain (search bar , menu bar , nav bar, content, article etc;)
- Merge all of these individual components to make a parent component which will be the final UI.

Two types of components

1. Functional Component
2. Class Component



Functional Component	Class Component
• Functional Components is a plain JavaScript, you do not have a choice to set the state in functional component. • There is no render function we are using in functional components. • Functional components only accept the props as an argument. • Functional components are sometimes called stateless components.	• Class components we have a feature to set the set state in component. • In class components, we have a render function which is use to return the react elements. • In class components, we have both options use the props and set the state also. • Class components are sometimes called stateful components.



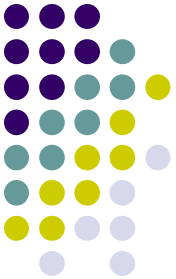
Class-based vs Functional Components

class-based	Functional
class XY extends Component	const XY = props => { ... }
✓ Access to State	✓ Access to State (useState())
✓ Lifecycle Hooks	✗ Lifecycle Hooks
Access State and Props via "this"	Access Props via "props"
this.state.XY & this.props.XY	props.XY
Use if you need to manage State or access to Lifecycle Hooks and you don't want to use React Hooks!	Use in all other Cases

Arrow Function

```
Function myfunction() {
.....
}
```

```
const myfunction = () => {
.....
}
```



- React ES6
- JSX
- Components
- Class based
- Functional Based
- Arrow Function


QUERIES