

1. Creating and using Components

We can create reusable components based on the website and interface. we can also use one component in another.

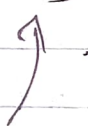
eg:

```
function App() {
```

```
  return (
```

```
    <Pizza />
```

```
  );
```



```
function Pizza() {
```

```
  return (
```

```
    source code ...
```

```
  );
```

```
);
```

2. JSX

JSX allows us to write JavaScript inside a HTML element.

eg:

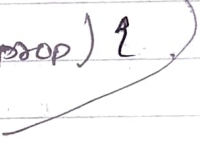
```
<p> The time now is {new Date().toLocaleTimeString} </p>
```

3. Passing and receiving props.

When we want to use the data in one component in another component we can use props.

```
eg: function pizzamenu() {  
  return <pizza name = spinach />  
}
```

```
function pizza (prop) {  
  return <  
    <h2> props.name </h2>  
  >
```



4. Rendering Lists :

It is used to render the data into the array ~~and~~ to the webpage.

We can use `map()` to get each of the ~~data~~ elements in the array and render it in the webpage.

8.

5. Conditional rendering (if :)

using `if` operator, we can give 2 condition, if the first condition is true, it will ~~execute~~ execute the second condition. if the first one fails it will return null.

Unary :

$x > 0$? it's greater! a lessor.

It has 3 parts first the condition, then it will execute the second part, else it will execute the 3rd part.

Restructuring props :

Instead of using props we can pass the data the we need to use in the function parameter to use it in the current function.

Function order (closure) (

) :

↓

data of another component

React fragments :

These fragments do not produce any visible element in DOM

< / >

< / >