

# Information Technologies for Industrial Engineers

เทคโนโลยีสารสนเทศสำหรับวิศวกรอุตสาหกรรม

# React Basics

**JSX**

App.tsx

```
import './App.css';

function App() {
  const msg = "This is from a variable.";
  const num = 0;
  return (
    <div>
      <div>This is just like HTML.</div>
      <div>{msg}</div>
      <div>Calculation: {num + 1}</div>
      <div>Call function: {sum(1, 2)}</div>
    </div>
  );
}

export default App;
```

App.tsx (Below)

```
function sum(a: number, b: number) {  
  return a + b;  
}
```

# Style

index.css

```
body {  
  margin: 0;  
  padding: 0;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

app.css

```
.wrapper {  
  margin: 0.5em auto;  
  width: 95vh;  
}  
  
.header {  
  color: teal;  
}
```

App.tsx

```
function App() {  
  // ...  
  return (  
    <div className="wrapper">  
      // ...  
      <div className="header">Test CSS</div>  
    </div>  
  );  
}
```

- Note the keyword `className`



# Event

App.tsx

```
return (  
  // ...  
  <button onClick="{handleClick}">Click</button>  
);  
  
function handleClick() {  
  alert("Hello!");  
}
```

- Note the keyword `onClick`

# Component

- Components are foundation upon which you build user interfaces (UI).
- Components are functions that return `JSX` (called React Nodes).

./src/Nav.tsx

```
function Nav() {  
  return (  
    <div className="nav">  
      <span>🔥</span>  
      <span>Home</span>  
      <span>About Me</span>  
    </div>  
  );  
}  
  
export default Nav;
```

App.tsx

```
import Nav from "../Nav"; ➡  
  
function App() {  
  return (  
    <>  
      <Nav /> ➡  
      <div className="wrapper">// ...</div>  
    </>  
  );  
}
```

- Notice the use of `<>...</>`

## App.css

```
.nav {  
  display: flex;  
  align-items: center;  
  gap: 0.5em;  
  color: teal;  
  background-color: rgb(243, 243, 243);  
  padding: 0.2em 0.5em;  
}
```

# Counter app

```
function App() {  
  let counter = 0;  
  function handleClick() {  
    counter += 1;  
    console.log(counter);  
  }  
  return (  
    <>  
      <div>{counter}</div>  
      <button onClick={handleClick}>Add</button>  
    </>  
  );  
}
```

# Why doesn't this work?

- [React doc](#)

# useState hook (function)

```
import { useState } from "react";

function App() {
  const [counter, setCounter] = useState(0); ➡
  function handleClick() {
    setCounter(counter + 1); ➡
    console.log(counter);
  }
  return (
    // ...
  );
}
```

- Code



## Common mistake

- Do not put `setCounter` in the main body.

# Form app

- Use `htmlFor` in `<label>`
- Type of event in an even handler is `ChangeEvent<HTMLInputElement>`
- Use `e.target.id` to determine the input clicked.
- use `e.target.value` to get the typed value.

## Code

# Todo app

- Use `interface` to specify object shape.
- Display a list of items using `map` array method.
  - Two ways of writing the arrow function.
- Use `todoInput` state to capture use input.
- Use `value={todoInput}` to sync the value of the form to the value of the state.
  - Controlled input form.
- Prevent adding item when the input is blank

## Code