useEffect





```
dependency array to contain dependencies that when changed trigger the effect to run

useEffect (effect, deps);

function that is to run imperative and likely effectful code
```

Run effect on every render

```
import React, { useEffect } from "react";

export const FunctionComponent = () => {
   useEffect(() => {
      console.log("run for every component render");
   });
};
```

function passed into the hook is the "effect" that is to run on every render

Run effect only on first render

```
import React, { useEffect } from "react";

export const FunctionComponent = () => {
   useEffect(() => {
      console.log(
        "run only for first component render (i.e. component mount)"
      );
   }, []);
};
```

passing an empty array as the dependencies list tells React to never re-run the effect.

effect will only run on first component render (i.e. component mount)

Run effect on first render and re-run when dependency changes

```
import React, { useEffect } from "react";

export const FunctionComponent = () => {
   useEffect(() => {
      console.log(
        "run for first component render and re-run when 'dependency' changes"
      );
   }, [dependency]);
};
```

passing a dependency in the dependencies list (e.g. a prop or state value) will tell React to re-run the effect every time the dependency changes

Run effect with cleanup

a function returned from the effect callback is the effect cleanup.

the effect cleanup runs before the effect of the next render is to run or when the component unmounts