



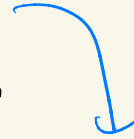
Class - 3

React

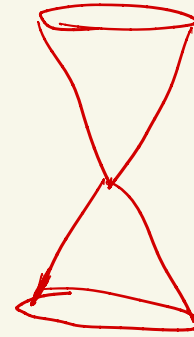
Debouncing - It's an optimisation technique which reduces function calls based on user actions.

"i" at $t = 0$

"ip" at $t < 200$ → ✓
 > 200



it will cancel prev
api call



api("i")

200ms

use Ref

use State

Both are used to Create Variables in React

Changing any ref
variable won't
cause any re-render

Changing any state variable
use setState fn leads
to re-render

when we don't want
to show variable data
on DOM / screen.

when we want to
show variable data
on screen.

. then why use ref var use simple var
then

??

```
function Card() {  
  const a = useRef(10);  
  a.current = a.current + 1;  
  return <div> this is a card </div>  
}
```

create use Prev hook?

```
export const useDebounce = (fn, delay) => {  
  const id = useRef(null);  
  const debouncedFn = useCallback(  
    (...args) => {  
      clearTimeout(id.current);  
      id.current = setTimeout(() => {  
        fn(...args);  
      }, delay);  
    },  
    [fn, delay]  
  );  
  return debouncedFn;  
};
```

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

```
const Stopwatch = () => {
```

```
  const [count, setCount] = useState(0);
```

```
  setInterval(() => {
```

```
    setCount(count + 1);
```

```
  }, 1000);
```

```
  return <div>{count}</div>;
```

```
};
```

```
export default Stopwatch;
```

setInterval()

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

```
const Stopwatch = () => {
```

```
  const [count, setCount] = useState(0);
```

```
  useEffect(() => {
```

```
    setInterval(() => {  
      |   setCount(count + 1);  
    }, 1000);
```

```
  }, []);
```

```
  return <div>{count}</div>;
```

```
};
```

```
export default Stopwatch;
```

count \rightarrow 0

() \Rightarrow }

setInterval(() \Rightarrow {

setCount(count + 1);

, 1000);

}