Debouncing & Throttling

Debouncing

=> To limit the amount of API calls we use debouncing because their are some API that takes money in each API call.

-> Amazon => Search bar => Each key input & it calls the API at each time when key is pressed.

=> Flipkart -> Search bar => Wait for some time after user input the search bar, then flipcart calls API.

=> You send the fetch request only when user stops typing.

```
<script>
  let timer;
  document.getElementById('searchInput').addEventListner('input', function(e){
      searchMovie(event.target.value)
      clearTimeout(timer)
      timer = setTimeout(() => {
                          console.log("The timer ended & API is called")
                          }, 10000)
  })
  function searchMovie(query){
                          console.log("Making API request" query)
  }
<script
```

Throttling

=> In a game, when user clicks on jump btn, the character jump, but if you click jump btn continiously, there is no way that game char jump that much time. Each jump takes some time to complete then and only then he will jump another.

=> This is done by throttling

```
document.getElementById("buy btn").addEventListner('click', buyItem)

let canClick = true
function buyItem(){
    // console.log("Button Clicked")
    if (canClick){
        // console.log('
        console.log('Item brought...')
        canClick = false
        setTimeout(()=>{
            canClick = true
        },1000)
    }
}
```

Throttling

=> Limiting the no. of times a function is called.