

41장 타이머

41.1 호출 스케줄링

- 호출 스케줄링이란?
 - 함수를 명시적으로 호출하지 않고 일정 시간이 경과된 이후에 호출되도록 타이머 함수를 사용하는 것
- 타이머 함수는 호스트 객체
 - Node.js와 브라우저에서 모두 사용 가능
- js엔진은 단 하나의 실행 컨텍스트만 갖는다.
 - 두 가지 태스크 동시에 진행 X
 - 싱글 스레드
 - 고로 `setTimeout`과 `setInterval`은 비동기 처리 방식으로 동작

41.2 타이머 함수

1. `setTimeout` / `clearTimeout`

- 첫 번째 인수는 콜백 함수
- 두 번째 인수는 타이머 시간
 - 디폴트 값 : 0
- 세 번째 인수는 콜백 함수에 전달할 수 있는 인수
- 타이머가 만료되면 첫 번째 인수로 전달받은 콜백함수 실행
- 단 한 번만 실행되도록 호출 스케줄링
- `clearTimeout(timerId)`로 타이머 취소 가능 => 콜백 함수 실행 X

2. `setInterval` / `clearInterval`


- 인수는 `setTimeout`과 동일
- 타이머를 식별할 수 있는 고유한 타이머 id 반환
 - 브라우저환경인 경우에는 숫자
 - Node.js 환경에는 객체
- 반환한 타이머 id로 `clearInterval` 인수로 전달하여 타이머 취소 가능

0. 머릿말


- 디바운스와 스로틀은 무엇인가?
 - 디바운스와 스로틀은 짧은 시간 간격으로 연속해서 발생하는 이벤트를 그룹화해서 과도한 이벤트 핸들러의 호출을 방지하는 프로그래밍 기법
 - `Scroll, resize, input, mousemove`와 같은 이벤트
 - 과도하게 호출시 성능에 문제
 - 디바운스와 스로틀 구현에는 타이머 함수가 사용된다.

41.3 디바운스와 스로틀

1. 디바운스

- 디바운스란?
 - 짧은 시간 간격으로 이벤트가 연속해서 발생하면 이벤트 핸들러 호출 X => 일정 시간이 경과한 이후에 이벤트 핸들러가 한 번만 호출되도록 한다.
 - 짧은 시간 이벤트 발생 => 이벤트 그룹화 => 하나의 이벤트만 발생 (일정한 주기 X)
 - 마지막 한 번만 실행
 - ex) 텍스트 입력 필드에서 input 입력 이벤트, API호출 빠르게 많이 클릭시
 - 서버에 부담을 주는 불필요한 처리 방지
 - 일정 시간 동안 값을 이벤트를 하지 않으면 그 때서야 이벤트 실행
- 
- 디바운스 구현 코드

2. 스로틀

- 스로틀이란?
 - 짧은 시간 간격으로 이벤트가 연속해서 발생하더라도 일정 시간 간격으로 이벤트 핸들러가 최대 한 번만 호출되도록 하는 것
 - 이벤트 그룹화 => 일정한 주기마다 발생
 - 호출 주기 만들
 - 보통 `scroll` 이벤트에 사용
 - 무한 스크롤 구현에 유용하게 사용
 - 실무에서는 스로틀은 Underscore의 `throttle` 함수나 Lodash의 `throttle` 함수를 사용하는 것을 권장
- 
- 스로틀 구현 코드

스로틀과 디바운스의 차이점?

- 일정한 주기의 유무
- 스로틀은 일정한 주기마다 이벤트 실행을 보장
- 디바운스는 아무리 많은 이벤트가 발생해도 모두 무시하고 특정 시간사이에 어떤 이벤트도 발생하지 않았을 때 딱 한번만 마지막 이벤트를 발생시키는 기법