



Chap41 타이머

▼ 호출 스케줄링

함수를 명시적으로 호출하지 않고 일정 시간이 경과된 이후에 호출되도록 함수 호출을 예약하려고 타이머 함수를 사용하는 것

▼ 타이머 함수

자바스크립트 엔진은 싱글스레드로 동작하기 때문에 타이머 함수 `setTimeout`과 `setInterval`은 비동기 처리 방식으로 동작한다.

▼ `setTimeout` / `clearTimeout`

1. `setTimeout`

- 두번째 인수로 전달받은 시간(ms, 1/1000초)으로 단 한 번 동작하는 타이머를 생성한다.
- 타이머가 만료되면 첫 번째 인수로 전달받은 콜백 함수가 호출된다.

```
//func : 타이머가 만료된 뒤 호출된 콜백함수
//delay : 타이머 만료 시간. 생략한 경우 기본값 0
//param1,param2... : 호출 스케줄링된 콜백 함수에 전달해야 할
const timeoutId=setTimeout(func|code[,delay,param1,pa
```

- 예시

```
//1초 후 타이머가 만료되면 콜백함수가 호출된다.
setTimeout(()=>console.log('Hi'),1000);
```

- `setTimeout` 함수는 생성된 타이머를 식별할 수 있는 고유한 타이머 id를 반환한다. 이 id를 `clearTimeout` 함수의 인수로 전달하면 타이머를 취소할 수 있다.

2. `clearTimeout`

- 호출 스케줄링을 취소한다.

```
const timerId=setTimeout(()=>console.log('Hi'),1000);

//타이머가 취소되면 setTimeout함수의 콜백함수가 실행되지 않는다.
clearTimeout(timerId);
```

▼ setInterval / clearInterval

1. setInterval

- 두번째 인수로 전달받은 시간으로 반복 동작하는 타이머를 생성한다.
- 타이머가 만료될 때마다 첫 번째 인수로 전달받은 콜백 함수가 반복 호출된다.
- 타이머가 취소될 때 까지 계속된다.

```
//func : 타이머가 만료된 뒤 호출된 콜백함수
//delay : 타이머 만료 시간. 생략한 경우 기본값 0
//param1,param2... : 호출 스케줄링된 콜백 함수에 전달해야 할 인=
const timeoutId=setTimeout(func|code[, delay,param1,param2...], delay);
```

- setInterval 함수는 생성된 타이머를 식별할 수 있는 고유한 타이머 id를 반환한다. 이 id를 clearInterval함수의 인수로 전달하여 타이머를 취소할 수 있다.

2. clearInterval

- 호출 스케줄링을 취소한다.

```
let count=1;
const timeoutId=setInterval(()=>{
  console.log(count);

  //count가 5이면 setInterval함수가 반환한 타이머 id를 clearInterval함수의 인수로 전달하여 타이머를 취소한다. 타이머가 취소되면 setInterval함수의 콜백 함수가 호출되지 않는다.
  if(count++===5)clearInterval(timeoutId);
}, 1000);
```

▼ 디바운스와 스로틀

짧은 시간 간격으로 연속해서 발생하는 이벤트를 그룹화해서 과도한 이벤트 핸들러의 호출을 방지하는 프로그래밍 기법

▼ 디바운스

짧은 시간 간격으로 이벤트가 연속해서 발생하면 이벤트 핸들러를 호출하지 않다가 일정 시간이 경과한 이후에 이벤트 핸들러가 한 번만 호출되도록 한다.

▼ 스로틀

짧은 시간 간격으로 이벤트가 연속해서 발생하더라도 일정 시간 간격으로 이벤트 핸들러가 최대 한 번만 호출되도록 한다.