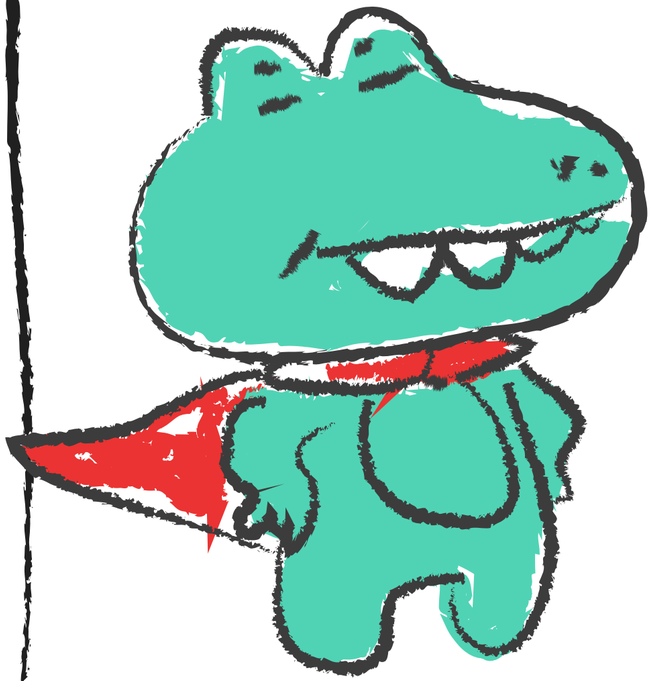
setTimeout과 SetInterval

호출 스케줄링



김민지

목차

1st setTimeout

2nd setInterval

3rd 스케줄링 취소

1. setTimeout

일정 시간이 경과한 후에 콜백 함수를 한 번 실행하는 데 사용

```
setTimeout(callback, delay);
```

- callback: 실행할 함수 또는 실행할 코드 블록
- delay: 실행을 지연할 시간(밀리초)

1. setTimeout

```
function greet() {
  console.log('안녕하세요!');
}

setTimeout(greet, 2000); // 2초 후에 '안녕하세요!' 출력
```

greet 함수가 2초 후에 출력

2. setInterval

일정한 시간 간격으로 콜백 함수를 반복해서 실행하는 데 사용

```
setInterval(callback, delay);
```

- callback: 실행할 함수 또는 실행할 코드 블록입니다.
- delay: 실행 간격(밀리초)입니다.

2. setInterval

```
let counter = 0;

function increment() {
  counter++;
  console.log(counter);
}

setInterval(increment, 1000); // 1초마다 counter 값 증가 및 출력
```

계속 실행되기 때문에
clearInterval 을 사용해 중지 시켜야함

2. setInterval

```
let counter = 0;

function increment() {
  counter++;
  console.log(counter);
}

setInterval(increment, 1000); // 1초마다 counter 값 증가 및 출력
```

계속 실행되기 때문에
clearInterval 을 사용해 중지 시켜야함

3. 스케줄링 취소

clearInterval 및 clearTimeout 함수에 전달하여 스케줄링된 작업을 취소

setInterval - clearInterval
setTimeout - clearTimeout

3. 스케줄링 취소

setTimeout - clearTimeout

```
const timeoutId = setTimeout(callback, delay);
clearTimeout(timeoutId);
```

```
function greet() {
  console.log('안녕하세요!');
}

const timeoutId = setTimeout(greet, 2000); // 2초 후에 '안녕하세요!' 출력

// 1초 후에 작업 취소
setTimeout(() => {
  clearTimeout(timeoutId);
}, 1000);
```

3. 스케줄링 취소

setInterval - clearInterval

```
const intervalId = setInterval(callback, delay);
clearInterval(intervalId);
```

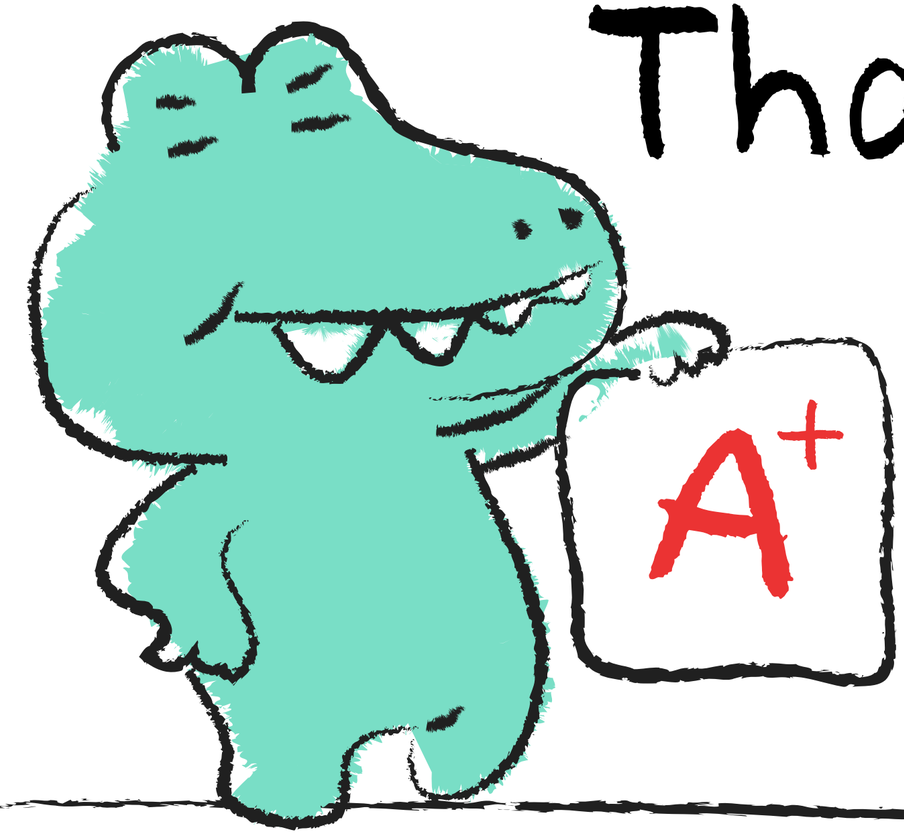
```
let counter = 0;

function increment() {
  counter++;
  console.log(counter);
}

const intervalId = setInterval(increment, 1000); // 1초마다 coun
ter 값 증가 및 출력

// 5초 후에 작업 중지
setTimeout(() => {
  clearInterval(intervalId);
}, 5000);
```

Finally,



Thank you for Watching!