

Javascript 이벤트 활용

[이벤트 \(https://www.w3schools.com/js/js_events_examples.asp\)](https://www.w3schools.com/js/js_events_examples.asp),
[this \(https://www.w3schools.com/js/js_this.asp\)](https://www.w3schools.com/js/js_this.asp),
[CSS 애니메이션 \(https://www.w3schools.com/css/css3_animations.asp\)](https://www.w3schools.com/css/css3_animations.asp)

그동안 몇 개의 강의를 통해 마우스, 키보드 이벤트 그리고 간략적인 이벤트에 대한 개념 등에 대해 이야기했습니다. 이번 장에서는 이벤트에 대한 마지막 내용을 정리해보도록 하겠습니다.

기타 이벤트

먼저, 이번 절에서는 사용자 행동에 대한 이벤트라기보다는 웹 페이지가 로드(Load)되면서 발생하는 이벤트에 대해서 알아보겠습니다.

메서드	의미
onload	페이지가 로드된 후 발생하는 이벤트/이미지가 로드된 후 발생하는 이벤트
onerror	이미지 로딩중 에러가 생길 경우에 발생하는 이벤트
onunload	브라우저에서 문서(웹 페이지)가 닫힐 경우 발생하는 이벤트
onresize	브라우저 크기가 조절될 때 발생하는 이벤트

위 이벤트들은 웹 페이지(Document) 자체와 관련이 있는 이벤트들입니다. 한 가지 특이한 점은 onload 이벤트는 웹 페이지가 로드된 후 발생하는 이벤트, 이미지가 정상 로드된 후 발생하는 이벤트로 나뉘지고, 같은 이름을 갖습니다.

아래 코드는 웹 페이지에서 사용되는 이미지 로딩과 관련된 이벤트에 대한 예시입니다.

```

```

위 코드는 태그에 onload 이벤트와 onerror 두 가지 이벤트를 인라인(속성으로 등록)으로 등록하여, 에러에 대한 alert 창을 띄우는 코드입니다. 일반적으로, 정상 로드된 경우엔 크게 처리를 해줄 필요가 없지만, 에러 발생 시 처리할 코드는 작성하는 것이 좋습니다.

나머지 이벤트는 직접 코드를 작성하고 확인해보세요! :)

this 키워드

자바스크립트 코드 작성을 위해선 꽤나 많은 키워드를 기억해야 합니다. 그 많은 키워드들 중에서 기억하기 쉽지만 다루기 상당히 까다로운? this라는 키워드가 있습니다. 강의에서 언급했듯이, 이벤트 코드에서 사용되면 이벤트가 발생한 엘리먼트 자기 자신을 지칭하는 키워드였습니다.

이 this 키워드는 객체(Object)와 function에서도 사용될 수 있는 키워드입니다. 이번 절에서는 간단히 그 내용을 정리하도록 하겠습니다.

결론부터 말하면, 모두 자기 자신을 가리킨다고 할 수 있습니다.

함수 안에서 사용되면, 함수의 소유를 나타내게 됩니다. 이게 무슨 말인지 코드를 보며 확인해보겠습니다.

```
function myFunction(){
  //1번 코드
  this.globalVar = "전역 변수!";
}
//2번 코드
myFunction();
//3번 코드
console.log(window.globalVar); //"전역 변수!"
```

위의 코드가 조금 생소하게 느껴질 수도 있습니다. 자바스크립트 함수에 대해서 이야기할 때 자바스크립트는 다른 프로그래밍 언어와 상당 부분 다른 문법 체계가 존재한다고 이야기했습니다. 다른 언어에도 이 `this`라는 키워드가 존재합니다. 하지만, 타 언어와는 전혀 다르게 동작하게 되는데요.

사실, 자바스크립트 함수에서 사용되는 `this` 키워드에 대한 얘기만으로 이번 장을 모두 채울 수 있을 정도로 많은 내용에 대한 정리가 필요하므로, 이번 절에서는 간단히 맛만 보도록 하겠습니다.

1번 코드는 함수 안에서 `this`라는 키워드를 붙이고 `닷(.)` 연산자를 이용해 어떤 `globalVar`라는 변수에 값을 할당하고 있습니다. 이렇게 코드를 작성하면 `myFunction`이 호출되는 2번 코드가 동작하는 순간에 해당 변수는 전역 변수가 됩니다. 여기서 말하는 전역(Global)이란, 최상위 객체인 `window` 객체의 속성 값처럼 `닷` 연산자(`.`)로 접근할 수 있다는 말이 됩니다. 신기하죠..?

함수에서 사용되는 `this`라는 키워드에 대해 더 궁금하시다면, [여기](https://github.com/FEDevelopers/tech.description/wiki/%EC%9E%90%EB%B0%94%EC%8A%A4%ED%EC%82%AC%EC%9A%A9%EB%90%98%EB%8A%94-this%EC%97%90-%EB%8C%80%ED%95%9C-%EC%84%A4%EB%AA%85-1#1-this%EC%97%90-%EB%8C%80%ED%95%9C-%EB%AF%B8%EC%8A%A4%ED%84%B0%EB%A6%AC) (<https://github.com/FEDevelopers/tech.description/wiki/%EC%9E%90%EB%B0%94%EC%8A%A4%ED%EC%82%AC%EC%9A%A9%EB%90%98%EB%8A%94-this%EC%97%90-%EB%8C%80%ED%95%9C-%EC%84%A4%EB%AA%85-1#1-this%EC%97%90-%EB%8C%80%ED%95%9C-%EB%AF%B8%EC%8A%A4%ED%84%B0%EB%A6%AC>)에 나온 글을 읽어보시는 걸 추천드립니다.

다음으로, 객체에서 사용되게 되면 객체 자기 자신을 나타내게 됩니다. 마찬가지로, 코드를 보며 확인해보겠습니다.

```
var user = {
  firstName: "Seongjae",
  lastName: "Moon",
  fullName: this.firstName + " " + this.lastName;
}
console.log(user.fullName); //Seongjae Moon
```

상대적으로 객체에서 사용된 `this` 키워드는 이해하기 쉽습니다. 우리가 이벤트에서 `this` 키워드를 이용해 이벤트가 발생한 엘리먼트 자신의 정보를 가져올 수 있다고 알아봤습니다.

객체를 선언하며 `this` 키워드를 사용하면, 객체 안에서 사용된 속성(Properties)을 지칭하게 됩니다. 마치 이벤트가 발생한 엘리먼트 자체를 지칭하는 것처럼 동작하게 할 수 있습니다.

CSS 애니메이션

이번 절에서는 강의에서 살펴봤던 `@keyframes` 키워드에 대해 조금 더 정리해보도록 하겠습니다. 우리는 역동적인 웹 페이지를 위해서 흔히 이야기하는 애니메이션 효과 등을 적용하고 싶을 경우가 있습니다.

이러한 애니메이션 효과를 주는 방법은 다양하게 있겠지만, 순수 Javascript 코드만으로 효과를 만들어내는 것보다는 다른 여러 기술들과 함께 조합해서 코드를 작성하면 같은 효과를 훨씬 효과적이고, 쉽게 작성할 수 있습니다.

그 방법 중 하나가 이번 절에서 살펴볼 @keyFrames 키워드입니다. 제목에서 느껴지듯 CSS 코드를 통해 애니메이션을 동작시키는 방법입니다. 동적으로 CSS style 코드를 변경시킬 수 있는 방법으로, 많은 웹 사이트가 적용하고 있는 것을 확인할 수 있는 방법이기도 합니다.

@keyFrames의 동작 방법은 간단합니다. @keyFrames 코드 안에 현재 스타일(from)과 특정 조건이 주어졌을 경우 변경될 스타일(to)을 지정해주는 형식입니다.

아래 코드는 간단한 @keyFrames 사용 예시입니다.

```
/* 기본 형식 */
@keyframes test {
  from {
    background-color: red;
  }
  to {
    background-color: powderblue;
  }
}
/* 애니메이션 적용 */
p#test {
  animation-name: test;
  animation-duration: 5s;
}
```

우선, 애니메이션을 적용할 태그를 특정합니다. 위 예제 코드에선 아이디 속성이 test인 <p> 태그에 애니메이션 코드를 적용하고 있습니다. from {css code} 구문에서 애니메이션 초기화를 진행하고, to {css code} 구문에서 애니메이션의 종료 형태를 작성해주면, 자동으로 애니메이션이 적용됩니다. 애니메이션이 동작하는 방식은 다양한 CSS 애니메이션 속성을 이용해 조절할 수 있습니다.

지금 당장은 애니메이션을 활용할 부분이 마땅히 없더라도, 기억해두셨다가 추후에 강의에서처럼 Javascript 코드를 통해 동적으로 css 속성을 변경하고 미리 작성된 CSS 애니메이션 코드를 적용시키는 형태로 활용하실 수 있었으면 좋겠습니다! :)

CSS 애니메이션에 대한 더 자세한 내용은 [여기 \(https://www.w3schools.com/css/css3_animations.asp\)](https://www.w3schools.com/css/css3_animations.asp)를 참고해주세요!

이벤트 코드를 비롯해서 자바스크립트를 정확히 잘 다루는 것은 어렵습니다. 다행히도, 이미 개발된 많은 라이브러리와 프레임워크 등을 통해서 많은 부분들을 원활하게 작성할 수 있습니다. 이와 관련된 더 자세한 내용은 다음 장에서 더 이야기하도록 하겠습니다. 이벤트에 대해 처음 이야기할 때 말씀드렸듯이, 처음에는 자바스크립트 기본 문법과 개념, 동작 방식 등에 초점을 맞추고 공부하고 하나하나 응용해나가시길 바라겠습니다. :)