

예외처리

예외(exception)

예외(exception)란 프로그램이 실행 중에 발생하는 런타임 오류를 의미합니다.

예외 발생

예외를 발생시킨다는 것은 명시적으로 오류를 발생시킨다는 의미뿐만 아니라 예외 상황을 알린다는 의미도 있습니다.

자바스크립트에서는 throw 키워드를 사용하여 예외를 발생시킬 수 있습니다.

문법

throw 표현식;

표현식에는 예외 코드를 나타내는 숫자나 오류 메시지를 담고 있는 문자열, Error 객체 등이 올 수 있습니다.

예외 처리(exception handling)

자바스크립트에서는 프로그램이 실행되는 도중 발생하는 예외를 처리하기 위해 try / catch / finally 문을 사용합니다.

```
//문법
try {
    예외를 처리하길 원하는 실행 코드;
} catch (ex) {
    예외가 발생할 경우에 실행될 코드;
} finally {
    try 블록이 종료되면 무조건 실행될 코드;
}
```

1. try 블록 : 기본적으로 맨 먼저 실행되는 코드이며, 여기에서 발생한 예외는 catch 블록에서 처리될 수 있습니다.

2. catch 블록 : try 블록에서 발생한 예외 코드나 Error 객체를 인수로 전달받아 그 처리를 담당합니다.
3. finally 블록 : 이 블록은 try 블록에서 예외가 발생하건 안 하건 맨 마지막에 무조건 실행됩니다.

Error 객체 런타임 오류가 발생할 때마다 Error 객체의 인스턴스가 만들어져 해당 오류의 정보를 저장합니다.

```
<h1>예외 발생</h1>
<button id="throwExBtn">예외 발생!</button>
<p id="text"></p>

<script>
// 아이디가 "throwExBtn"인 요소를 선택함.
var btn = document.getElementById("throwExBtn");
// 선택한 요소에 click 이벤트 리스너를 등록함.
btn.addEventListener("click", throwEx);
function throwEx() {
    try {
        // Error 객체를 사용해 명시적으로 오류를 발생시킴
        throw new Error("직접 발생시킨 오류입니다!");
        // 발생한 오류를 매개변수 ex로 접근할 수 있음
    } catch (ex) {
        // name 프로퍼티는 오류의 타입을 저장함.
        document.getElementById("text").innerHTML = ex.name
        + "<br>";
        // message 프로퍼티는 오류 메시지를 저장함.
        document.getElementById("text").innerHTML += ex.message;
    }
}
</script>
```