



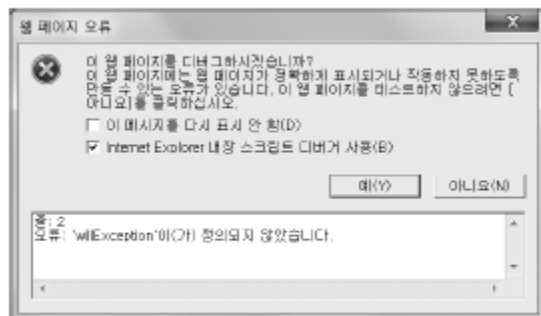
# 11 예외 처리

모던 웹을 위한 Javascript jQuery 입문

## ❖ 예외 처리Exception handling 란?

- 프로그램이 실행되는 동안 문제가 발생하면 프로그램 자동 중단
  - 프로그램이 대처할 수 있게 처리하는 것
- 예외Exception
  - 프로그램 실행 중 발생하는 오류
- 에러Error
  - 프로그래밍 언어의 문법적 오류

그림 11-1 웹 페이지 오류



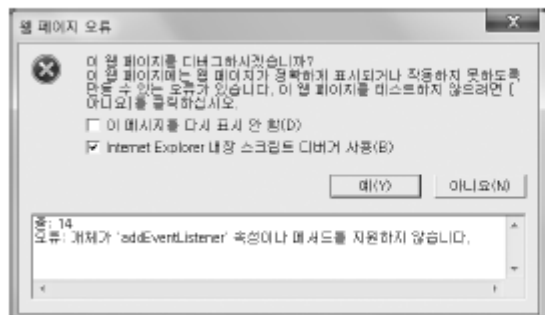
## ❖ 기본 예외 처리

- 예외가 발생하지 않게 사전에 해결하는 것

코드 11-1 이벤트 연결의 예외 처리

```
function registerEventListener(node, event, listener) {  
    if (node.addEventListener) {  
        // 파이어폭스, 크롬, 사파리, 오페라  
        node.addEventListener(event, listener, false);  
    } else if (node.attachEvent) {  
        // 인터넷 익스플로러  
        node.attachEvent('on' + event, listener);  
    }  
}
```

그림 11-2 익스플로러에서 addEventListener() 메서드를 사용했을 경우의 예외 발생



## ❖ 이벤트 객체의 예외 처리

- 브라우저 별 이벤트 객체 위치 다름
- 브라우저에 구애 받지 않는 방식으로 코드 작성
- 대부분의 예외 처리는 기본 예외 처리를 사용해 처리 가능

코드 11-2 이벤트 객체의 예외 처리

```
function whenClick(e) {  
    var event = window.event || e;  
    var willAlert = '';  
    willAlert += 'clientX: ' + event.clientX + '\n';  
    willAlert += 'clientY: ' + event.clientY + '\n';  
    alert(willAlert);  
}
```



## ❖ 고급 예외 처리

- try 키워드, catch 키워드, finally 키워드 사용해 예외 처리
  - try catch finally 구문
    - try 구문 안에서 예외가 발생하면 catch 구문에서 처리
    - finally 구문
      - » 필수 사항은 아님
      - » 예외 발생 여부와 상관없이 수행돼야 하는 작업이 있을 때 사용

```
try {  
  
} catch (exception) {  
  
} finally {  
  
}
```



## ❖ 예제 코드 11-3]

- willExcept라는 객체 없고 byeBye() 메서드도 존재하지 않음
  - 프로그램은 정말 'Bye'
  - 강제로 예외를 발생시킨 예제
  - try 구문 안에서 예외 발생
    - 더 이상 try 구문 진행하지 않고 catch 구문 실행

코드 11-3 try catch 구문의 사용

```
<script>
  try{
    willExcept.byeBye();
  } catch (exception) {

  }
</script>
```



## ❖ 예제 코드 11-4]

- willExcept.byeBye()를 실행하려는 순간 예외 발생
  - catch 구문을 실행
  - 첫 번째 경고창은 출력하지 않으며, 두 번째 경고창만 출력

코드 11-4 예외 처리

```
<script>
  try {
    willExcept.byeBye();
    alert('try 구문 종료');
  } catch (exception) {
    alert('예외가 발생했습니다.');
```

```
  }
</script>
```



## ❖ 예제 코드 11-5]

- try catch finally 구문 동작
  - try 구문에서 예외 발생하고 catch 구문 실행
  - try 구문에서 예외가 발생하든 안 하든 무조건 finally 구문 실행

코드 11-5 finally 구문

```
<script>
  try {
    willExcept.byeBye();
  } catch (exception) {
    alert('예외가 발생했습니다.');
```

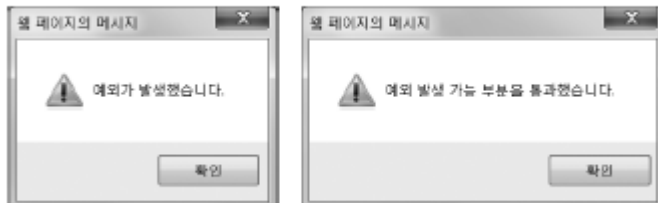
```
  } finally {
```

```
    alert('예외 발생 가능 부분을 통과했습니다.');
```

```
  }
```

```
</script>
```

그림 11-3 try catch finally 구문





## ❖ 예제 11-6]

- 고급 예외 처리 기법에서 이벤트 리스너 추가 하기
  - 파이어폭스, 크롬, 사파리, 오페라 브라우저라면 try 구문 실행
  - 인터넷 익스플로러라면 try 구문에서 예외 발생해 catch 구문 실행

코드 11-6 try catch 구문을 사용한 이벤트 연결

```
<script>
function registerEventListener(node, event, listener) {
    try{
        // 파이어폭스, 크롬, 사파리, 오페라
        node.addEventListener(event, listener, false);
    } catch(exception) {
        // 인터넷 익스플로러
        node.attachEvent('on' + event, listener);
    }
}

window.onload = function () {
    var header = document.getElementById('header');
    registerEventListener(header, 'click', function () {
        alert('Click');
    });
};
</script>
<body>
    <h1 id="header">Click</h1>
</body>
```



## ❖ 예외 객체란?

- try catch 구문을 사용할 때 catch의 괄호 안에 입력하는 식별자
- e나 exception이라는 식별자 사용

코드 11-7 예외 객체

```
try {  
  
} catch (exception) {  
  
}
```

표 11 -1 예외 객체의 속성

속성 이름	설명
message	예외 메시지
description	예외 설명
name	예외 이름



## ❖ 예제 코드 11-8]

- 자바스크립트는 배열의 크기가 한정돼 있으므로 예제는 예외 발생
  - 객체의 속성 출력

**참고** 자바스크립트의 배열의 크기는 4,294,967,295까지 가능합니다.

코드 11-8 예외 객체의 속성

```
<script>
  try {
    var array = new Array(9999999999999);
  } catch (exception) {
    var output = '';
    for (var i in exception) {
      output += i + ': ' + exception[i] + '\n';
    }
    alert(output);
  }
</script>
```

그림 11-4 인터넷 익스플로러의 실행 결과



# 11.4 에러와 예외

## ❖ 에러와 예외를 구분하는 가장 간단한 방법

- try catch 구문으로 해결할 수 있는 것이 예외
- 해결할 수 없는 것이 에러
- 예제 코드 11-9]

코드 11-9 오류

```
<script>
  try {
    error+-error+++;
  } catch (exception) {
    alert('Exception');
  }
</script>
```

그림 11-5 에러

```
try {
  error+-error+++;
} catch (exception) {
  alert('Exception');
}
```

✖ ExpAndErr.htm:7Uncaught SyntaxError: Unexpected token ;



# 11.5 예외 강제 발생

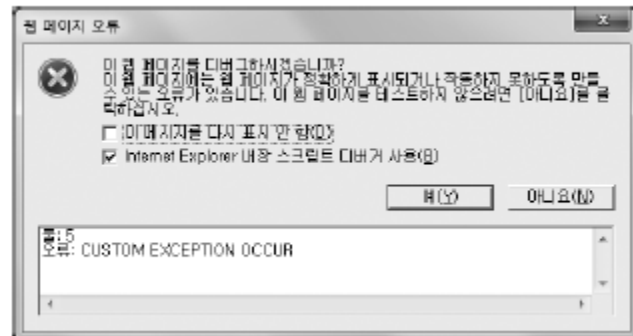
## ❖ 예외를 강제로 발생시켜야 할 때

- throw 키워드 사용
- 예제 코드 11-10]

코드 11-10 throw 키워드

```
<script>  
    throw 'CUSTOM EXCEPTION OCCUR';  
</script>
```

그림 11-6 예외의 강제 발생



## ❖ 예외 강제 발생 예제

- 객체 사용하는 사용자에게 주의를 줄 수도 있고 예외 관련 처리 가능
- 예제 코드 11-11]

코드 11-11 강제로 발생한 예외의 처리

```
<script>
  try {
    throw 'DivideByZeroException';
  } catch (exception) {
    if (exception == 'DivideByZeroException') {
      alert('ㅇㅇㅇ');
    }
  }
</script>
```



## ❖ 예제 코드 11-12]

- 코드 11-12의 divide() 함수는 0으로 대상을 나누지 못하게
- 자바스크립트는 특정 숫자를 0으로 나눈다고 해서 문제는 없음
  - 0으로 나누는 것을 막고 싶고, 발생하면 알려줄 수 있게 예외 강제 발생
  - divide() 함수를 실행할 때 예외 발생하므로 catch 구문 실행
    - 경고창에 문자열 ‘CATCH’ 출력

코드 11-12 적절한 강제적 예외 발생과 처리

```
<script>
// 함수를 선언합니다.
function divide(alpha, beta) {
    if (beta == 0) {
        throw 'DivideByZeroException';
    } else {
        return alpha / beta;
    }
}

// 출력합니다.
try {
    divide(10, 0);
} catch (exception) {
    alert('CATCH');
}
</script>
```





# Thank You !

모던 웹을 위한 Javascript jQuery 입문