

# 웹 앱 개발을 위한 JavaScript 기초 강의 노트

제 7회차

JavaScript의 내장 함수

### ■ 학습목표

- 대화상자와 타이머 함수를 이해하고 사용할 수 있다.
- 인코딩, 디코딩 함수와 eval() 함수를 이해하고 사용할 수 있다.
- 숫자 확인과 변환 함수를 이해하고 사용할 수 있다.

### ■ 학습내용

- 대화 상자와 타이머 함수
- 인코딩, 디코딩 함수와 eval() 함수
- 숫자 관련 함수

## 1. 대화 상자와 타이머 함수

### 1) 대화 상자 함수

#### - 내장 함수

- 자바스크립트에서 제공하는 함수
- 구현하고자 하는 기능과 일치하는 내장 함수 이용 시 쉽게 프로그래밍 가능

#### - 대화 상자 함수

- alert()
  - 사용자에게 메시지를 전달하기 위한 대화 상자 출력
  - 확인만 선택 가능
  - 사용 예 : alert('이미 존재하는 id입니다');
- prompt()
  - 사용자로부터 입력을 받기 위한 대화 상자 출력
  - '확인'과 '취소' 버튼 선택 가능
  - 사용 예 : prompt('이름을 입력하세요','홍길동');
  - '확인' 선택 시 : 입력 받은 문자열 리턴
  - '취소' 선택 시: null 리턴
- Confirm()
  - 사용자로부터 확인 받기 위한 대화 상자 출력
  - '확인'과 '취소' 버튼 선택 가능
  - '확인' 선택 시 : true 리턴, '취소' 선택 시: false 리턴
  - 사용 예 : confirm('계속 진행하시겠습니까?');

## 1. 대화 상자와 타이머 함수

### 1) 대화 상자 함수

- 대화 상자 함수

- 예

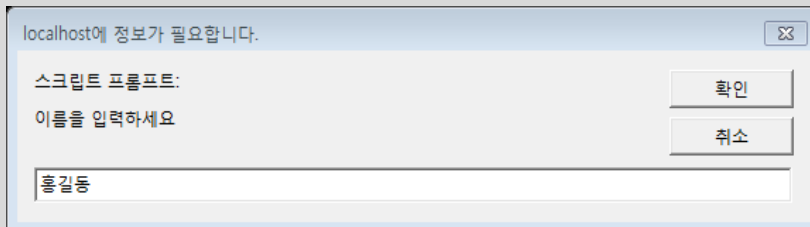
```
<script>

var name = prompt('이름을 입력하세요', '홍길동');
alert(name);

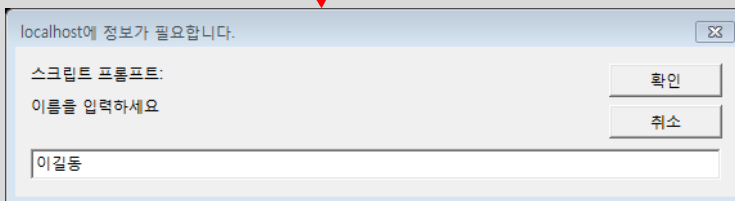
var go = confirm('계속 진행하시겠습니까?');
if (go) {
    alert('진행하겠습니다. ');
} else {
    alert('진행을 멈추겠습니다. ');
}

</script>
```

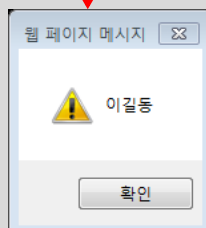
- 결과



↓ 이름 수정



↓ '이길동'으로 수정 후 '확인' 선택 변수 name 저장된 문자열 출력



## 1. 대화 상자와 타이머 함수

### 2) 타이머 함수

#### - 타이머 함수

- 특정한 시간에 특정한 함수를 실행할 수 있게 하는 함수
- 함수와 시간을 매개변수로 이용
- 1000 millisecond = 1초
- 함수의 종류

함수	설 명
setTimeout(function, millisecond)	일정 시간이 지난 후 함수를 한 번 실행
clearTimeout(id)	일정 시간이 지난 후 함수를 한 번 실행하는 것을 중지
setInterval(function, millisecond)	일정 시간 간격으로 함수를 반복해서 실행
clearInterval(id)	일정 시간 간격으로 함수를 반복해서 실행하는 것을 중지

\* 표 출처: 윤인성.(2013). 모던 웹을 위한 JavaScript jQuery 입문. 한빛미디어

- setTimeout() 함수를 실행하기 전에 타이머를 멈추게 하기

```
<script>
  var setid = setTimeout(function () {
    alert('test');
  }, 3000);

  clearTimeout(setid);
</script>
```

→ 더 많이 사용하는 형태

setInterval() 이용해 일정 시간 간격으로 특정한 기능을 수행하도록 함

→ setTimeout()를 이용하여 반복 실행을 멈추게 함

## 2. 인코딩, 디코딩 함수와 eval() 함수

### 1) 인코딩과 디코딩 함수

#### - 인코딩(encoding)

- 정보의 형태나 형식을 변환하는 처리 방식
- 문자를 컴퓨터에 저장하거나 통신에 사용하기 위해 부호화하는 방법
- 웹 상에서 통신 시 한글 등의 유니코드의 오작동 방지 위해 사용
- 예 : 웹 검색 시 퍼센트(%)와 알파벳 대문자 조합의 주소
- 함수의 종류

함수	설 명
escape()	<ul style="list-style-type: none"> <li>• 영문 알파벳과 숫자, 일부 특수 문자(@, *, -, ., +, . . . /)를 제외하고 모두 인코딩</li> <li>• 1바이트 문자는 %XX의 형태로, 2바이트 문자는 %uXXXX 형태로 변환</li> <li>• 예)                         <ul style="list-style-type: none"> <li>escape('http://www.test.com/가.html')</li> </ul> </li> <li>• 인코딩 결과:                         <ul style="list-style-type: none"> <li>http%3A//www.test.com/%uAC00.html</li> </ul> </li> </ul>
encodeURIComponent()	<ul style="list-style-type: none"> <li>• escape() 함수에서 인터넷 주소에 사용되는 일부 특수 문자 (:, ;, /, ?, &amp;)를 인코딩하지 않음</li> <li>• 파라미터 전달하는 인터넷주소 전체를 인코딩할때 사용</li> <li>• 예) encodeURIComponent('http://www.test.com/가.html')</li> <li>• 인코딩 결과 :                         <ul style="list-style-type: none"> <li>http://www.test.com/%EA%B0%80.html</li> </ul> </li> </ul>
encodeURIComponent()	<ul style="list-style-type: none"> <li>• 알파벳과 숫자를 제외한 모든 문자를 인코딩</li> <li>• 예)                         <ul style="list-style-type: none"> <li>encodeURIComponent('http://www.test.com/가.html')</li> </ul> </li> <li>• 인코딩 결과 :                         <ul style="list-style-type: none"> <li>http%3A%2F%2Fwww.test.com%2F%EA%B0%80.html</li> </ul> </li> </ul>

## 2. 인코딩, 디코딩 함수와 eval() 함수

### 1) 인코딩과 디코딩 함수

#### - 디코딩(decoding)

- 인코딩 된 것을 다시 원래의 문자나 정보로 되돌리는 것
- 디코딩 함수

함수	설 명
unescape()	<ul style="list-style-type: none"> <li>• escape()를 이용하여 인코딩한 정보를 원래의 정보로 디코딩</li> <li>• 예) unescape('http%3A//www.test.com/%uAC00.html') 디코딩 결과 : http://www.test.com/가.html</li> </ul>
decodeURI()	<ul style="list-style-type: none"> <li>• encodeURI()를 이용하여 인코딩한 정보를 원래의 정보로 디코딩</li> <li>• 예) decodeURI('http://www.test.com/%EA%B0%80.html')</li> <li>• 디코딩 결과 : http://www.test.com/가.html</li> </ul>
decodeURIComponent()	<ul style="list-style-type: none"> <li>• encodeURIComponent()를 이용하여 인코딩한 정보를 원래의 정보로 디코딩</li> <li>• 예) decodeURIComponent('http%3A%2F%2Fwww.test.com%2F%EA%B0%80.html')</li> <li>• 디코딩 결과 : http://www.test.com/가.html</li> </ul>

## 2. 인코딩, 디코딩함수와 eval()함수

### 1) 인코딩과 디코딩 함수

### - 인코딩 함수 결과

- 스크립트

```
<script>
    var encode = 'http://www.test.com?test=테스트';

    var out = '';
    out += 'escape() \n';
    out += escape(encode) + '\n\n';

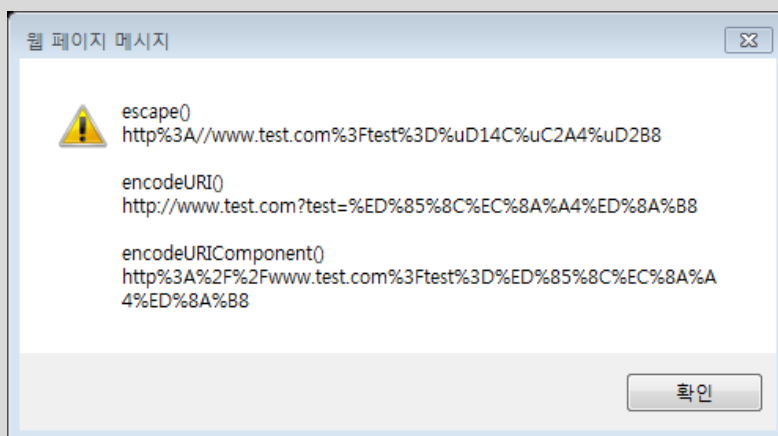
    out += 'encodeURIComponent \n';
    out += encodeURIComponent(encode) + '\n\n';

    out += 'encodeURIComponent() \n';
    out += encodeURIComponent(encode);

    alert(out);
</script>
```

- 결과

→ 인코딩한 결과가 인코딩 함수별로 다르게 나옴





## 2. 인코딩, 디코딩 함수와 eval() 함수

### 2) 코드 실행 함수

#### - eval() 함수

- 문자열을 코드로 실행할 수 있는 함수
- 표현식, 변수 등을 코드로 변환하여 실행
- eval() 함수의 괄호 안의 문자열이 JavaScript 코드로 해석됨
- 예) eval('2+6\*2')

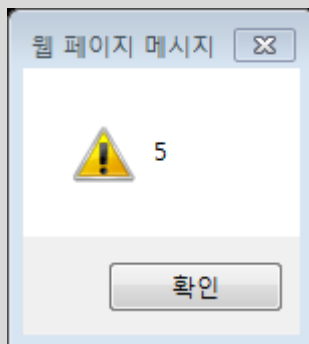
문자열 계산 결과 : 14

```
<script>
```

```
var x = 3;  
var y = 2;  
alert(eval('x+y ; '));
```

```
var strEval = '';  
strEval += 'var test=10;';  
strEval += 'alert(test)';  
eval(strEval);  
alert(test);
```

```
</script>
```



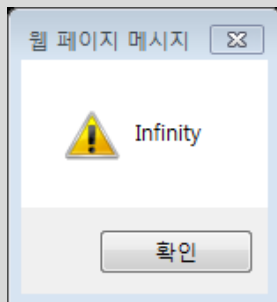
### 3. 숫자 관련 함수

#### 1) 숫자 확인 함수

- Infinity

- 무한대를 나타내는 특수한 값
- 예  
→ 3을 0으로 나눈 값을 변수 number에 저장

```
<script>  
  var number = 3 / 0;  
  alert(number);  
</script>
```



### 3. 숫자 관련 함수

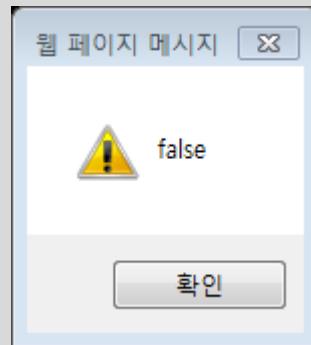
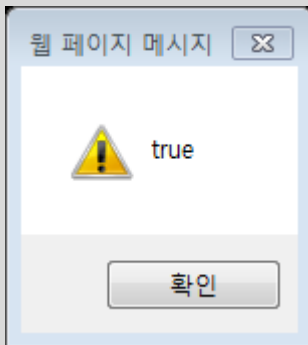
#### 1) 숫자 확인 함수

##### - isFinite()

- 값의 유한성과 숫자 여부를 판단하기 위해 사용하는 함수
- +무한대, -무한대, NaN 의 경우 false, 나머지는 true
- 예

```
<script>
  alert(isFinite(123));
  alert(isFinite(-1.23));
  alert(isFinite(5-2));
  alert(isFinite(0));
  alert(isFinite(1/0));
  alert(isFinite(-1/0));
  alert(isFinite('Hello'));
</script>
```

Red arrows indicate the results: true for the first three alerts and false for the last three.



##### - NaN

- Not a Number(숫자가 아님)의 약어
- 자료형은 숫자지만 표현 불가능 시 입력되는 값

##### - isNaN

- NaN 여부를 판단하기 위해 사용하는 함수
- 값이 NaN이면 true, 아니면 false

### 3. 숫자 관련 함수

#### 2) 숫자 변환 함수

##### - Number()

- 다른 자료형을 숫자로 변환할 때 사용하는 함수
- 문자를 포함하고 있으면 NaN을 리턴

##### - parseInt()

- 문자열을 정수로 변환할 때 사용하는 함수
- 문자열이 0x로 시작하면 16진수
- 예) 0x10(16진수) => 16  
문자열이 0으로 시작하면 8진수로 해석하여 값을 돌려주었으나, 현재는 사용하지 않음
- 변환하고자 하는 문자열의 처음이 문자로 시작하여 변환이 불가능할 경우 NaN을 리턴
- 두 번째 매개 변수에 진법을 입력하면 앞의 수를 해당 진법의 수로 인식

##### - parseFloat()

- 문자열을 부동 소수점 수로 변환할 때 사용하는 함수
- 문자열에서 숫자가 끝날 때까지 숫자로 변환
- 문자열의 처음이 문자로 시작하여 변환이 불가능할 경우 NaN을 리턴

### 3. 숫자 관련 함수

#### 2) 숫자 변환 함수

- parseInt()의 사용 예

```
<script>
  alert(parseInt('10'));
  alert(parseInt('10.00'));
  alert(parseInt('10.33'));
  alert(parseInt('34 55 66'));
  alert(parseInt(' 60 '));
  alert(parseInt('30 years'));
  alert(parseInt('He was 30'));
  alert(parseInt('010'));
  alert(parseInt('10',8));
  alert(parseInt('0x10'));
  alert(parseInt('10',16));
</script>
```

- parseFloat() 예시

```
<script>
  alert(parseFloat('10'));
  alert(parseFloat('10.00'));
  alert(parseFloat('10.33'));
  alert(parseFloat('34 55 66'));
  alert(parseFloat(' 60 '));
  alert(parseFloat('30 years'));
  alert(parseFloat('He was 30'));
</script>
```

## ▣ 정리하기

### 1. 대화 상자와 타이머 함수

- 사용자와의 상호작용을 위해 대화상자를 출력하는 함수는 `alert()`, `confirm()`, `prompt()` 등이 있음
- 타이머함수 : 특정한 시간 후나 간격으로 특정한 함수를 실행할 수 있게 하는 함수
- `setTimeout()`, `setInterval()` : 타이머를 설정
- `clearTimeout()`, `clearInterval()` : 타이머를 멈춤

### 2. 인코딩, 디코딩 함수와 `eval()` 함수

- 인코딩 : 문자를 컴퓨터에 저장하거나 통신에 사용하기 위해 부호화하는 것
- 디코딩 : 원래 정보로 복원하는 것
- 인코딩 함수 : `escape()`, `encodeURIComponent()`, `encodeURIComponent()`
- 디코딩 함수 : `unescape()`, `decodeURI()`, `decodeURIComponent()`

## ▣ 정리하기

### 3. 숫자 관련 함수

- Infinity : 무한대를 나타내는 특수한 값
- NaN : Not a Number의 약어로서, 숫자로 표현할 수 없을 때 입력되는 값
- isFinite( ) 함수 : 값의 유한성과 숫자 여부를 판단하기 위해 사용
- isNaN( ) 함수 : NaN 여부를 판단하기 위해 사용