



2 자바 스크립트 기본

모던 웹을 위한 Javascript jQuery 입문

2.1 출력

❖ Javascript 출력

- 기본으로 사용할 페이지 구성 예제 코드

코드 2-1 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
    <script>

    </script>
</head>
<body>

</body>
</html>
```



❖ Javascript 출력

- 가장 기본적인 출력 방법 - alert() 함수 사용
 - 웹 브라우저에 경고창 띄울 수 있음
 - alert() 함수의 사용 예
 - 함수의 괄호 안에는 문자열 입력

코드 2-2 기본적인 출력

```
<script>  
    alert('Hello JavaScript..!');  
</script>
```



❖ Javascript 출력

■ 매개 변수

- 함수의 괄호 안에 들어가는 것
- 코드를 입력하다 보면 괄호를 열 때 아래 그림과 같은 보조 기능 확인 가능
- HTML 페이지 저장 후 실행 결과

그림 2-1 alert() 함수의 매개 변수

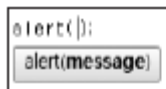


그림 2-2 alert() 함수의 실행 결과



❖ 문자열이란?

- 문자를 표현할 때 사용하는 자료의 형태
- alert() 함수의 매개 변수로 쓰인 'Hello JavaScript..!' 와 같은 자료
- 문자열을 만드는 방법
 - "동해물과 백두산이" (큰 따옴표)
 - '동해물과 백두산이' (작은 따옴표)
 - 두 가지 중 어떤 방법으로 문자열을 만들어도 되지만, 일관되게 사용할 것



❖ 예외적인 문자열 사용법

- 문자열 안에 쓰는 따옴표
 - 내부에 작은 따옴표를 쓰고 싶으면 외부에 큰 따옴표
 - 내부에 큰 따옴표를 쓰고 싶으면 외부에 작은 따옴표

코드 2-3 문자열

```
<script>  
    alert('This is "string"');  
    alert("This is 'string'");  
</script>
```



❖ 예외적인 문자열 사용법

- 한 가지 따옴표로만 쓰고 싶을 경우 이스케이프 문자 사용
 - 이스케이프 문자란? 특수한 기능을 수행하는 문자
 - 문자 그대로 따옴표를 사용하고 싶다면?
 - 예제 코드 2-4처럼 따옴표 앞에 \를 사용해 따옴표를 문자 그대로 사용하겠다고 표시 (\와 ₩ 는 같은 표시)

코드 2-4 이스케이프 문자 (1)

```
<script>
    alert("This is \"string\"");
    alert('This is \'string\'');
</script>
```



❖ 예외적인 문자열 사용법

- 이스케이프 문자를 이용한 출력법 예제
 - 이스케이프 문자 \n은 문자열을 줄바꿈할 때 사용

코드 2-5 이스케이프 문자 (2)

```
<script>  
    alert('동해물과 백두산이\n마르고 닳도록');  
</script>
```

그림 2-3 출력 결과



❖ 예외적인 문자열 사용법

- 자주 사용되는 이스케이프 문자

표 2-1 자주 사용하는 이스케이프 문자

이스케이프 문자	설명
\t	수평 탭
\n	줄바꿈
\'	작은따옴표
\"	큰따옴표
\\	역 슬래시



2.2 문자열

❖ 예외적인 문자열 사용법

■ 연결 연산자 '+'

- '가나다' + '라마' + '바사아' + '자차카타' + '파하'

– 가나다라마바사아자차카타파하



❖ 숫자 자료형

- 정수와 유리수의 구분 없이 숫자는 모두 숫자
- Ex)
 - 273
 - 52.273
- 문자열과 마찬가지로 alert() 함수의 괄호 안에 사용해 출력

코드 2-6 숫자

```
<script>  
    alert(273);  
    alert(52.273);  
</script>
```



❖ 숫자를 사용한 기본적인 사칙 연산

- 표 2-2의 연산자를 사용해 기본적인 사칙연산 가능
- 자바스크립트에서는 연산자 우선순위 고려함
 - $5 + 3 * 2 ? 5 + (3 * 2)$
 - 덧셈을 먼저 실행하고 싶다면 괄호의 위치를 바꿀 것

코드 2-7 기본적인 사칙 연산

```
<script>
    alert(5 + 3 * 2);
    alert((5 + 3) * 2);
</script>
```

그림 2-4 사칙 연산 결과



❖ 숫자를 사용한 기본적인 사칙 연산

■ % 연산자

- 좌변을 우변으로 나눈 나머지를 표시하는 연산자

표 2-3 나머지 연산자

연산자	설명
%	나머지 연산자

코드 2-8은 10을 7로 나눈 나머지를 출력합니다.

코드 2-8 나머지 연산자

```
<script>  
    alert(10 % 7);  
</script>
```



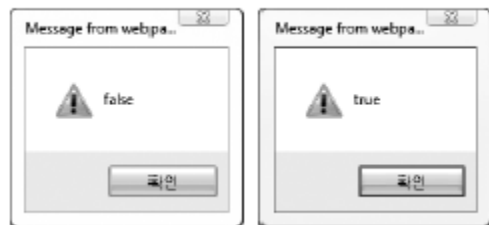
❖ 불리언

- 자바스크립트에서는 참과 거짓이라는 값을 표현할 때 사용
- 예제
 - $52 > 273$
 - $52 < 273$
- 위의 예제를 예제코드로 바꾼 결과

코드 2-9 비교 연산자를 사용한 참과 거짓의 구별

```
<script>  
    alert(52 > 273);  
    alert(52 < 273);  
</script>
```

그림 2-5 코드 2-9의 실행 결과



❖ 불리언의 의미를 확실히 해주는 비교 연산자

- 비교연산자란?
 - 두 대상을 비교할 수 있는 연산자
- 자바 스크립트에서 쓰이는 비교 연산자

표 2-4 비교 연산자

연산자	설명
>=	좌변이 우변보다 크거나 같다.
<=	우변이 좌변보다 크거나 같다.
>	좌변이 크다.
<	우변이 크다.
==	좌변과 우변이 같다.
!=	좌변과 우변이 다르다.



❖ 비교 연산자의 확장된 쓰임 - 문자열 비교

- 문자열은 국어사전의 앞쪽에 있을수록 작은 값을 가짐
 - '가방' > '하마' ⇨ false
- 유니코드 문자를 사용해 비교하므로 모든 언어 비교 가능
 - '尹' == '尹' ⇨ true
- 자바스크립트는 불리언끼리 크기 비교도 가능
 - 자바스크립트는 true를 1로 변환하고 false를 0으로 변환한 뒤 비교 연산
 - 예제 코드의 결과는 true

코드 2-10 불리언과 불리언의 비교

```
<script>  
    alert(true > false);  
</script>
```



❖ 조건문에서 불리언의 사용

- 조건문 괄호 안의 불리언 표현식이 참이면 중괄호 속 문장 실행
- 거짓이면 중괄호 속 문장 무시

```
if(불리언 표현식){  
    불리언 표현식이 참일 때 실행할 문장  
}
```

코드 2-11 불리언과 조건문

```
<script>  
    if(273 < 52){  
        alert('273은 52보다 작습니다.');    }  
  
    if(273 > 52){  
        alert('273은 52보다 큼니다.');    }  
</script>
```



❖ 자바 스크립트 논리연산자의 종류

표 2-5 논리 연산자

연산자	설명
!	논리 부정 연산자
&&	논리곱 연산자
	논리합 연산자

- 1) 논리부정 연산자
 - 참을 거짓으로, 거짓을 참으로 바꿈

코드 2-12 논리 부정 연산자

```
<script>
    alert(!true);
    alert(!false);
</script>
```



❖ 자바 스크립트 논리연산자의 종류

■ 논리곱 연산자

- 좌변과 우변이 모두 참일 때만 참

표 2-6 논리곱 연산자

좌변	우변	결과
true	true	true
true	false	false
false	true	false
false	false	false



❖ 자바 스크립트 논리연산자의 종류

- 논리합 연산자
 - 좌변과 우변이 모두 거짓일 때만 거짓

표 2-7 논리합 연산자

좌변	우변	결과
true	true	true
true	false	true
false	true	true
false	false	false



❖ 잘못된 비교 연산자의 사용

코드 2-13 잘못된 비교 연산자의 사용

```
<script>
    alert(30 > 20 > 10);
</script>
```

그림 2-6 표현식 '30 > 20 > 10'의 연산 과정

```
alert(30 > 20 > 10);
```



```
alert((30 > 20) > 10);
```



```
alert(true > 10);
```



```
alert(1 > 10);
```



```
alert(false);
```



❖ 비교 연산자와 논리 연산자의 적절한 사용 필요

코드 2-14 비교 연산자와 논리 연산자

```
<script>  
    alert(30 > 20 && 20 > 10);  
</script>
```

그림 2-7 표현식 "30 > 20 && 20 > 10"의 연산 과정

```
alert(30 > 20 && 20 > 10);
```



```
alert(true && true);
```



```
alert(true);
```

그림 2-8 그림으로 나타낸 범위 연산



❖ 변수

- 값을 저장할 때 사용하는 식별자
 - 변수지만 숫자뿐만 아니라 모든 자료형 저장 가능
 - 변수를 사용하려면?
 - 1. 변수 선언 : 변수를 만들
» Var 식별자 ;

코드 2-15 변수의 선언

```
<script>  
    // 변수를 선언합니다.  
    var pi;  
</script>
```

- 2. 변수 초기화



❖ 변수에 값 할당하기

- 변수에 값을 저장하는 것
- 변수 초기화
 - 변수를 선언한 후에 처음 값을 지정하는 것
 - 일반적으로 변수의 선언과 할당은 함께 일어남
 - 할당은 여러 번 할 수 있지만 초기화는 한 번만 가능

코드 2-16 변수의 선언과 할당

```
<script>
  // 변수를 선언합니다.
  var pi;

  // 변수에 값을 할당합니다.
  pi = 3.14159265;
</script>
```

코드 2-17 변수의 선언과 초기화

```
<script>
  // 변수를 선언하고 초기화합니다.
  var pi = 3.14159265;
</script>
```



❖ 변수에 값 할당하기

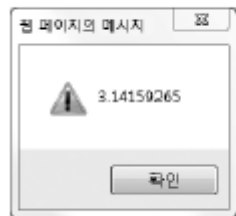
■ 변수 사용 (1)

코드 2-18 변수의 사용 (1)

```
<script>
  // 변수를 선언하고 초기화합니다.
  var pi = 3.14159265;

  // 출력합니다.
  alert(pi);
</script>
```

그림 2-9 변수에 저장된 값 출력



❖ 변수에 값 할당하기

■ 변수 사용 (2)

- 숫자가 들어간 변수는 숫자와 관련된 연산자 사용
- 문자열이 들어간 변수는 문자열과 관련된 연산자 사용

코드 2-19 변수의 사용 (2)

```
<script>
    // 변수를 선언하고 초기화합니다.
    var radius = 10;
    var pi = 3.14159265;

    // 출력합니다.
    alert(2 * pi * radius);
</script>
```



❖ 변수에 값 할당하기

■ 변수 사용 (3)

- var 키워드를 사용해 여러 변수를 한꺼번에 선언
 - var 키워드 뒤에 쉼표를 사용해 식별자를 연속으로 입력

코드 2-20 여러 변수의 선언

```
<script>
    // 변수를 선언합니다.
    var radius, pi;

    // 변수에 값을 할당합니다.
    radius = 10;
    pi = 3.14159265;

    // 출력합니다.
    alert(2 * pi * radius);
</script>
```



❖ 변수에 값 할당하기

■ 변수 사용 (4)

- 여러 개의 변수를 한 번에 선언하고 초기화

코드 2-21 복수 변수의 선언과 초기화

```
<script>
    // 변수를 선언하고 초기화합니다.
    var radius = 10, pi = 3.14159265;

    // 출력합니다.
    alert(2 * pi * radius);
</script>
```



❖ 변수에 값 할당하기

■ 자료형

- 문자열, 숫자, 불리언과 같은 것
- 자바스크립트에는 총 여섯 가지 자료형이 있음

– Cf. undefined 자료형

- » 선언되지 않거나 할당되지 않은 변수
- » 변수에 저장해도 의미가 없음

코드 2-22 자바스크립트의 자료형

```
<script>
  // 변수를 선언합니다.
  var stringVar = 'String';
  var numberVar = 273;
  var booleanVar = true;
  var functionVar = function () {};
  var objectVar = {};
</script>
```



❖ 복합 대입 연산자

- 대입 연산자와 다른 연산자를 함께 사용하는 연산자

표 2-8 복합 대입 연산자

연산자	설명
+=	기존 변수의 값에 값을 더합니다.
-=	기존 변수의 값에 값을 뺍니다.
*=	기존 변수의 값에 값을 곱합니다.
/=	기존 변수의 값에 값을 나눕니다.
%=	기존 변수의 값에 나머지를 구합니다.



❖ 복합 대입 연산자의 사용

- 변수 value를 10으로 초기화
- 이후 + = 복합 대입 연산자를 사용해 value의 기존 값에 10을 더함
- 결과는 20 출력

코드 2-23 복합 대입 연산자

```
<script>
    // 변수를 선언합니다.
    var value = 10;

    // 연산자를 사용합니다.
    value += 10;

    // 출력합니다.
    alert(value);
</script>
```



❖ 복합 대입 연산자의 사용

- 변수 list를 빈 문자열("")로 초기화
- + = 복합 대입 연산자를 사용해 문자열 만들고
- HTML 문서의 body 태그에 넣음

코드 2-24 복합 대입 연산자의 활용

```
<script>
    window.onload = function () {
        // 변수를 선언합니다.
        var list = '';

        // 연산자를 사용합니다.
        list += '<ul>';
        list += '    <li>Hello</li>';
        list += '    <li>JavaScript..!</li>';
        list += '</ul>';

        // 문서에 출력합니다.
        document.body.innerHTML = list;
    }
</script>
```

그림 2-11 복합 대입 연산자를 사용한 문서 객체 생성

- Hello
- JavaScript..!



❖ 증감 연산자

- 복합 대입 연산자를 간략하게 사용한 형태

표 2-9 증감 연산자

연산자	설명
변수++	기존의 변수의 값에 1을 더합니다(후위).
++변수	기존의 변수의 값에 1을 더합니다(전위).
변수--	기존의 변수의 값에 1을 뺍니다(후위).
--변수	기존의 변수의 값에 1을 뺍니다(전위).



❖ 증감 연산자의 활용 (1)

- 변수 number를 초기화하고 ++ 증감 연산자 사용
 - 코드를 실행하면 10에 1을 더한 11이 출력
 - ++ number로도 출력해봐도 차이가 없음
 - 한 줄에 독립적 증감 연산자를 사용할 때는 전위와 후위의 차이가 없음
 - 다른 연산자나 함수와 함께 사용할 때 차이가 있음

코드 2-25 증감 연산자 (1)

```
<script>
    // 변수를 선언합니다.
    var number = 10;

    // 연산자를 사용합니다.
    number++;

    // 출력합니다.
    alert(number);
</script>
```



❖ 증감 연산자의 활용 (2)

- 실행하면 순서대로 10, 11, 12 출력
- 후위는 해당 문장을 실행한 후에 값을 더하라는 의미
- `alert(number+ +)`는 `alert(number)`를 실행한 후 숫자 1을 더함

코드 2-26 증감 연산자(2)

```
<script>
    // 변수를 선언합니다.
    var number = 10;

    // 출력합니다.
    alert(number++);
    alert(number++);
    alert(number++);
</script>
```

코드 2-27 증감 연산자 (2) - 풀이

```
<script>
    // 변수를 선언합니다.
    var number = 10;

    // 출력합니다.
    alert(number); number += 1;
    alert(number); number += 1;
    alert(number); number += 1;
</script>
```



❖ 증감 연산자의 활용 (3)

- 해당 문장을 실행하기 전에 값을 더하는 것이 전위

코드 2-28 증감 연산자 (3)

```
<script>
  // 변수를 선언합니다.
  var number = 10;

  // 출력합니다.
  alert(++number);
  alert(++number);
  alert(++number);
</script>
```



❖ 증감 연산자의 활용 (4)

코드 2-29 증감 연산자 (4)

```
<script>
    // 변수를 선언합니다.
    var number = 10;

    // 출력합니다.
    alert(number++);
    alert(++number);
    alert(number--);
    alert(--number);
</script>
```

코드 2-30 증감 연산자 (5)

```
<script>
    // 변수를 선언합니다.
    var number = 10;

    // 출력합니다.
    alert(number);
    number++;
    number++;
    alert(number);
    alert(number);
    number--;
    number--;
    alert(number);
</script>
```



❖ 변수의 특성

- 변수는 하나만 담을 수 있음

코드 2-31 변수의 특성

```
<script>
    // 변수를 선언합니다.
    var cup = 'Coffee';

    cup = 'Green Tea';
    cup = 'Water';

    // 출력합니다.
    alert('Drink ' + cup + '..!');
</script>
```

그림 2-12 Drink Water..!



❖ 변수의 재선언 (1)

코드 2-32 변수의 재정의

```
<script>
    // 변수를 선언합니다.
    var favoriteFood = '김치 찌개';
    var favoriteFood = '라면';
    var favoriteFood = '냉면';
    // 출력합니다.
    alert(favoriteFood);
</script>
```



❖ 변수의 재선언 (2)

- 기존에 사용하던 식별자를 재선언하면 문제 발생

코드 2-33 재 선언의 문제점

```
<script>  
    // 변수를 선언합니다.  
    var alert = 'Red Alert';  
  
    // 출력합니다.  
    alert(alert);  
</script>
```



❖ 변수의 재선언 (3)

- 예제 코드의 출력 값은?

코드 2-34 변수 정리

```
<script>
  // 1번 문제
  var value = 10;
  value += 20;
  alert(value);

  // 2번 문제
  var value = 'Hello' + '!!';
  alert(value + ' JavaScript');
</script>
```



2.6 자료형 검사

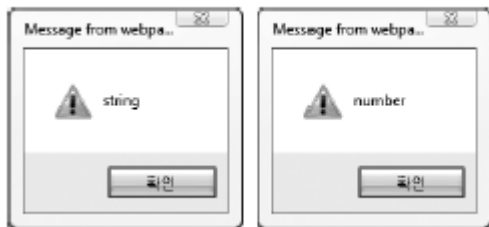
❖ 자료형

- 숫자, 문자열, 불리언 같은 자료의 형태
- typeof 연산자
 - 자료형을 확인할 때 사용

코드 2-35 typeof 연산자

```
<script>  
    alert(typeof ('String'));  
    alert(typeof (273));  
</script>
```

그림 2-13 typeof 연산자를 사용해 자료형 확인



❖ 자료형을 출력하는 예제 코드

■ Undefined

- 정의하지 않은 자료형 의미
- 선언하지 않은 식별자 alpha 사용

코드 2-36 자바스크립트의 자료형 확인

```
<script>
  // 문자열
  alert(typeof ('String'));
  // 숫자
  alert(typeof (273));
  // 불리언
  alert(typeof (true));
  // 함수
  alert(typeof (function () { }));
  // 객체
  alert(typeof ({}));
  // Undefined
  alert(typeof (alpha));
</script>
```



❖ 문자열을 입력하는 방법

■ 숫자를 입력 받는 방법

- 문자열을 입력 받은 후 숫자로 변환
- 문자열을 입력을 할 때 사용하는 함수는 prompt() – 매개변수 두 개 필요

그림 2-14 prompt() 함수 사용

```
prompt(  
  prompt(message, defstr)
```

코드 2-37 기본 입력 (1) – prompt() 함수

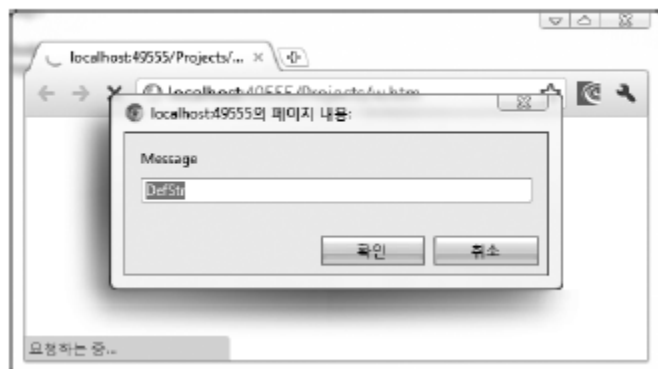
```
<script>  
  // 변수를 선언합니다.  
  var input = prompt('Message', 'DefStr');  
  
  // 출력합니다.  
  alert(input);  
</script>
```



❖ Prompt() 함수 실행

- 입력 칸에 변수를 입력하면 그대로 코드 변경

그림 2-15 prompt() 함수의 실행 결과



```
<script>
    // 변수를 선언합니다.
    var input = 'Hello World..!';

    // 출력합니다.
    alert(input);
</script>
```



❖ 불리언을 입력하는 방법

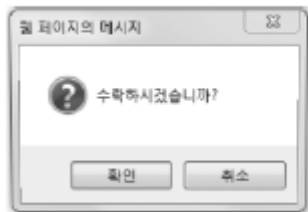
- confirm() 함수
 - prompt() 함수와 비슷한 방식으로 사용
 - 사용자가 확인을 누르면 true 리턴
 - 취소를 누르면 false 리턴
 - 변수 input에 불리언이 들어가고 곧바로 변수 input을 출력

코드 2-38 기본적인 입력 (2) - confirm() 함수

```
<script>
  // 변수를 선언합니다.
  var input = confirm('수락하시겠습니까?');

  // 출력합니다.
  alert(input);
</script>
```

그림 2-16 confirm() 함수의 실행 결과



❖ 배열이란?

- 여러 개의 변수를 한꺼번에 다룰 수 있는 자료형
- 모든 형태의 변수를 다룰 수 있는 자료형
- 객체 중 하나
- 대괄호([])를 사용해 생성
 - 안에 쉼표로 구분해 자료 입력 (배열 요소라 부름)

코드 2-39 배열

```
<script>  
    // 변수를 선언합니다.  
    var array = [273, 32, 103, 57, 52];  
</script>
```



❖ 배열 내용 출력 예제 (1)

코드 2-40 배열의 선언과 출력

```
<script>
    // 변수를 선언합니다.
    var array = [273, 'String', true, function () { }, {}, [273, 103]];

    // 출력합니다.
    alert(array);
</script>
```

그림 2-17 배열의 출력



❖ 배열 내용 출력 예제 (2)

- 배열의 맨 앞에 있는 요소는 0번째
- 인덱스 - '몇 번째' 라 불리는 숫자

코드 2-41 배열 요소에 접근하는 방법

```
<script>
  // 변수를 선언합니다.
  var array = [273, 32, 103, 57, 52];

  // 출력합니다.
  alert(array[0]);
  alert(array[2]);
  alert(array[4]);
</script>
```



2.9 undefined 자료형

❖ undefined 자료형이란?

- '존재하지 않는 것'을 표현하는 자료형

코드 2-42 undefined (1)

```
<script>  
    // 출력합니다.  
    alert(typeof (variable));  
</script>
```

그림 2-18 undefined 자료형



2.9 undefined 자료형

❖ undefined 자료형이란?

- 변수를 선언했지만 초기화하지 않았을 때

코드 2-43 undefined (2)

```
<script>
    // 변수를 선언합니다.
    var variable;

    // 출력합니다.
    alert(typeof (variable));
</script>
```



2.10 숫자와 문자열 자료형 변환

❖ 문자열과 숫자를 더하는 자료형

- 숫자와 문자열을 덧셈 연산하면 문자열 우선

코드 2-44 숫자와 문자열 자료형 변환 (1)

```
<script>
    // 1번
    alert('52 + 273');
    // 2번
    alert(52 + 273);
    // 3번
    alert('52' + 273);
    // 4번
    alert(52 + '273');
    // 5번
    alert('52' + '273');
</script>
```



2.10 숫자와 문자열 자료형 변환

❖ 문자열과 숫자를 곱하는 자료형

- 더하기 연산자를 제외한 사칙 연산자는 숫자가 우선
 - 첫 번째를 제외하면 14196을 출력

코드 2-45 숫자와 문자열 자료형 변환 (2)

```
<script>
    alert('52 * 273');
    alert(52 * 273);
    alert('52' * 273);
    alert(52 * '273');
    alert('52' * '273');
</script>
```



2.10 숫자와 문자열 자료형 변환

❖ 강제로 자료형 변환시키기

- 다른 자료형을 숫자로 - Number() 함수
- 다른 자료형은 문자열로 - String() 함수

❖ prompt() 함수를 사용하면 문자열만 입력 가능

- 코드 2-46은 아무리 숫자를 입력해도 문자열의 자료형 string 출력

코드 2-46 숫자의 입력 (1)

```
<script>
    // 변수를 선언합니다.
    var input = prompt('숫자를 입력해주세요.', '숫자');

    // 출력합니다.
    alert(typeof (input));
</script>
```



2.10 숫자와 문자열 자료형 변환

- ❖ **prompt() 함수를 사용하면 문자열만 입력 가능**
 - 숫자로 바꾸려면 Number() 함수 사용

코드 2-47 숫자의 입력 (2)

```
<script>
// 변수를 선언합니다.
var input = prompt('숫자를 입력해주세요.', '숫자');
var numberInput = Number(input);

// 출력합니다.
alert(typeof (numberInput) + ': ' + numberInput);
</script>
```

그림 2-19 숫자를 입력한 경우의 결과

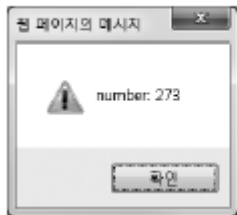


그림 2-20 숫자가 아닌 값을 입력한 경우의 결과



2.10 숫자와 문자열 자료형 변환

❖ 자바스크립트는 복소수를 지원하지 않음

- $\sqrt{3}$ 과 같은 복소수는 NaN으로 표시

코드 2-48 NaN

```
<script>
  // 변수를 선언합니다.
  var number = Math.sqrt(-3);

  // 변수를 출력합니다.
  alert(number);
</script>
```



❖ Boolean() 함수

- 불리언 자료형으로 변환할 때 사용
- 코드 2-49의 다섯 가지 경우가 false로 변환
 - 이 다섯 가지를 제외한 모든 경우 true로 변환
 - 문자열 '0' 과 문자열 'false' 는 문자열이므로 true

코드 2-49 불리언 자료형 변환 (1)

```
<script>
    alert(Boolean(0));
    alert(Boolean(NaN));
    alert(Boolean(''));
    alert(Boolean(null));
    alert(Boolean(undefined));
</script>
```



2.11 불리언 자료형 변환

❖ 조건문 사용/ 논리 부정 연산자 사용

- 자동으로 불리언 자료형으로 변환
- **undefined** 자료형은 **false**

코드 2-50 불리언 자료형 변환 (2)

```
<script>
  alert (!!0);
  alert (!!NaN);
  alert (!!'');
  alert (!!null);
  alert (!!undefined);
</script>
```



❖ 자동 자료형 변환

- 비교 연산자를 사용할 때 뜻하지 않는 경우가 발생
 - 네 가지 모두 true 출력

코드 2-51 비교 연산자의 사용

```
<script>
  alert('' == false);
  alert('' == 0);
  alert(0 == false);
  alert('273' == 273);
</script>
```



❖ 일치 연산자의 용도

- 자료형이 다른 것을 확실하게 구분 짓고 싶을 때 사용
 - 예제 코드의 결과는 모두 false

표 2-10 일치 연산자

연산자	설명
===	양 변의 자료형과 값이 일치합니다.
!==	양 변의 자료형과 값이 다릅니다.

코드 2-52 일치 연산자의 사용

```
<script>
  alert('' === false);
  alert('' === 0);
  alert(0 === false);
  alert('273' === 273);
</script>
```





Thank You !

모던 웹을 위한 Javascript jQuery 입문