



HTML Form



Web Programming
Dept of Computer Science

폼 관련 태그

attribute
특성

value
값

클라이언트가 제출한
사용자 입력 중
JSP로 전달된
범위 설정

태그명	결과	설명
<form></form>	안 보임	폼 양식의 시작과 끝
<input type=text>	<input type="text" value="happylucy"/>	한 줄 입력 상자
<input type=password>	<input type="password" value="****"/>	*로 표시되는 암호 입력상자
<input type=radio>	<input type="radio"/> 회사	라디오 버튼 단일 선택
<input type=checkbox>	<input type="checkbox"/> 영화감상	체크상자 다중 선택
<input type=button>	<input type="button" value="버튼"/>	일반 버튼
<input type=submit>	<input type="submit" value="쿼리 전송"/>	전송확인 버튼 사용자가 JSP 서버로
<input type=reset>	<input type="reset" value="원래대로"/>	초기화 버튼
<input type=image>	<input type="image" value="중복확인"/>	이미지 버튼
<input type=file>	<input type="file" value="찾아보기..."/>	파일 업로드
<input type=hidden>	안 보임	숨겨진 필드
<textarea> </textarea>	<input type="text"/>	여러 줄 입력상자
<select> </select>	<input type="text" value="2002"/>	목록 상자의 시작과 끝
<option>	<input type="text" value="2002"/>	목록 상자에 포함될 항목 지정



<INPUT> 태그 관련 속성

- size
- maxlength
- value
- readonly
- checked
- disabled



Form Buttons

- 일반 버튼
 - 아무런 역할도 하지 않으며, 자바 스크립트나 JSP와 연결하여 사용한다.
 - `<input type=button value="국적변경">` *html에서는 " "가 필수X*
- 초기화 버튼
 - 폼 양식 안에 입력한 값을 모두 초기화시킨다.
 - `<input type=reset value="다시 작성">`
- 전송 버튼
 - `<form>` 태그의 action에 지정한 웹 프로그램이 실행된다.
 - `<input type=submit value="가입하기">`
- 이미지 버튼
 - 일반 버튼과 같이 웹 프로그램을 연결하여 사용한다.
 - `<input type=image src="check.gif">`
→ 이미지 파일 src



Radio Buttons

- 라디오 버튼은 두 개 이상의 항목 중 하나만 선택할 수 있다.
- 속성 checked은 선택된 상태로 보이게 한다.
- 선택의 종류별로 같은 name 속성을 공유한다.
 중 하나만 선택
- 사용 예제
같은 세트
 - `<input type=radio name=gen value=1>남`
 - `<input type=radio name=gen value=2 checked> 여`



Checkbox

- 체크박스는 두 개 이상이 선택 가능하다.
- 선택의 종류별로 같은 name 속성을 공유한다.
- 사용 예제
 - `<input type="checkbox" name="family" value="1">` 조부모
 - `<input type="checkbox" name="family" value="2" checked>` 부모
 - `<input type="checkbox" name="family" value="3">` 형제자매
 - `<input type="checkbox" name="family" value="4">` 배우자
 - `<input type="checkbox" name="family" value="5">` 자식



목록 상자

- <select> <option> 태그 관련 속성
 - selected
 - size
 - multiple
- 사용 예제
 - <select name=phone>
 - <option value=1 selected>010</option>
 - <option value=2>011</option>
 - <option value=3>017</option>
 - <option value=4>070</option>
 - <option value=5>080</option>
 - </select>

초기 화면



<TEXTAREA> 태그 관련 속성

- <textarea> 태그 관련 속성
 - cols
 - rows
 - <textarea> 초기 입력값 </textarea>



HTML 폼 양식

- <input> 태그

표 4-2 HTML5에 추가된 <input> 태그의 type 속성

속성	속성 값	설명
type	color	색상을 선택할 수 있는 폼 양식
	datetime-local	날짜와 시간을 선택할 수 있는 폼 양식
	month	연도와 월을 선택할 수 있는 폼 양식
	number	숫자를 선택할 수 있는 폼 양식
	range	슬라이더 바로 값을 선택할 수 있는 폼 양식
	time	시간을 선택할 수 있는 폼 양식
	week	연도와 주를 선택할 수 있는 폼 양식
	date	연도와 월, 일을 선택할 수 있는 폼 양식



HTML5에서 추가된 폼 양식

예제 4-11 HTML5에 추가된 폼 양식 사용하기

source/04/html5_form.html

<!-- 생략 -->

07 <form>

08 <input type="color">

09 <input type="datetime-local">

10 <input type="month">

11 <input type="number">

12 <input type="range">

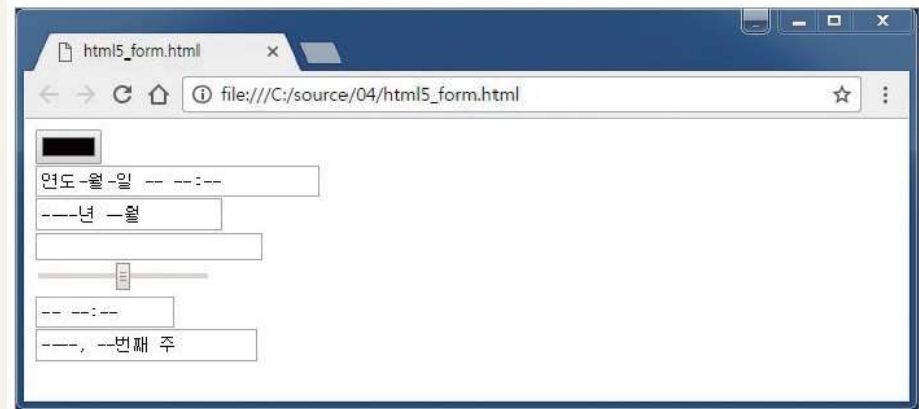
13 <input type="time">

14 <input type="week">

15 </form>

<!-- 생략 -->

▼ 실행 결과





JavaScript

Web Programming

Department of Computer Science

자바스크립트란?

■ 자바스크립트(Javascript)

- 정적인 HTML 콘텐츠에서 사용자와 상호작용하며 동적으로 변경하는 부분을 담당
- 객체(Object) 기반의 스크립트 언어로 기본적으로는 웹 브라우저에서 해석되는 인터프리터 언어임
- Node.js와 같은 프레임워크를 사용하면 서버 프로그래밍에도 사용할 수 있음

■ 자바스크립트의 특징

- 동적이며 타입을 명시할 필요가 없는 인터프리터 언어
- 객체지향 프로그래밍과 함수형 프로그래밍을 모두 표현할 수 있음
- HTML의 내용, 속성, 스타일을 변경할 수 있음
- 이벤트를 처리하고 사용자와의 상호작용을 가능하게 함
- 서버와 실시간 통신 기능을 제공

자바스크립트란? → 파생어

■ 자바스크립트의 현황

- 뷰(Vue), 앵귤러(Angular), 리액트(React)와 같은 프론트엔드 프레임워크에서부터 Node.js를 기반으로 하는 서버 프로그래밍 영역까지 자바스크립트가 널리 사용됨
- 자바스크립트는 네이버, 구글과 같은 웹 서비스에서부터 데스크톱 응용 프로그램까지 사용되지 않는 곳이 없음

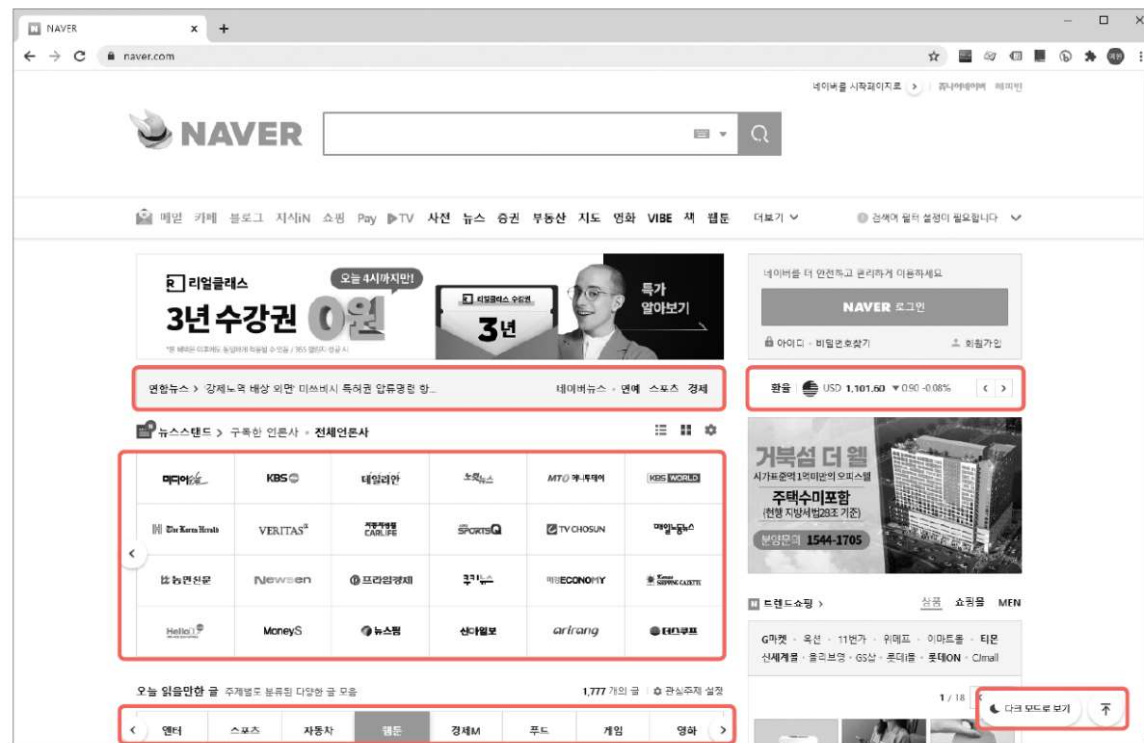


그림 3-6 네이버 화면에서 사용되는 자바스크립트의 예시

자바스크립트란?

- 자바스크립트가 사용되는 기능
 - 최신 뉴스 제목, 뉴스, 환율 등이 자동으로 스크롤되는 화면
 - 쇼핑, 뉴스 스탠드 등에서 현재 화면은 유지하면서 해당 콘텐츠의 페이지 섹션의 페이지만 좌우로 변경되는 화면
 - 웹 페이지에 날씨 정보 출력(날씨 정보 사이트에 실시간 데이터 요청 후 출력)
 - 홈쇼핑 등에서 웹 브라우저를 통한 고객 상담 채팅
 - 클릭하면 숨겨진 메뉴가 나오거나 탭 형태의 화면 구성
 - 선택에 따라 일부 화면이 바뀌거나 움직이는 형태


자바스크립트의 기본 문법

- 자바스크립트가 지원하는 프로그램 구성요소
 - 변수, 함수, 객체, 클래스
 - 반복문, 조건문
 - 배열, 리스트, 맵(Map) 등의 자료 구조
 - 비동기 처리 지원
 - HTTP 요청 및 응답 처리
 - HTML 문서에 접근할 수 있는 내장 객체와 함수
 - 이를 통해 DOM(Document Object Model)을 다룰 수 있음
 - 객체를 정의하는 JSON(JavaScript Object Notation) 구문

자바스크립트의 기본 문법

■ 자바스크립트의 문법적 특징

- 변수 타입을 따로 지정하지 않으며, 선언은 var, let, const를 사용함
- 범위 지정 없이 변수를 선언하면 전역변수가 되고 위치에 상관없이 호이스팅(끌어올림)되므로 주의해야 함
- 문자열을 표현할 때는 큰따옴표(“”)와 작은따옴표(‘ ’)를 모두 사용할 수 있음
- 함수형 언어를 지원하며 함수는 변수, 함수 인자, 객체 멤버 등에 사용될 수 있음
- <script> 태그는 HTML의 <head>와 <body> 영역 모두에서 사용 가능함
 - 하지만 DOM 접근을 위해서는 HTML 문서가 모두 로딩된 다음에 접근 가능
- JSON 구조를 광범위하게 사용함

여기서  **잠깐!** 같은 변수명으로 변수를 선언할 때 var, let, const의 차이점

- var(함수범위)는 이미 만들어진 변수 이름으로 재선언이 가능하다.
- let(블록범위 변수)은 이미 만들어진 변수 이름으로 재선언은 불가능하지만, 재할당은 가능하다.
- const(블록범위 상수)는 이미 만들어진 변수 이름으로 재선언이 불가능하며 재할당 또한 불가능하다.



JavaScript Basics



Web Programming
Dept of Computer Science



자바스크립트 작성과 실행

- 자바 스크립트는 HTML 문서의 일부로 작성 가능
- 자바스크립트로만 이루어진 파일의 경우는 ".js"라는 확장자를 가짐
- 문서 내부의 위치에 따라서 다음과 같이 종류가 있다
 - 내장형
 - 행 입력형
 - 함수형
 - 링크형



자바스크립트 삽입과 실행 (1)

- 내장형

- <BODY> 태그 내에서 사용됨
 <script type="text/javascript">
 // 자바스크립트 소스
 </script>

→ 코드 읽기 불편 → 가독성 ↓

- 행 입력형

- <태그 이벤트핸들러="자바스크립트 소스">

- <input type=button onClick="document.location='m1.htm'">

괄호를 때

- 함수형

- <HEAD> 태그 내부에서 정의
 <script type="text/javascript">
 function 함수명()
 {
 // 자바스크립트 소스
 }
 </script>

- <BODY> 태그 내부에서 호출
 <태그명 이벤트핸들러="함수명()">

- 링크형

- <script type="text/javascript" src="js 파일의 전체 경로">
- </script>

기일 때



자바 스트립트의 문법적 특징

- 대소문자를 반드시 구분해야 한다
- 구문은 한 줄에 한 개씩 위치시킨다 : 세미콜론 생략 가능 *vs ...*
- 소속객체, 소속변수, 소속메소드의 사용은 마침표(.) 연산자를 사용한다.
- 변수
 - 변수 타입을 따로 지정하지 않으며, 선언은 var, let, const를 사용함
 - 범위 지정 없이 변수를 선언하면 전역변수가 되고 위치에 상관없이 호이스팅(끌어올림)되므로 주의해야 함
- 문자열
 - 문자열을 표현할 때는 큰따옴표(“ ”)와 작은따옴표(‘ ’)를 모두 사용할 수 있음
 - 작은따옴표나 큰따옴표를 중첩해서 사용할 때는 반드시 나중에 시작한 따옴표를 먼저 닫아야 한다.
 - 따옴표 자체를 문자열에 포함시켜야 할 경우에는 역슬래시(\)와 따옴표를 함께 사용한다.



주석 달기

- JavaScript 영역
 - 한 행을 주석문 처리
`//주석 처리할 행, 문장`
 - 두 행 이상에 걸치는 주석문 처리
`/* 주석 처리할 영역 */`
- HTML 문서 영역
`<!-- 주석 처리할 영역 -->`



기본 실행문

- 변수 선언문 : `var i = 10;`
- 대입문 : `i = 10`이나 `i = "Dongduk Women's University"`
- 조건문

```
if(i < 10) {  
    document.write("조건만족");  
}
```
- 반복문

```
for(var i = 0; i < 10; i++) {  
    document.write(i);  
}
```



연산문

- 산술연산문 : +, -, *, /, %
 - 증감연산 : ++, --
- 대입연산문 : =, +=, -=, *=, /=, %=
- 조건 연산자:
 - 변수명=(조건식)? 명령1 : 명령2
- 논리연산문 : &&, ||, !,
 - 관계연산자 : >, <, >=, <=
 - 비교연산 : ==, !=
- 연결연산문 : "happy" + "day"



IF 조건문

- 형식1

```
if(조건)  
    명령문
```

- 형식2

```
if(조건){  
    명령문1  
    명령문2  
}  
else{  
    명령문1  
    명령문2  
}
```

- 중첩 IF문

```
if(조건) {  
    명령문  
}  
else if(조건) {  
    명령문  
}  
else if(조건) {  
    명령문  
}  
else(조건) {  
    명령문  
}
```




SWITCH 문

```
switch (표현식) {  
    case value1:  
        명령문1;  
        break;  
    case value2:  
        명령문;  
        break;  
    .....  
    default:  
        명령문n  
}
```



FOR 문

- 기본 형식

```
for(초기 값;조건부;증감식) {  
    코드부  
}
```

- 예제

```
for(var a=1;a<11;a++) {  
    document.write(a + "*" + a + "=" + a*a + "<br>");  
}
```



WHILE 문

- 기본 형식

```
while (조건) {  
    명령문  
}
```

- 예제

```
var a=1;  
while (a<11) {  
    document.write(a+ "*" +a+ "=" +a*a+"<br>");  
    a++;  
}
```



DO WHILE 문

- 기본 형식

```
do {  
    명령문  
} while (조건);
```

- 예제

```
var a=1;  
do {  
    document.write(a+ "*" + a + "=" + a*a + "<br>");  
    a++;  
} while (a<11);
```



BREAK, CONTINUE 문

- break : 제어문 종료
- continue : 제어문 반복
- 예제

```
var a=10;
while (true) {
    a --;
    if(a>10) continue;
    if(a==0) break;
    document.write(a+ "*" +a+ "="+a*a+"<br>");
}
```



RETURN 문

- 함수에서 특정 값을 리턴 값으로 보내고 싶을 때 사용
- 예제

```
function square(a) {
```

```
    return a*a;
```

```
}
```

```
...
```

```
for(var a=1;a<11;a++) {
```

```
    document.write(a + "*" + a + "=" + square(a) + "<br>");
```

```
}
```



사용자 정의 함수

- 함수의 정의

- HTML 문서의 <head> 태그 부분에 정의

- ```
<script type="text/javascript">
```

- ```
function makeWindow() {
```

- ```
 window.open("allim.htm","new","width=200, height=200");
```

- ```
}
```

- ```
</script>
```

- 함수의 호출

- HTML 문서의 <body> 태그 부분에 정의

- ```
<body onload="makeWindow()">
```



사용자 정의 함수의 종류

- 매개변수가 없는 함수

```
function test() {...}
```

- 매개변수가 있는 함수

```
function test(name, addr) {...}
```

매개변수 타입 X

- 리턴 값이 있는 경우

```
function test(question) {  
    var ans=confirm(question) ;  
    return ans ;  
}
```


Questions & Comments

