

# HTML5 문서 구조화와 웹 폼

담당교수: 장 은 실

# 학습 목표

- HTML5의 문서 구조화의 목적과 시맨틱 태그에 대해 이해한다.
- HTML5 시맨틱 태그로 구조화된 웹 페이지를 만들 수 있다.
- 웹 폼의 목적을 이해한다.
- 웹 폼의 요소를 활용하여 사용자로부터 입력 받는 웹 페이지를 작성할 수 있다.

# 문서의 전형적인 구조 사례

자바 GUI 스윙 기초
8
CHAPTER
323

GUI 프로그램은 이미지나 도형을 다루는 클래스와 이벤트를 처리를 위한 클래스를 사용하기 때문에, 다음 import 문이 함께 필요한 경우가 많다.

```
import java.awt.*; // 그래픽 처리를 위한 클래스들의 경로명
import java.awt.event.*; // AWT 이벤트를 사용하기 위한 경로명
import javax.swing.*; // 스윙 컴포넌트 클래스들의 경로명
import javax.swing.event.*; // 스윙 이벤트를 위한 경로명
```

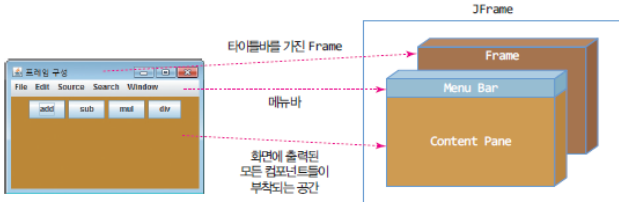
### 스윙 프레임과 콘텐츠팬

스윙 프레임은 모든 스윙 컴포넌트들을 담는 최상위 컨테이너(Top Level Container)이다. 프레임이 출력될 때, 프레임 내에 부착된 모든 컴포넌트들이 화면에 출력된다. 컴포넌트들은 프레임 없이 독립적으로 화면에 출력될 수 없다. 프레임이 닫히면 프레임 내의 모든 컴포넌트들도 프레임과 함께 화면에서 사라진다.

스윙에서 프레임의 역할을 하는 클래스가 JFrame이다. JFrame 객체는 [그림 8-4]와 같이 구성된다. JFrame 객체는 Frame(java.awt.Frame), 메뉴바(Menu Bar), 콘텐츠팬(Content Pane)의 세 공간으로 구성된다. Frame은 AWT 패키지에 있는 클래스로서 JFrame이 [그림 8-2]와 같이 java.awt.Frame을 상속받기 때문에 당연히 존재하며, 모든 메뉴는 메뉴바에 부착된다.

콘텐츠팬은 메뉴를 제외한 모든 GUI 컴포넌트들을 부착하는 공간이므로, 개발자는 화면에 출력하고자 하는 GUI 컴포넌트들을 콘텐츠팬에 붙여야 한다.

[그림 8-4]의 왼쪽은 JFrame의 사용 예로서 5개의 메뉴가 메뉴바에 부착되어 있으며, 콘텐츠팬에는 add, sub, mul, div 이름의 버튼 컴포넌트들이 부착되어 있다.



The diagram illustrates the internal structure of a JFrame. It is composed of three main parts: a 'Frame' container at the top, a 'Menu Bar' below it, and a 'Content Pane' at the bottom. A screenshot of a Java Swing window on the left shows a menu bar with five items ('add', 'sub', 'mul', 'div', and an unlabeled one) and a content area with four buttons labeled 'add', 'sub', 'mul', and 'div'. Arrows point from the screenshot to the corresponding parts of the 3D block diagram.

설명 섹션

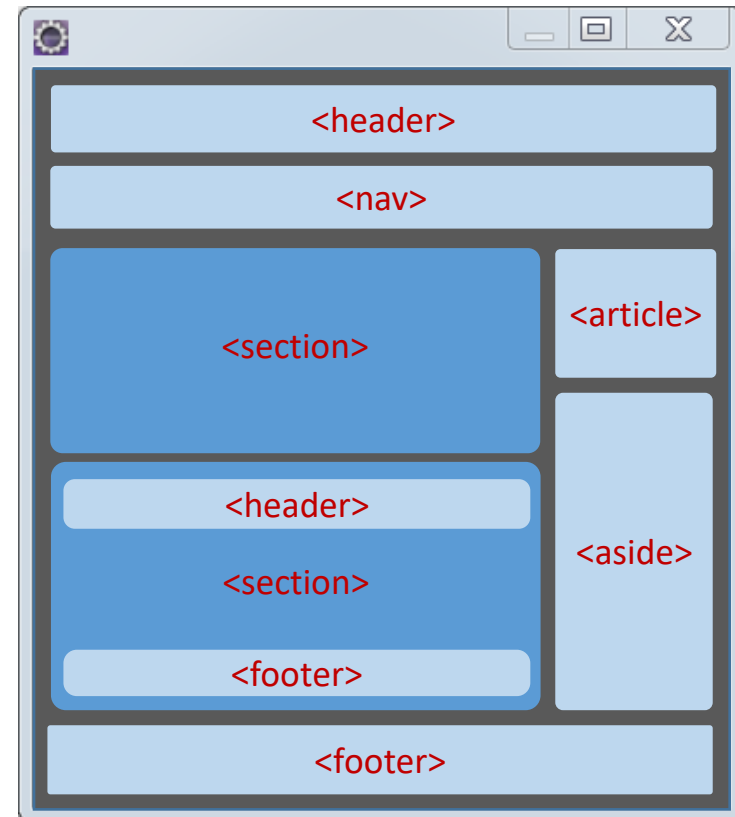
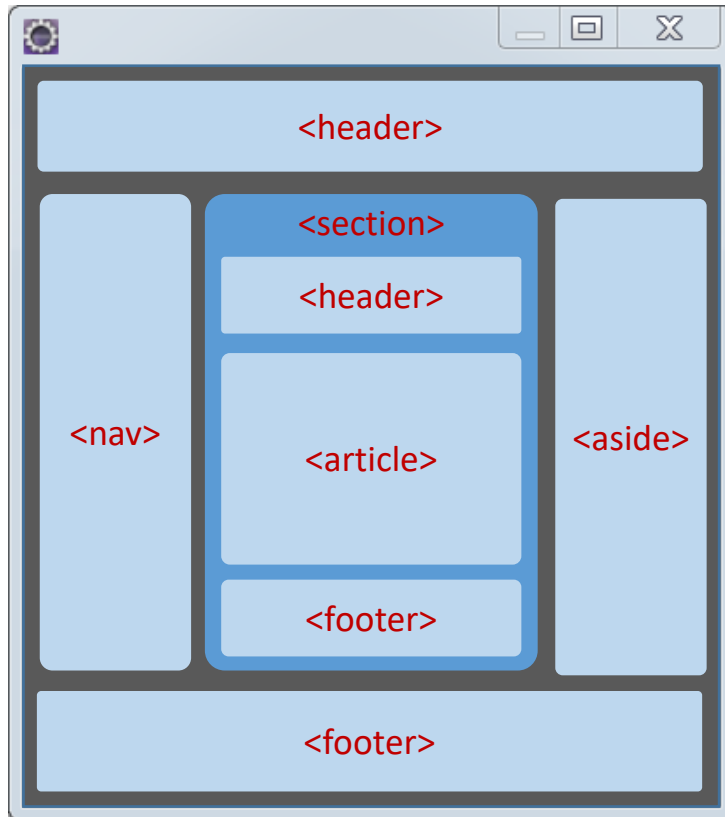
그림

[그림 8-4] 스윙 프레임(JFrame)의 구성

# HTML5의 문서 구조화

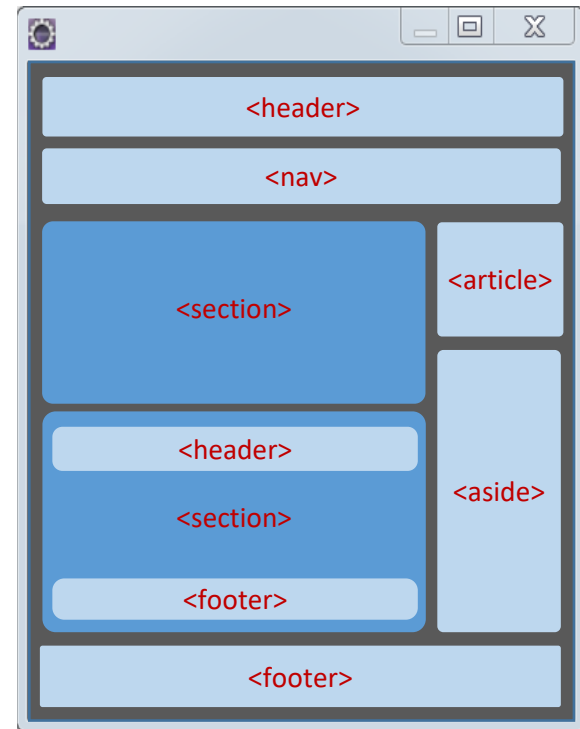
- 기존 HTML의 한계 : 웹 문서 구조를 표현하는 태그 없음
  - <div> 태그나 <table> 태그로 구조화되어 보이게 작성
  - HTML 페이지의 소스를 보면 문서 구조 파악 불가능
- 문서 구조화의 이유
  - 검색 엔진이 좋아하는 웹 페이지 작성의 필요성 대두
    - 정보 탐색이 중요해진 시대
    - 사물인터넷으로 IT 장치들이 스스로 정보 검색하는 시대
    - 사용자가 만든 웹 페이지 가치 극대화 – 탐색이 쉽도록 작성
- 시맨틱 웹
  - 웹 문서를 구조화하여 의미 있는 내용 탐색이 용이한 웹
  - 기존 태그
    - <p>, <div>, <h1>, <h2> 등 태그 사용. 문서의 구조나 의미 전달 어려움
  - 시맨틱 태그
    - 문서의 구조와 의미를 전달하는 태그
    - <header>, <section>, <article>, <main>, <summary>, <mark>, <time>
- 구글 검색 엔진 사례
  - 웹 페이지에서 시맨틱 태그 검색
  - 리뷰, 사람, 제품, 업체, 이벤트, 음악 등 검색 결과 제공

# HTML5 시맨틱 태그로 구조화한 웹 페이지 사례



# 시맨틱 태그

- `<header>`
  - 페이지나 섹션의 머리말 표현
  - 페이지 제목, 페이지를 소개하는 간단한 설명
- `<nav>`
  - 하이퍼링크들을 모아 놓은 특별한 섹션
  - 페이지 내 목차를 만드는 용도
- `<section>`
  - 문서의 장(chapter, section) 혹은 절을 구성하는 역할
  - 일반 문서에 여러 장이 있듯이 웹 페이지에 여러 `<section>` 가능
  - 헤딩 태그(`<h1>~<h6>`)를 사용하여 절 혹은 섹션의 주제 기입
- `<article>`
  - 본문과 연관 있지만, 독립적인 콘텐츠를 담는 영역
  - 혹은 보조 기사, 블로그 포스트, 댓글 등 기타 독립적인 내용
  - `<article>`에 담는 내용이 많은 경우 여러 `<section>` 둘 수 있음
- `<aside>`
  - 본문에서 약간 벗어난 노트나 팁
  - 신문, 잡지에서 주요 기사 옆 관련 기사, 삽입 어구로 표시된 논평 등
  - 페이지의 오른쪽이나 왼쪽에 주로 배치
- `<footer>`
  - 꼬리말 영역, 주로 저자나 저작권 정보

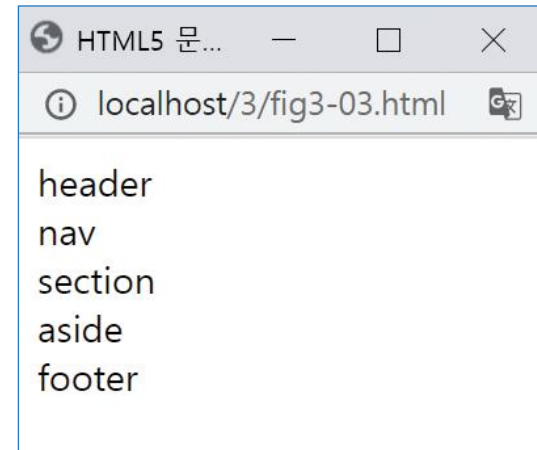


# 문서의 모양은 구조와 별개

- 시맨틱 태그

- <header>, <section>, <article>, <footer>, <nav> 태그
- 위치와 색, 모양이 자동으로 결정되는 것이 아님

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>HTML5 문서 구조 시맨틱 태그 사용</title>
</head>
<body>
  <header>header</header>
  <nav>nav</nav>
  <section>section</section>
  <aside>aside</aside>
  <footer>footer</footer>
</body>
</html>
```



- 개발자가 CSS3를 이용하여 직접 위치와 색, 모양을 지정해야 함

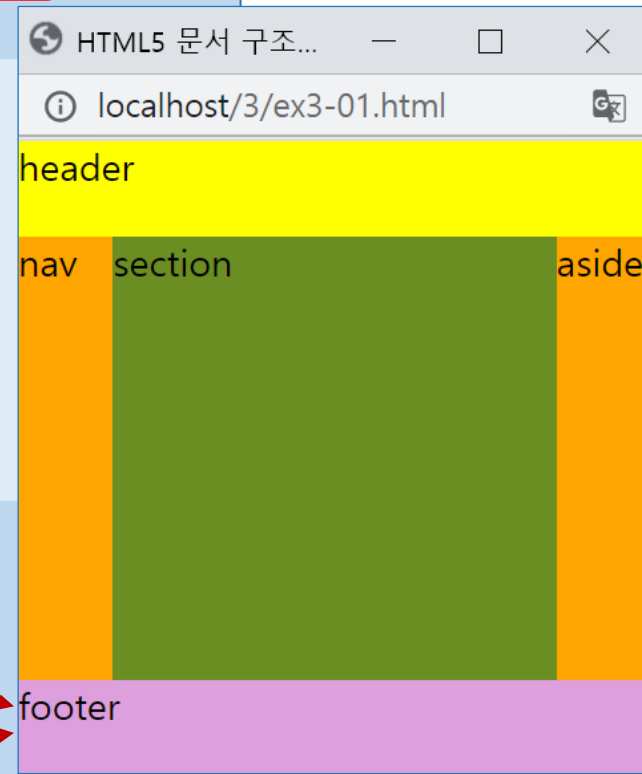
# 구조화된 HTML5 문서 작성

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>HTML5 문서 구조 시맨틱 태그 사용</title>
<style>
```

CSS3 스타일 시트.  
각 영역의 색과  
위치 꾸미기

```
html, body { margin: 0; padding: 0; height: 100%; }
header { width: 100%; height: 15%;
background: yellow; }
nav { width: 15%; height: 70%; float: left;
background: orange; }
section { width: 70%; height: 70%; float: left;
background: olivedrab; }
aside { width: 15%; height: 70%; float: left;
background: orange; }
footer { width: 100%; height: 15%; clear: both;
background: plum; }
```

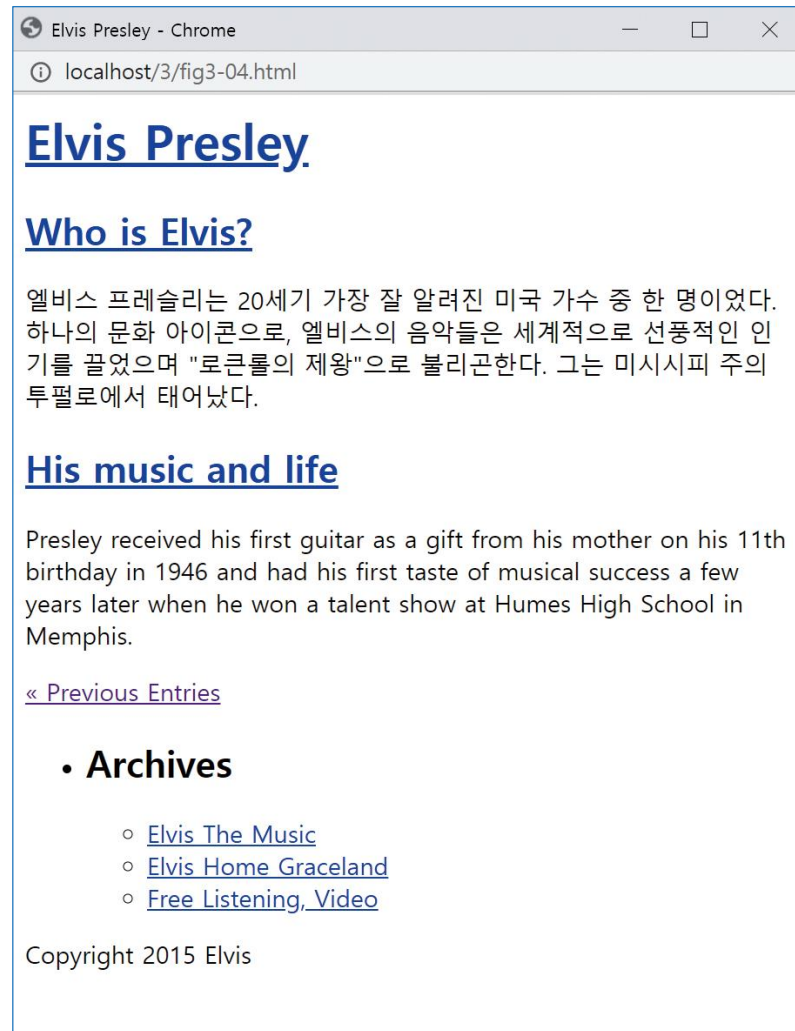
```
</style>
</head>
<body>
<header>header</header>
<nav>nav</nav>
<section>section</section>
<aside>aside</aside>
<footer>footer</footer>
</body>
</html>
```





# 기존 HTML 문서와 HTML5 문서 비교

샘플 웹 페이지



1. 기존 HTML 태그로만 만들기

2. HTML5 시맨틱태그를 사용하여 만들기

```

<html>
<head>
  <meta charset="utf-8">
  <title>Elvis Presley</title>
</head>
<body>
<div>
  <div id="header">
    <h1><a href="https://www.facebook.com/elvis">Elvis Presley</a></h1>
  </div>

  <div id="container">
    <div>
      <div id="post-1">
        <h2><a href="https://ko.wikipedia.org/wiki/엘비스_프레슬리">
          Who is Elvis?</a></h2>

        <div class="entry">
          <p>엘비스 프레슬리는 20세기 가장 잘 알려진 미국 가수 중 한
            명이였다. 하나의 문화 아이콘으로, 엘비스의 음악들은
            세계적으로 선풍적인 인기를 끌었으며 "로큰롤의 제왕"으로
            불리곤한다. 그는 미시시피 주의 투펠로에서 태어났다.</p>
        </div>
      </div>

      <div id="post-2">
        <h2><a href="http://www.biography.com/people/elvis-presley-9446466">
          His music and life</a></h2>

        <div class="entry">
          <p>Presley received his first guitar as a gift
            from his mother on his 11th birthday in 1946
            and had his first taste of musical success
            a few years later when he won a talent show
            at Humes High School in Memphis. </p>
        </div>
      </div>
    </div>
    <div>
      <a href="">&laquo; Previous Entries</a>
    </div>
  </div>

  <div id="navigation">
    <ul>
      <li><h2>Archives</h2>
      <ul>
        <li><a href="http://www.elvisthemusic.com/">Elvis The Music</a></li>
        <li><a href="http://www.graceland.com/">Elvis Home Graceland</a></li>
        <li><a href="http://www.last.fm/music/Elvis+Presley">Free Listening, Video</a></li>
      </ul>
      </li>
    </ul>
  </div>

  <div id="footer">
    <p>Copyright 2015 Elvis</p>
  </div>
</div>
</body>
</html>

```

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Elvis Presley</title>
</head>
<body>
  <header>
    <h1><a href="https://www.facebook.com/elvis">Elvis Presley</a></h1>
  </header>

  <section>
    <article>
      <h2><a href="https://ko.wikipedia.org/wiki/엘비스_프레슬리">
        Who is Elvis?</a></h2>

      <p>엘비스 프레슬리는 20세기 가장 잘 알려진 미국 가수 중 한
        명이였다. 하나의 문화 아이콘으로, 엘비스의 음악들은
        세계적으로 선풍적인 인기를 끌었으며 "로큰롤의 제왕"으로
        불리곤한다. 그는 미시시피 주의 투펠로에서 태어났다.</p>
    </article>

    <article>
      <h2><a href=
        "http://www.biography.com/people/elvis-presley-9446466">
        His music and life</a></h2>

      <p>Presley received his first guitar as a gift
        from his mother on his 11th birthday in 1946
        and had his first taste of musical success
        a few years later when he won a talent show
        at Humes High School in Memphis.</p>
    </article>

    <nav>
      <a href="">&laquo; Previous Entries</a>
    </nav>
  </section>

  <nav>
    <ul>
      <li><h2>Archives</h2>
      <ul>
        <li><a href="http://www.elvisthemusic.com/">Elvis The Music</a></li>
        <li><a href="http://www.graceland.com/">Elvis Home Graceland</a></li>
        <li><a href="http://www.last.fm/music/Elvis+Presley">Free Listening, Video</a></li>
      </ul>
      </li>
    </ul>
  </nav>
  <footer>
    <p>Copyright 2015 Elvis</p>
  </footer>

</body>
</html>

```

1. 기존 HTML 태그로 작성

2. HTML5 시맨틱 태그로 작성  
바람직한 웹 페이지

# HTML5 문서 구조화 연습

- 문서를 구조화할 때 권장 사항
  - 웹 페이지 전체를 구조화 시맨틱 태그로 분할
  - 웹 페이지 전체 제목과 소개는 <header> 태그로 작성
  - 본문은 <section>으로 묶고,
  - 본문 내에 각 절이나 영역은 <article>로 작성
  - 링크나 메뉴들은 <nav> 태그로 작성
  - <header>, <section>, <article>, <aside> 등에는 헤딩 태그(<h1>~<h6>)를 이용하여 제목을 붙임
  - 배경 음악을 연주하는 <audio> 태그는 <header> 영역에 삽입
  - 문서의 모양(<header>, <section>, <article>, <aside>의 배치)은 CSS3 스타일 시트로 꾸미기

# HTML5 문서 구조화 연습용 샘플 웹 페이지

본문  
(<section>)

머리말  
(<header>)

목차  
(<nav>)

절  
(<article>)

절  
(<article>)

절  
(<article>)

노트  
(<aside>)

꼬리말  
(<footer>)

시맨틱 태그로 구조화 된 x
localhost/3/fig3-07.html

## 볼프강 아마데우스 모차르트

1756년 1월 27일 잘츠부르크에서 태어난 천재적인 오스트리아의 작곡가 모차르트(1756년 1월 27일 ~ 1791년 12월 5일)를 소개한다.



1770년대 초상화

### 목차

- [생애](#)
- [죽음](#)
- [음악](#)

### 생애

모차르트는 1756년 1월 27일 잘츠부르크에서 태어나서, 궁정 음악가였던 아버지에게 피아노와 바이올린을 배웠고, 다섯살 때 이미 작곡을 하기 시작했으며, 1764년에서 1765년 사이에 바흐로부터 처음으로 교향곡을 작곡하는 법을 배웠는데 이것이 모차르트가 수많은 교향곡을 남기는 계기가 되었다. 모차르트는 빈에서 1784년에 14세인 베토벤을 만나 베토벤을 교육시키는데 전념하기도 했다.

### 죽음

모차르트는 1791년 12월 5일 오전 0시 55분경에 갑자기 병으로 죽었으며 모차르트가 완성하지 못한 작품 레퀴엠은 프란츠 크사버 쥐스마이어(Franz Xaver Süssmayr)가 완성시켰다.

### 음악

오페라, 교향곡, 행진곡, 관현악용 무곡, 피아노 협주곡, 바이올린 협주곡, 교회용 성악곡, 칸타타, 미사곡 등 다양한 장르를 아우르며 600 여곡을 작곡하여 후대에 남겼다.

### 모차르트의 죽음에 얽힌 전설

모차르트의 장례식 날 비가 오고 천둥이 쳤다고 하나 New Groove에 따르면 사실은 구름 한 점 없는 쾌청한 날이었다고 한다.

2017년 10월 7일 작성, 위키피디아 참고

HTML5 문서 구조화 연습  
- HTML 텍스트

```
<!DOCTYPE html> <!-- 참고: https://ko.wikipedia.org/wiki/볼프강_아마데우스_모차르트 -->
<html>
<head><meta charset="utf-8">
<title>시맨틱 태그로 구조화 연습</title></head>
<body>
  <header>
    <h1>볼프강 아마데우스 모차르트</h1>
    <p>모차르트(1756년 1월 27일 ~ 1791년 12월 5일)는 1756년 1월 27일 잘츠부르크에서 태어난 천재적인 오스트리아의 작곡가를 소개한다.</p>
    <figure>
      
      <figcaption>1770년대 초상화</figcaption>
    </figure>
  </header>
  <nav>
    <h2>목차</h2>
    <ul>
      <li><a href="#life">생애</a></li>
      <li><a href="#death">죽음</a></li>
      <li><a href="#music">음악</a></li>
    </ul>
  </nav>
  <section>
    <article id="life">
      <h2>생애</h2>
      <p>모차르트는 1756년 1월 27일 잘츠부르크에서 태어나서, 궁정 음악가였던 아버지
      에게 피아노와 바이올린을 배웠고, 다섯살 때 이미 작곡을 하기 시작했으며, 1764년에서 1765년 사이에 바흐로부터 처음으로 교향곡을 작곡하는 법을 배웠는데
      이것이 모차르트가 수많은 교향곡을 남기는 계기가 되었다. 모차르트는 빈에서 1784년에 14세인 베토벤을 만나 베토벤을 교육시키는데 전념하기도 했다.
      </p>
    </article>
    <article id="death">
      <h2>죽음</h2>
      <p>모차르트는 1791년 12월 5일 오전 0시 55분경에 갑자기 병으로 죽었으며 모차르트가 완성하지 못한 작품
      레퀴엠은 프란츠 크사버 쥐스마이어(Franz Xaver Süssmayr)가 완성시켰다.</p>
    </article>
    <article id="music">
      <h2>음악</h2>
      <p>오페라, 교향곡, 행진곡, 관현악용 무곡, 피아노 협주곡, 바이올린 협주곡, 교회용 성악곡, 칸타타, 미사곡 등 다양한 장르를 아우르며
      600 여곡을 작곡하여 후대에 남겼다.</p>
    </article>
  </section>
  <aside id="legend">
    <h3>모차르트의 죽음에 얽힌 전설</h3>
    <p>모차르트의 장례식 날 비가 오고 천둥이 쳤다고 하나 New Groove에 따르면 사실은 구름 한 점 없는 쾌청한 날이었다고 한다.</p>
  </aside>
  <footer>
    <p>2017년 10월 7일 작성, 위키피디어 참고</p>
  </footer>
</body>
</html>
```

# 시맨틱 태그들

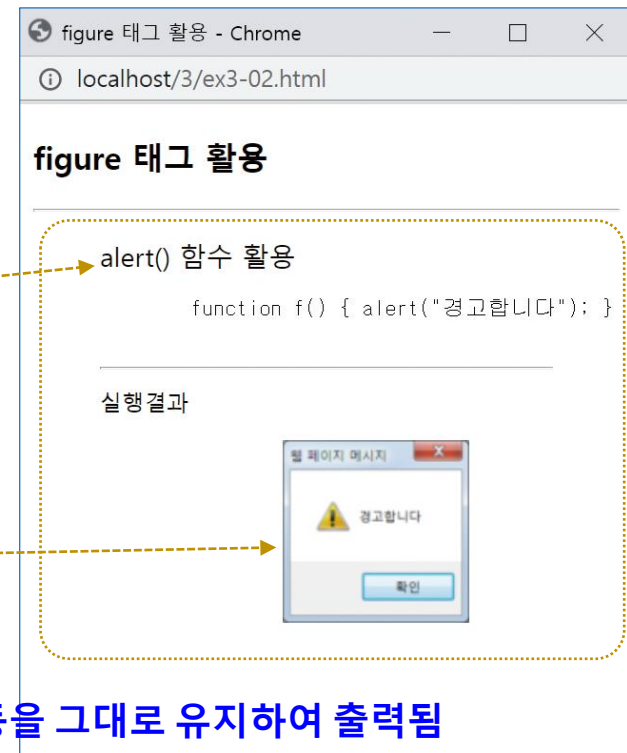
- 시맨틱 블록 태그
  - <figure>
    - 책이나 보고서 등 본문에 삽입하는 사진, 차트, 삽화, 소스 코드 등을 통상적으로 ‘그림’으로 표현
  - <details>와 <summary>
    - <details>는 상세 정보를 담는 시맨틱 블록 태그
    - <summary> 태그는 <details>로 구성되는 블록의 제목 표현
- 시맨틱 인라인 태그
  - <mark>
    - 중요한 텍스트임을 표시
  - <time>
    - 텍스트의 내용이 시간임을 표시
  - <meter>
    - 주어진 범위나 %의 데이터 량 표시
  - <progress>
    - 작업의 진행 정도 표시

# <figure> 태그 활용

'alert() 함수에 대한 설명'과 '실행 결과'를 하나의 그림으로 블록화하기 위해 <figure>를 사용한 예를 보여준다.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>figure 태그 활용</title>
</head>
<body>
<h3>figure 태그 활용</h3>
<hr>
<figure id="1-1">
  <figcaption>alert() 함수 활용</figcaption>
  <pre>
    <code>function f() { alert("경고합니다"); }</code>
  </pre>
  <hr>
  <small>실행결과</small>
  <pre>
    
  </pre>
</figure>
</body>
</html>
```

**<pre> 태그는 공백, 줄바꿈, 들여쓰기 등을 그대로 유지하여 출력됨**

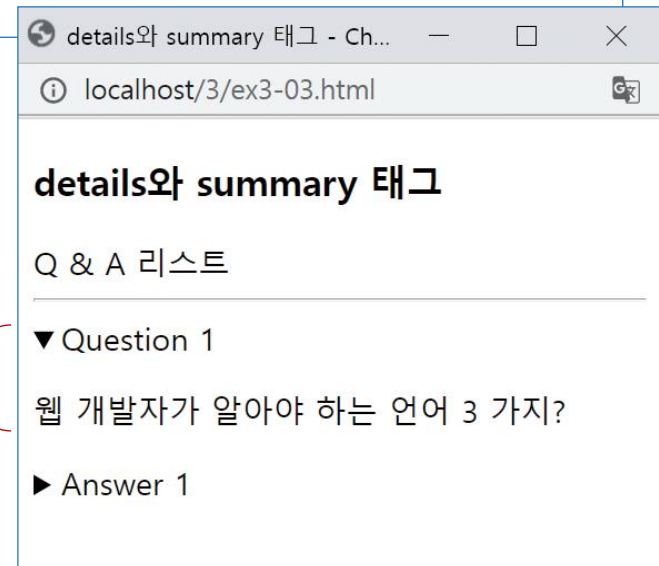
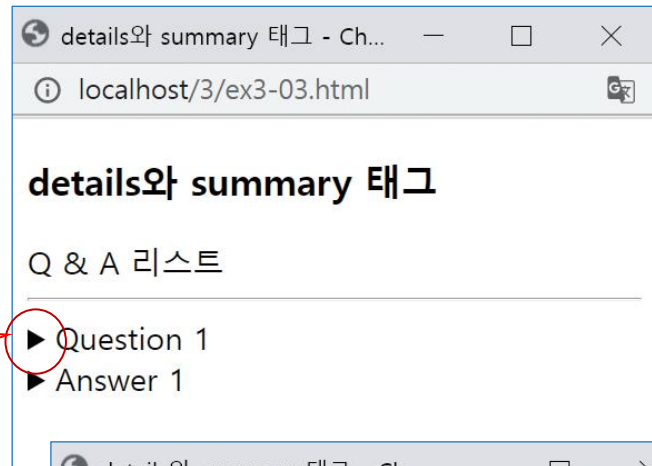


# <details>와 <summary> 활용

이 예제는 <details> 태그를 이용하여 Q&A 리스트를 만든 사례이다. 사용자가 핸들(▶)을 클릭하여 항목을 보거나 숨길 수 있다.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>details와 summary 태그</title>
</head>
<body>
<h3>details와 summary 태그</h3>
Q & A 리스트
<hr>
<details>
  <summary>Question 1</summary>
  <p>웹 개발자가 알아야 하는 언어 3 가지?</p>
</details>
<details>
  <summary>Answer 1</summary>
  <p>HTML5, CSS, Javascript</p>
</details>
</body>
</html>
```

핸들을 클릭하면  
아래와 같이 상세  
정보가 펼쳐짐

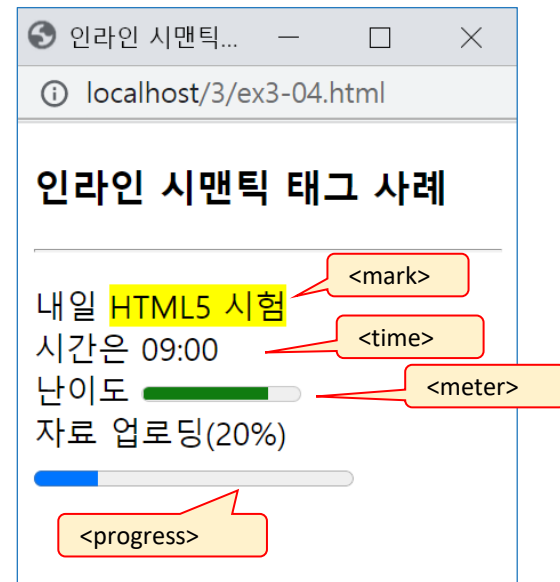




# 시맨틱 인라인 태그

4가지 인라인 시맨틱 태그의 사용 사례를 보여준다. <mark>의 텍스트는 브라우저에 따라 다르게 표현되며 크롬에서는 노란색 배경으로 출력된다.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>인라인 시맨틱 태그</title>
</head>
<body>
<h3>인라인 시맨틱 태그 사례</h3>
<hr>
<p>내일 <mark>HTML5 시험</mark><br>
시간은 <time>09:00</time><br>
난이도 <meter value="0.8" max="1.0">80%</meter><br>
자료 업로딩(20%)
  <progress value="2" max="10"></progress><br>
</p>
</body>
</html>
```



**<meter> 태그**는 low와 high 속성에 따라 값의 높고 낮음에 따라 색상을 다르게 지정하여 진행 상태를 표현할 수 있지만, **<progress> 태그**는 값의 상태만 전달

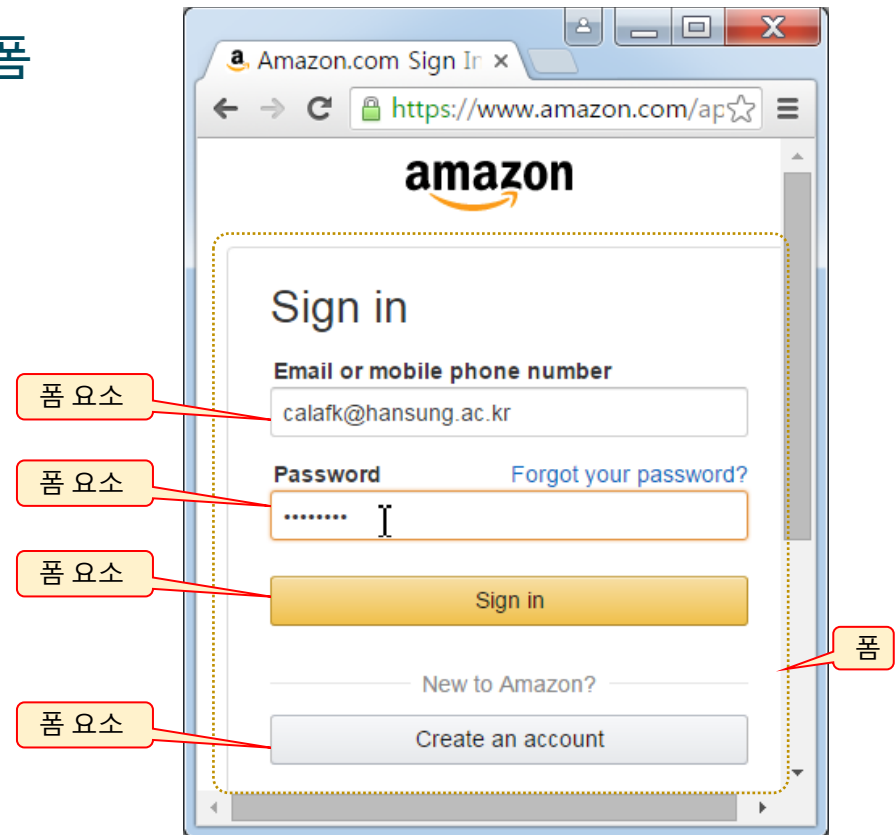
# HTML5에서 제거된 태그

다음 태그들은 문서의 시맨틱 구조를 저해한다는 이유로 HTML5에서 제거되었으므로 사용하지 말기 바람

<big>, <center>, <dir>, <font>, <tt>, <u>, <xmp>,  
<acronym>, <applet>, <basefont>, <frame>, <frameset>,  
<noframes>, <strike>

# 웹 폼

- 웹 폼
  - 웹 페이지에서 사용자 입력을 받는 폼
  - 로그인, 등록, 검색, 예약, 쇼핑 등
- 폼 요소
  - 폼을 만드는 다양한 태그
  - <input>, <textarea>, <select> 등



# 간단한 로그인 폼 만들기

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>로그인 폼</title></head>
<body>
<h3>로그인 폼</h3>
<hr>
<form name="fo" method="get">
사용자 ID : <input type="text" size="15" value=""><br>
비밀 번호 : <input type="password" size="15" value="">
<input type="submit" value="완료">
</form>
</body>
</html>
```

로그인 폼 - Chrome

localhost/3/ex3-05.html

### 로그인 폼

---

사용자 ID : Elvis

비밀 번호 : .....

# 폼 작성

- 폼 태그 <form> 태그로 둘러싸는 모양

```
<form name="fo" method="get">  
    ...  
</form>
```

- name 속성
  - 폼의 이름 지정
- action 속성
  - 폼 데이터를 처리할 웹 서버 응용프로그램의 이름
  - submit 버튼이 눌리면 브라우저는 action에 지정된 웹 서버 응용프로그램 실행 요청
  - 웹 서버 응용프로그램은 Java, JSP, PHP, C/C++ 등 다양한 언어로 작성
- method 속성
  - 폼 데이터를 웹 서버로 전송하는 형식
  - 대표적인 전송 방식 : GET, POST

# <form> 태그

```
<form action="웹 서버 응용 프로그램의 URL"
      enctype="데이터의 인코딩 타입"
      method="GET|POST"
      name="폼 이름"
      target="윈도우 이름">
  ... 여러 개의 폼 요소 작성
</form>
```

- action : 폼 데이터를 처리할 웹 서버 응용프로그램 이름 URL
- enctype : 폼 데이터를 웹 서버로 전송할 때 암호화 방식 지정
- method : 폼 데이터를 웹 서버에 전송하는 방식
- target : 웹 서버 응용프로그램으로부터 전송받은 데이터(HTML 문서나 이미지)를 출력할 윈도우 이름

# 네이버 검색 사례로 폼 전송 과정 이해

## ❶ 네이버 사이트에 접속한다.

검색어 창을 포함하는 폼 코드는 다음과 같다.

```
<form name="sform" action="https://search.naver.com/search.naver" method="get">  
  <input name="query" type="text">  
  <input type="submit" value="검색">  
</form>
```

구글: `action="https://www.google.com/search"`

입력 창에 "Elvis" 입력, "검색" 버튼을 누르면, 브라우저는 <form> 태그의 `action="https://search.naver.com/search.naver"`을 참고. search.naver.com 서버에 접속하여 search.naver 응용프로그램의 실행을 요구해야 한다는 것 확인

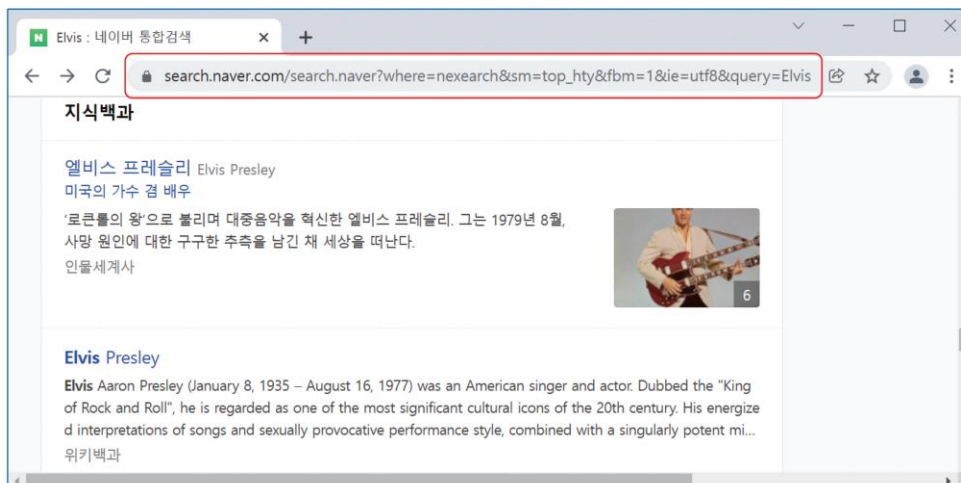
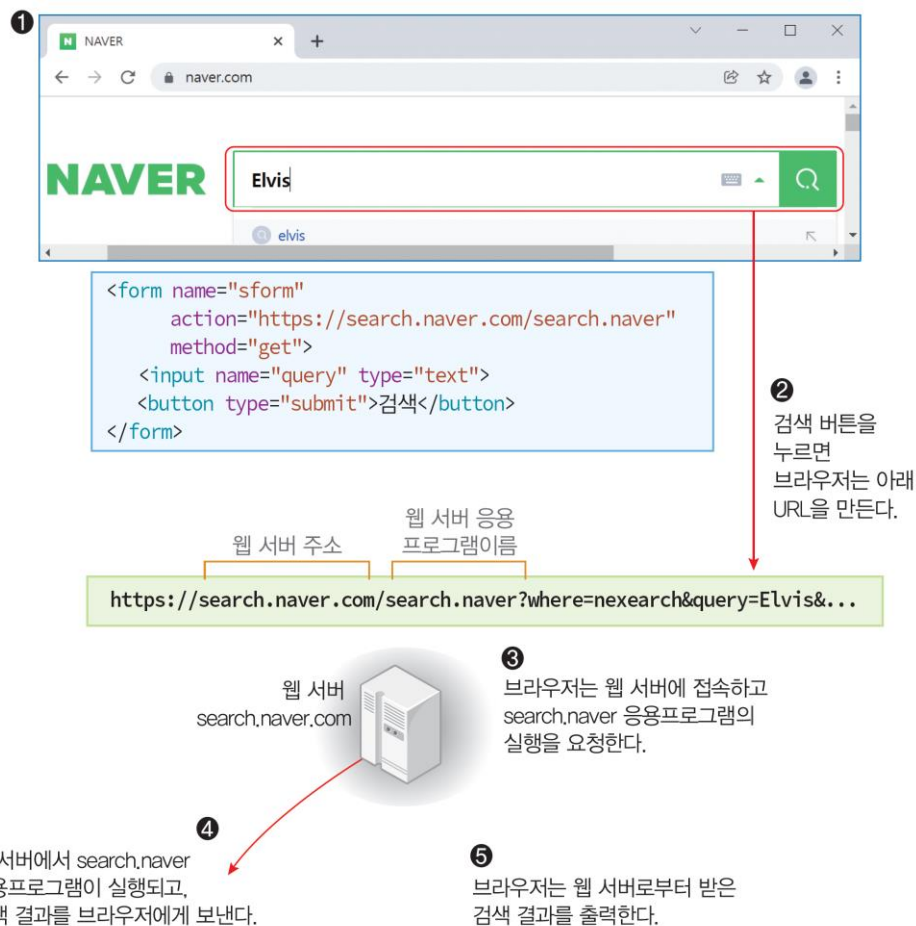
## ❷ 입력 창에 'Elvis'를 입력하고 검색 버튼을 누르면 웹 서버 응용프로그램에 보낼 폼 데이터를 만든다.

`https://search.naver.com/search.naver?... query=Elvis ...`

## ❸ 브라우저는 search.naver.com에 접속. search.naver 실행 요청. query=Elvis를 전달한다.

## ❹ 웹 서버에서 search.naver 응용프로그램이 실행되고, 검색 결과를 브라우저에게 보낸다.

## ❺ 브라우저는 검색 결과를 화면에 출력한다. HTML5 문서 구조화와 웹 폼





# 폼 요소의 종류

폼 요소	설명
<code>&lt;input type="text"&gt;</code>	한 줄 텍스트 입력 창
<code>&lt;input type="password"&gt;</code>	암호 입력을 위한 한 줄 텍스트 입력 창
<code>&lt;input type="button"&gt;</code>	단순 버튼
<code>&lt;input type="submit"&gt;</code>	웹 서버로 폼 데이터를 전송시키는 버튼
<code>&lt;input type="reset"&gt;</code>	입력된 폼 데이터를 초기화시키는 버튼
<code>&lt;input type="image"&gt;</code>	이미지 버튼
<code>&lt;input type="checkbox radio"&gt;</code>	체크박스과 라디오버튼
<code>&lt;select&gt;</code>	드롭다운 리스트를 가진 콤보박스
<code>&lt;input type="month week date time datetime-local"&gt;</code>	년, 월, 일, 시간 등의 시간 정보 입력 창
<code>&lt;input type="number range"&gt;</code>	스핀 버튼과 슬라이드바로 편리한 숫자 입력 창
<code>&lt;input type="color"&gt;</code>	색 입력을 쉽게 하는 컬러 다이얼로그
<code>&lt;input type="email url tel search"&gt;</code>	이메일, URL, 전화번호, 검색키워드 등 형식 검사 기능을 가진 텍스트 입력 창
<code>&lt;input type="file"&gt;</code>	로컬 컴퓨터의 파일을 선택하는 폼 요소
<code>&lt;button type="button reset submit"&gt;</code>	단순 버튼, reset, submit 버튼
<code>&lt;textarea&gt;</code>	여러 줄의 텍스트 입력 창

# 텍스트 입력

- `<input type="text">`
  - 한 줄 짜리 입력 창
- `<input type="password">`
  - 암호 입력 창
    - 사용자 입력 문자 대신 '\*' 등 다른 글자로 출력
- `<textarea>`
  - 여러 줄 입력 창

```
<input type="text|password"
      name="요소이름"
      maxlength="문자개수"
      size="문자개수"
      value="초기텍스트">
```

- `maxlength` : 입력할 수 있는 문자의 최대 개수
- `size` : 입력 창의 크기. 단위는 문자 개수

```
<textarea cols="열 개수"
          rows="행 개수"
          name="요소이름"
          wrap="OFF|HARD|SOFT">
      초기 출력될 텍스트
```

```
</textarea>
```

- `cols`, `rows` : 텍스트 입력창의 크기로 가로세로 문자 수(영어를 기준으로)
- `wrap` : 자동 줄바꿈 처리 지정

# 텍스트 입력

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>텍스트 입력 폼</title>
</head>
<body>
<h3>자기 소개서 작성</h3>
<hr>
<form>
  이름 : <input type="text" value=""><br>
  암호 : <input type="password" value="" maxlength="4"><br>
  자소서 : <textarea cols="20" rows="5">
            이곳에 자기소개서 작성
          </textarea>
</form>
</body>
</html>
```

텍스트 입력 폼 - Chr... — □ ×

localhost/3/ex3-06.html 🔑

### 자기 소개서 작성

---

이름 :

암호 :

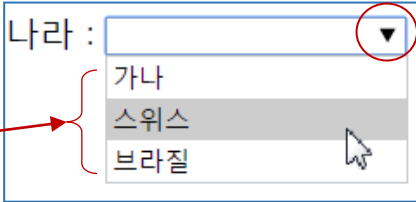
자소서 : 

저는 엘비스입니다. 저를 채용해 주시면, 최고의 회사를 만들겠습니다.

# 데이터 목록을 가진 텍스트 입력 창

- `<datalist>`
  - 목록 리스트를 작성하는 태그
    - `<option>` 태그로 항목 하나 표현
  - `<input type="text">`에 입력 가능한 데이터 목록 제고

나라 : `<input type="text" list="countries">`  
`<datalist id="countries">`  
    `<option value="가나">`  
    `<option value="스위스">`  
    `<option value="브라질">`  
`</datalist>`

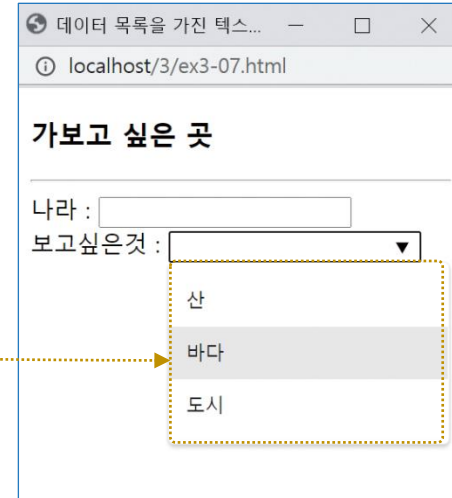
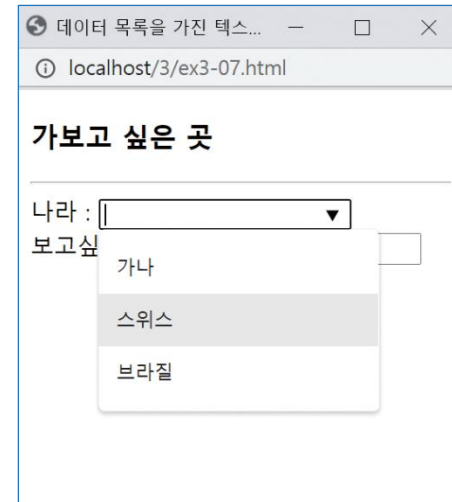
나라 :   
드롭다운 리스트

**`<datalist>` 태그는 `<select>` 태그와 다르게 목록에 없는 값을 입력할 수 있음**

# 데이터 목록을 가진 텍스트 입력

<datalist>를 사용하여 선택 목록을 제공하는 사례를 보인다.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>데이터 목록을 가진 텍스트 입력</title>
</head>
<body>
<h3>가보고 싶은 곳</h3>
<hr>
<form>
나라 : <input type="text" list="countries"><br>
    <datalist id="countries">
        <option value="가나">
        <option value="스위스">
        <option value="브라질">
    </datalist>
보고싶은것 : <input type="text" list="what"><br>
    <datalist id="what">
        <option value="산">
        <option value="바다">
        <option value="도시">
    </datalist>
</form>
</body>
</html>
```



# 텍스트/이미지 버튼 만들기

- `<input type="button|reset|submit|image" value="버튼의 문자열">`
- `<button type="button|reset|submit">버튼의 문자열</button>`

```
<input type="button|reset|submit|image"
      name="버튼 이름"
      value="문자열"
      src="이미지 URL">
```

- value : 버튼에 출력되는 문자열
- src : type="image"인 경우에만 필요한 것으로 이미지의 URL

```
<button type="button|reset|submit"
      name="버튼 이름"
      value="문자열">
```

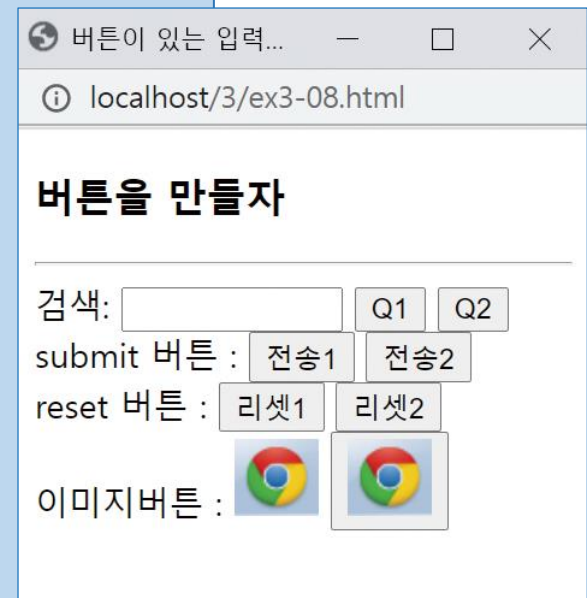
버튼 문자열이나 이미지

```
</button>
```

**<input> 태그는 스스로 닫는 태그이고, <button> 태그는 스스로 닫지 않는 태그이기 때문에 <button> 태그는 하위 태그들을 추가할 수 있음**

# 다양한 버튼 만들기

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>버튼이 있는 입력 폼</title>
</head>
<body>
<h3>버튼을 만들자</h3>
<hr>
<form>
  검색: <input type="text" size="10" value="">
        <input type="button" value="Q1">
        <button type="button">Q2</button><br>
  submit 버튼 : <input type="submit" value="전송1">
                  <button type="submit">전송2</button><br>
  reset 버튼 : <input type="reset" value="리셋1">
                 <button type="reset">리셋2</button><br>
  이미지버튼 : <input type="image" src="media/button.png"
                alt="이미지 버튼">
                <button type="button">
                  
                </button>
</form>
</body>
</html>
```



# 선택형 입력 : 체크박스과 라디오버튼

- `<input type="checkbox">`
  - 체크박스 만들기
- `<input type="radio">`
  - 라디오버튼 만들기
  - name 속성 값이 같은 라디오버튼들이 하나의 그룹 형성

```
<input type="checkbox|radio"  
      name="요소 이름"  
      value="요소 값"  
      checked>
```

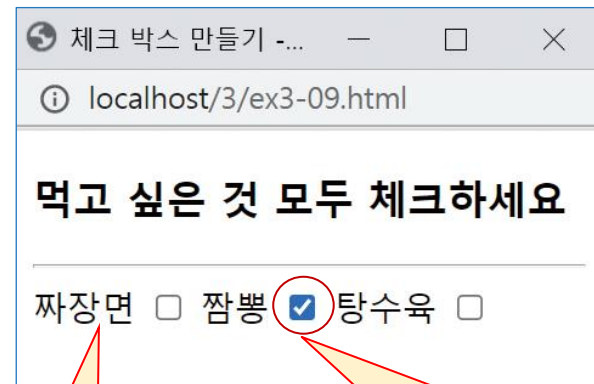
- value : 폼 요소가 선택된 상태일 때, 웹서버에 전송되는 값
- checked : 이 속성이 있으면, 초기에 선택 상태로 출력



# 체크박스 만들기

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>체크 박스 만들기</title>
</head>
<body>
<h3>먹고 싶은 것 모두 체크하세요</h3>
<hr>
<form>
  짜장면 <input type="checkbox" value="1">
  짬뽕 <input type="checkbox" value="2" checked>
  탕수육 <input type="checkbox" value="3">
</form>
</body>
</html>
```

캡션



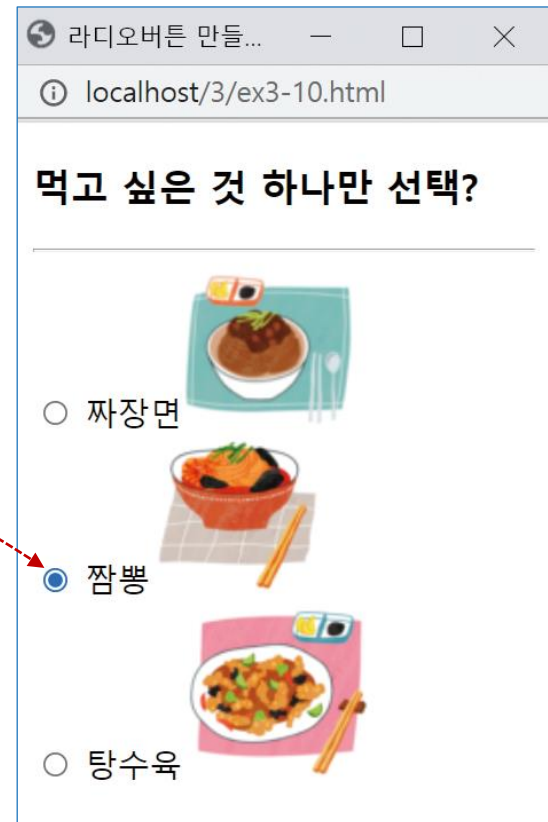
캡션

checked 속성이 사용되면  
초기에 체크된 상태로 출력

# 라디오버튼 만들기

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>라디오버튼 만들기</title>
</head>
<body>
<h3>먹고 싶은 것 하나만 선택?</h3>
<hr>
<form>
  <input type="radio" name="china" value="1">
    짜장면<br>
  <input type="radio" name="china" value="2" checked="">
    짬뽕<br>
  <input type="radio" name="china" value="3">
    탕수육
</form>
</body>
</html>
```

같은 name을 가진  
라디오버튼 중 하나만  
선택 가능



# 선택형 입력 : 콤보 박스

- `<select>`
  - 드롭다운 리스트에 목록 출력, 목록을 선택하는 입력 방식
    - `<option>` 태그로 항목 하나 표현

```
<select name="요소 이름"
        size="개수"
        multiple>
```

**`<select>` 태그는 목록에 있는 값만 선택할 수 있음**

여러 개의 `<option>...</option>` 태그로 선택 항목 만들기

```
</select>
```

- `size` : 콤보박스 창 크기(보일 수 있는 최대 항목 개수). 디폴트는 1
- `multiple` : 이 속성이 있으면 다수 항목 선택 가능

```
<option value="옵션 값"
        selected>
```

항목 문자열

```
</option>
```

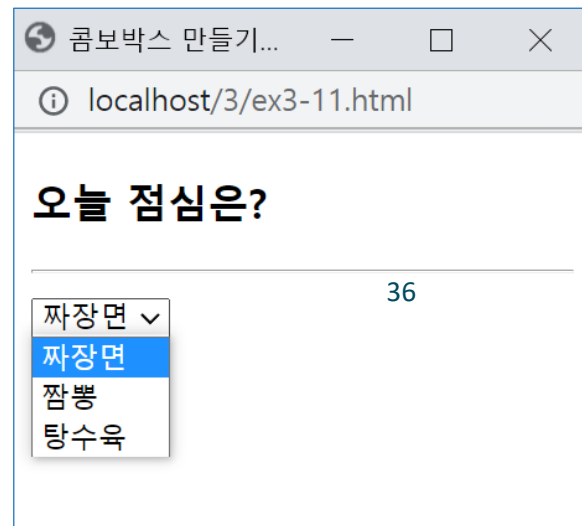
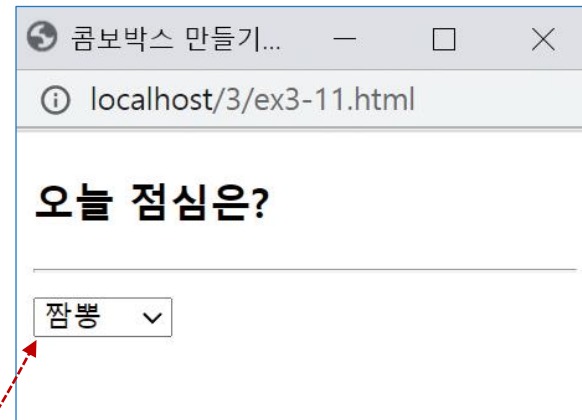
- `value` : 이 항목이 선택되었을 때 웹 서버에 전송되는 값
- `selected` : 이 속성이 있으면 초기에 선택 상태로 출력

# 콤보박스 만들기

짜장면, 짬뽕, 탕수육의 3개의 선택 항목을 가지는 콤보박스를 보여준다.

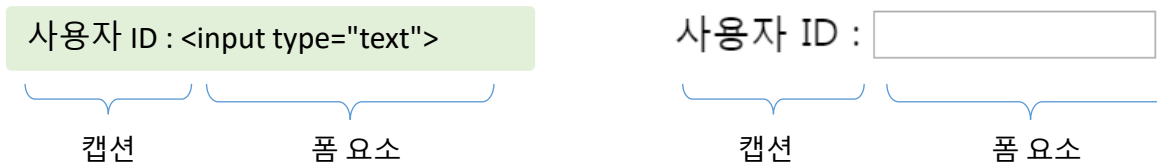
```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>콤보박스 만들기</title>
</head>
<body>
<h3>먹고 싶은 음식은?</h3>
<hr>
<form>
  <select name="china">
    <option value="1">짜장면</option>
    <option value="2" selected>짬뽕</option>
    <option value="3">탕수육</option>
  </select>
</form>
</body>
</html>
```

selected 속성이  
사용되면 선택 상태

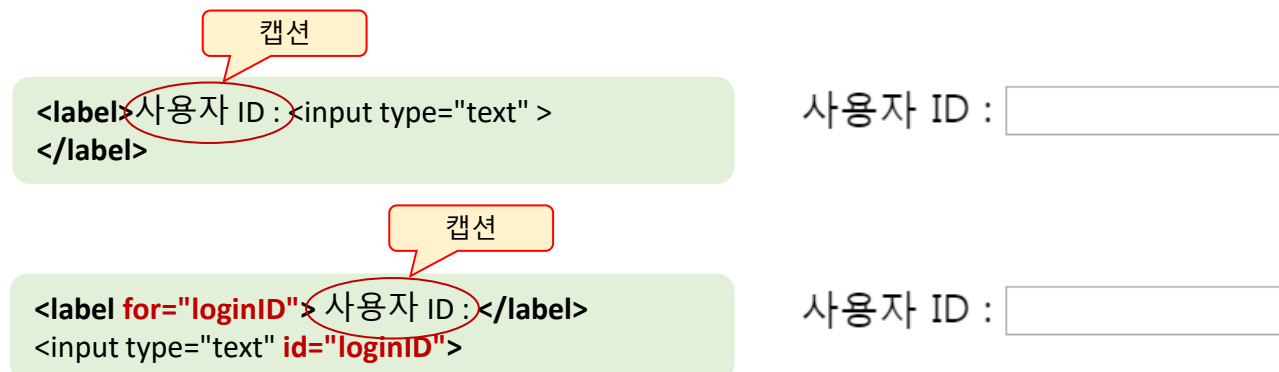


# <label>로 폼 요소의 캡션 만들기

- 캡션이란?

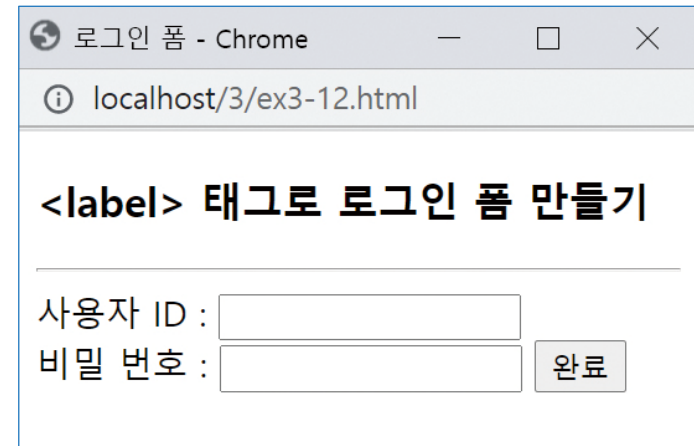


- <label> 태그로 캡션과 폼 요소를 한 단위로 묶음
  - 캡션 텍스트를 명료하게 하는 장점
  - 2 가지 방법



# <label> 태그로 로그인 폼 만들기

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>로그인 폼</title>
</head>
<body>
<h3>로그인 폼</h3>
<hr>
<form name="fo" method="get">
  <label>사용자 ID : <input type="text" size="15" value="">
  </label><br>
  <label for="pass">비밀 번호 : </label>
  <input id="pass" type="password" size="15" value="">
  <input type="submit" value="완료">
</form>
</body>
</html>
```



The screenshot shows a web browser window titled "로그인 폼 - Chrome" with the address bar displaying "localhost/3/ex3-12.html". The page content displays the title "<label> 태그로 로그인 폼 만들기" followed by a horizontal line. Below the line, there are two input fields: "사용자 ID :" followed by a text input box, and "비밀 번호 :" followed by a password input box. To the right of the password input box is a button labeled "완료".

# 선택형 요소의 캡션을 <label>로 감싸기

- 선택형 요소에 <label> 태그 사용
  - 캡션 텍스트나 이미지를 선택해도 폼 요소를 선택한 것으로 처리

```
<label>짜장면<input type="radio" name="china" value="1">
</label>
```

캡션

짜장면 ☐

```
<label for="china">짜장면</label>
<input type="radio" name="china" id="china" value="1">
```

짜장면 ☐

```
<label>
  <input type="radio" name="china" value="1">
  짜장면
</label>
```

캡션

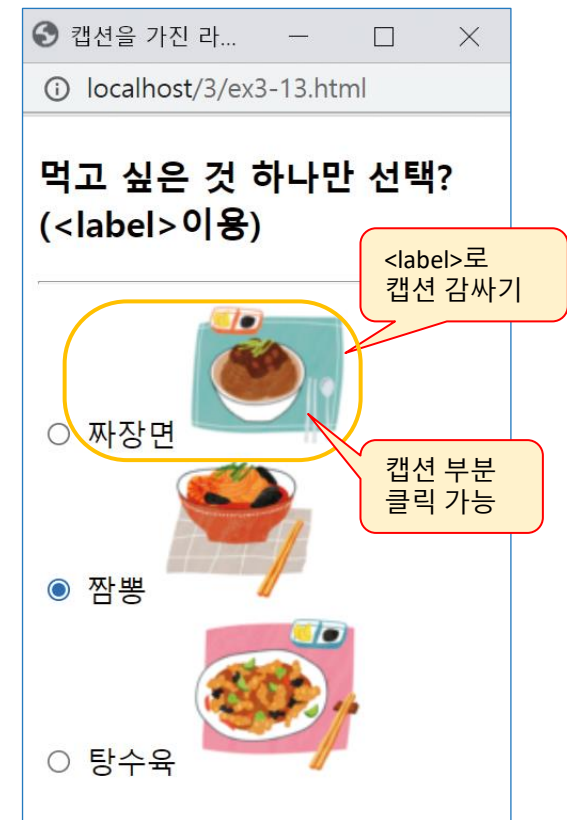
캡션



짜장면

# <label>로 라디오버튼에 캡션 만들기

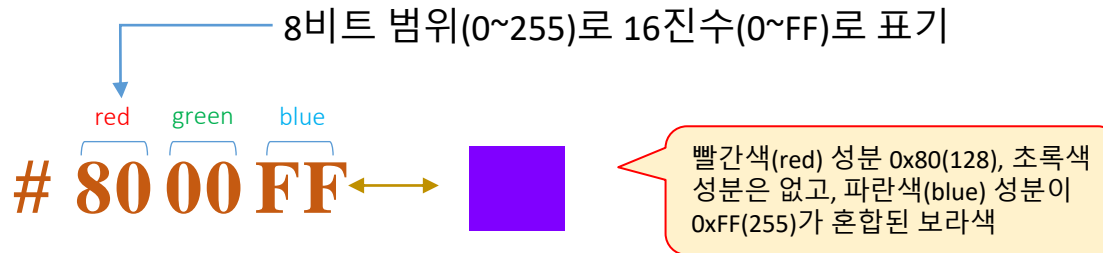
```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>캡션을 가진 라디오버튼</title>
</head>
<body>
<h3>먹고 싶은 것 하나만 선택?(&lt;label&gt;이용)</h3>
<hr>
<form>
  <label>
    <input type="radio" name="china" value="1">
    짜장면 
  </label><br>
  <label>
    <input type="radio" name="china" value="2" checked>
    짬뽕 
  </label><br>
  <label>
    <input type="radio" name="china" value="3">
    탕수육 
  </label>
</form>
</body>
</html>
```











# HTML에서의 색 표현

- 색 코드 - #rrggbb
  - rr은 빨간색, gg는 초록색, bb는 파란색 농도



- 색 이름과 색 코드 샘플

이름	코드	색	이름	코드	색
brown	#A52A2A		deepskyblue	#00BFFF	
blueviolet	#8A2BE2		gold	#FFD700	
darkorange	#FF8C00		olivedrab	# 6B8E23	

# 색 입력 폼

- `<input type="color">`
  - 컬러 다이얼로그 출력, 사용자로부터 색 선택
  - 사용 예

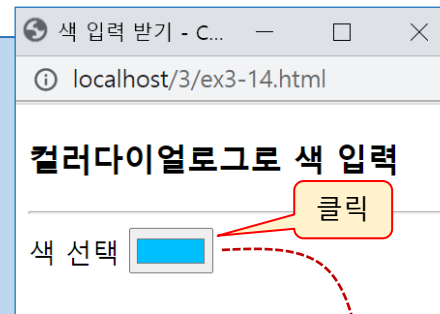
```
<input type="color" value="#00FF00">
```

value는 초기 색

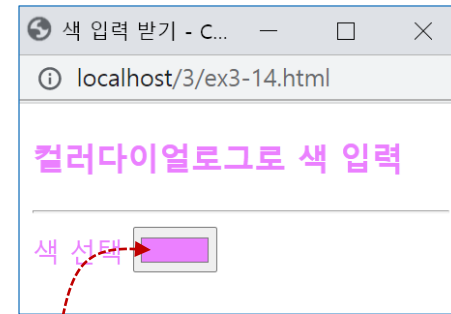
# 컬러 다이얼로그로 색 입력 응용

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>색 입력</title>
</head>
<body>
<h3>컬러다이얼로그로 색 입력</h3>
<hr>
<form>
  색 선택
  <input type="color" value="#00BFFF"
    onchange="document.body.style.color=this.value">
</form>
</body>
</html>
```

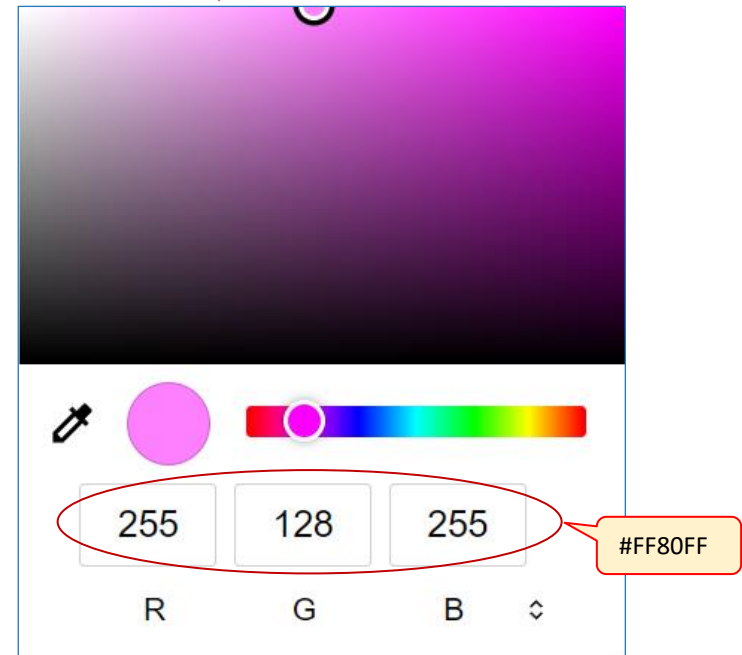
선택한 색을 브라우저의  
바탕 글자에 적용하는 자  
바스크립트 코드



value="#00BFFF" 색은  
컬러 다이얼로그의 초기 색



사용자가 #FF80FF 색 선택



# 시간 정보 입력 폼 요소

- `<input type="month|week|date|time|datetime-local">`
  - 시간 정보만 입력 가능한 폼 요소


type 속성값	입력 데이터	value의 형식
<code>&lt;input type="month"&gt;</code>	년/월	2016-09
<code>&lt;input type="week"&gt;</code>	년/몇 번째 주	2016-W16
<code>&lt;input type="date"&gt;</code>	년/월/일	2016-09-01
<code>&lt;input type="datetime-local"&gt;</code>	해당 나라의 시간 년/월/일/오전, 오후/시/분/초/100분의 1초	2016-09-01T21:30:10.32
<code>&lt;input type="time"&gt;</code>	시/ 분	21:30

# 폼 요소 작성 예

(a) 달 입력

<input type="month" value="2022-03">

2022년 03월 


2022년 03월  버튼

2022			
1	2	3	4
5	6	7	8
9	10	11	12


삭제 이번 달

(b) 주 입력

<input type="week" value="2022-W09">

2022, 09번째 주 

2022년도의 9번째 주를 보여줌

2022, 09번째 주  버튼

2022년 03월 ▾ ↑ ↓


주	일	월	화	수	목	금	토
9	27	28	1	2	3	4	5
10	6	7	8	9	10	11	12
11	13	14	15	16	17	18	19
12	20	21	22	23	24	25	26
13	27	28	29	30	31	1	2
14	3	4	5	6	7	8	9

삭제 이번 주

(c) 날짜 입력

`<input type="date" value="2022-03-01">`

2022-03-01 

2022-03-01  버튼


2022년 03월 ▾ ↑ ↓

일	월	화	수	목	금	토
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9


삭제 오늘

(d) 로컬 날짜시간 입력

`<input type="datetime-local" value="2022-03-01T21:30:10.32">`

2022-09-01 오후 09:30:10.320 

9시 30분 10초 32

2022-09-01 오후 09:30:10.320  버튼

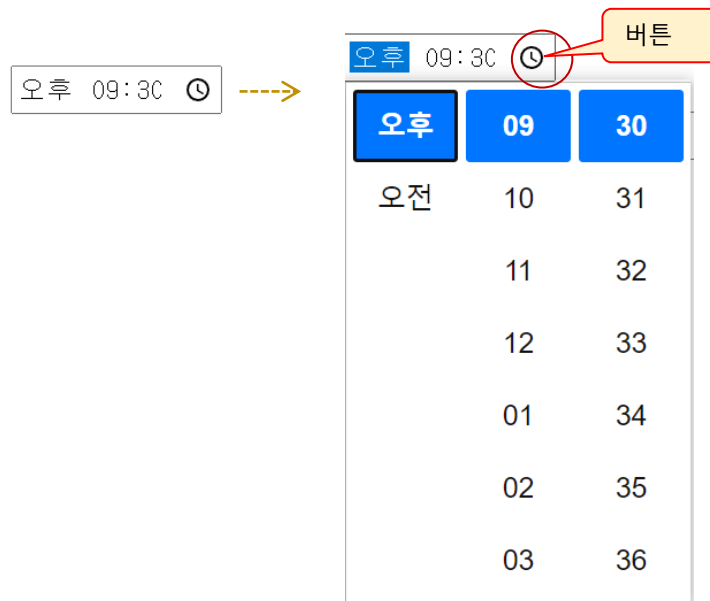
2022년 09월 ▾ ↑ ↓

일	월	화	수	목	금	토	09	30	10	320	오후
28	29	30	31	1	2	3	10	31	11	400	오전
4	5	6	7	8	9	10	11	32	12	500	
11	12	13	14	15	16	17	12	33	13	600	
18	19	20	21	22	23	24	01	34	14	700	
25	26	27	28	29	30	1	02	35	15	800	
2	3	4	5	6	7	8	03	36	16	900	

삭제 오늘

(e) 시간 입력

`<input type="time" value="21:30">`



# 시간 정보 입력 폼 요소 활용

이 예제는 시간 정보를 입력 받는 HTML5 폼 요소들의 사례를 보인다.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>시간 정보를 입력하는 폼 활용</title>
</head>
<body>
<h3>시간 정보 입력 HTML5 폼 요소들</h3>
초기 세팅 : 2022년 3월 1일 밤 9시 30분(10초 32)<br>
시간을 변경해 보세요
<hr>
<form>
<pre>
month : <input type="month" value="2022-03"><br>
week : <input type="week" value="2022-W09"><br>
data : <input type="date" value="2022-03-01"><br>
time : <input type="time" value="21:30"><br>
local: <input type="datetime-local"
        value="2022-03-01T21:30:10.32"><br>
</pre>
</form>
</body>
</html>
```


HTML5 문서 구조화와 웹 폼


시간 정보를 입력하는 폼 활용 - ...


localhost/3/ex3-15.html


### 시간 정보 입력 HTML5 폼 요소들


초기 세팅 : 2022년 3월 1일 밤 9시 30분(10초 32)  
시간을 변경해 보세요

month : 2022년 03월 

week : 2022, 09번째 주 

data : 2022-03-01 

time : 2022년 03월 

local: 

일	월	화	수	목	금	토
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

삭제 오늘



# 생일 날짜 입력 받기

시간 정보를 입력 받는 `<input>` 태그를 사용하여 생일 날짜를 입력 받아라.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>시간 정보 입력 응용</title>
</head>
<body>
<h3>생일 축하합니다</h3>
<hr>
당신의 생일은 2003년 2월 9일입니다. 틀리면 수정하시고
파티 시간과 장소를 입력하세요.
<hr>
<form>
<table>
<tr><td>생일</td>
<td><input type="date" value="2003-02-09"></td></tr>
<tr><td>생일파티시간</td>
<td><input type="time"></td></tr>
<tr><td>생일파티장소</td>
<td><input type="text"></td></tr>
</table>
</form>
</body>
</html>
```

HTML5 문서 구조화와 웹 폼

시간 정보 입력 응용 - Chrome

localhost/3/ex3-16.html

## 생일 축하합니다

당신의 생일은 2003년 2월 9일입니다.  
틀리면 수정하시고 파티 시간과 장소를  
입력하세요.

생일	<input type="text" value="2003-02-09"/>
생일파티시간	<input type="text" value="-- : --"/>
생일파티장소	<input type="text"/>

시간 정보 입력 응용 - ...

localhost/3/ex3-16.html

## 생일 축하합니다

당신의 생일은 2003년 2월 9일입니다.  
틀리면 수정하시고 파티 시간과 장소를  
입력하세요.

생일	<input type="text" value="2000-05-20"/>
생일파티시간	<input type="text" value="오후 07:30"/>
생일파티장소	<input type="text" value="우리집"/>

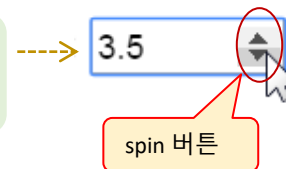
버튼을 클릭하여 분 입력

# 스핀버튼과 슬라이드 바로 편리한 숫자 입력

- `<input type="number">`
  - 스펀버튼으로 정교한 값 입력

```
<input type="number" min="0.0" max="10.0"
      step="0.5">
```

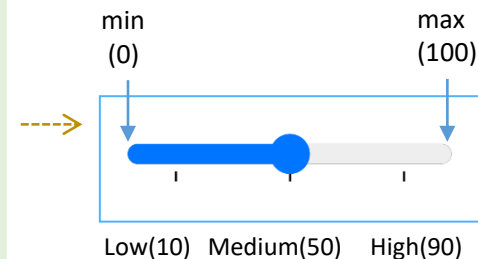
(a) spin 버튼 클릭 시 0.0~10.0 사이에서 0.5씩 증감



- `<input type="range">`
  - 슬라이드 바로 대략적인 값 입력

```
<input type="range" min="0" max="100"
      list="temperatures">
<datalist id="temperatures">
  <option value="10" label="Low">
  <option value="50" label="Medium">
  <option value="90" label="High">
</datalist>
```

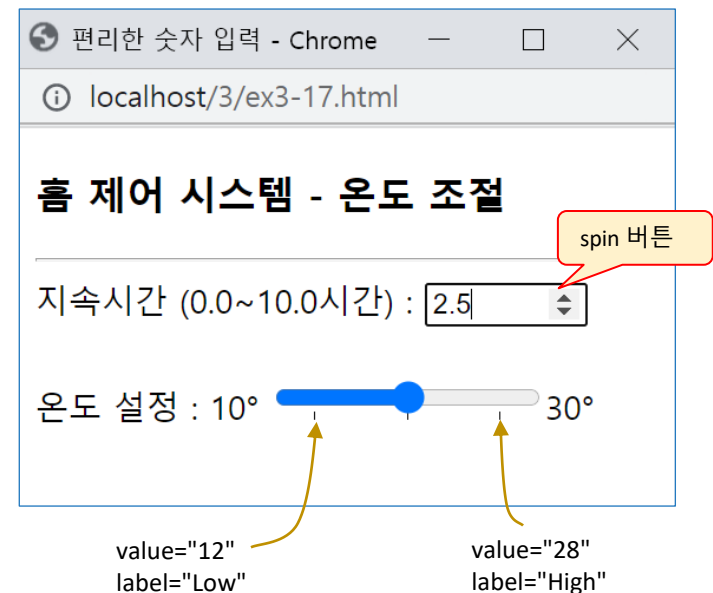
(b) 슬라이드 바를 움직여 0~100사이의 값 입력



# <input type="number|range">로 편리한 숫자 입력

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>편리한 숫자 입력</title></head>
<body>
<h3>홈 제어 시스템 - 온도 조절</h3>
<hr>
<form>
지속시간 (0.0~10.0시간) :
<input type="number" min="0.0" max="10.0"
step="0.5"><br><br>
온도 설정 :10&deg;
<input type="range" min="10" max="30"
list="temperatures">30&deg;
<datalist id="temperatures">
<option value="12" label="Low">
<option value="20" label="Medium">
<option value="28" label="High">
</datalist>
</form>
</body>
</html>
```

3개의 눈금



# 입력할 정보의 힌트 보여주기

- placeholder 속성에 사용자가 입력할 데이터 힌트 주기

이메일 주소:

```
<input type="email" placeholder="id@host">
```

이메일 주소:

# 형식을 가진 텍스트 입력

- `<input type="email">`, email 주소 입력

이메일 주소 :

```
<input type="email" placeholder="id@host">  
<input type="submit" value="전송">
```

이메일 주소: name@domain.com

전송

(a) 초기 화면



이메일 주소 : happy.com

전송

사용자가 입력한 내용

오류메시지

! 이메일 주소에 '@'를 포함해 주세요. 'happy.com'에 '@'가 없습니다.

- (b) 이메일 주소를 잘못 입력하고 '전송' 버튼을 클릭하여 전송한 경우

- `<input type="url">`, URL 입력

```
<input type="url" placeholder="http://">
```

- `<input type="tel">`, 전화번호 입력

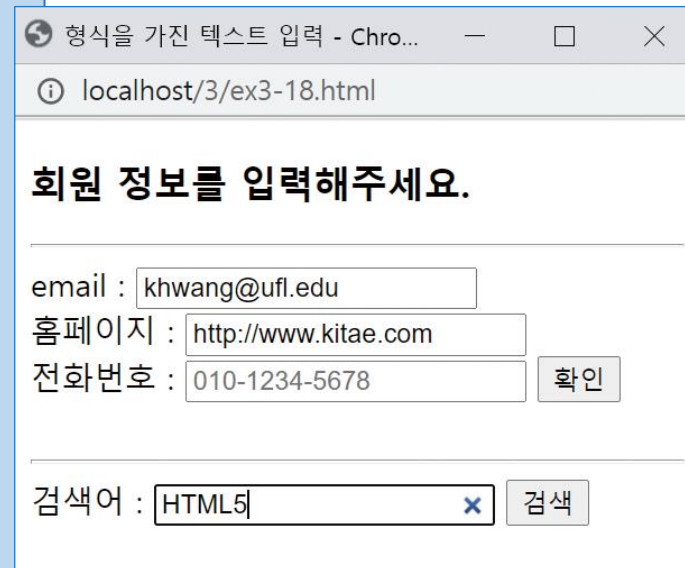
```
<input type="tel" placeholder="010-1234-5678">
```

- 검색어 입력, `<input type="search">`

```
<input type="search" placeholder="검색어">
```

# 형식을 가진 텍스트 입력

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>형식을 가진 텍스트 입력</title>
</head>
<body>
<h3>회원 정보를 입력해주세요.</h3>
<hr>
<form>
  email : <input type="email" placeholder="id@host"><br>
  홈페이지 : <input type="url" placeholder="http://"><br>
  전화번호 : <input type="tel" placeholder="010-1234-5678">
    <input type="submit" value="확인"><br><br>
</hr>
  검색어 : <input type="search" placeholder="검색어">
    <input type="button" value="검색">
</form>
</body>
</html>
```



The screenshot shows a web browser window titled "형식을 가진 텍스트 입력 - Chro...". The address bar displays "localhost/3/ex3-18.html". The page content includes the heading "회원 정보를 입력해주세요." followed by a form with three input fields: "email" (containing "khwang@ufl.edu"), "홈페이지" (containing "http://www.kitae.com"), and "전화번호" (containing "010-1234-5678"). A "확인" button is next to the phone number field. Below this is a search section with a label "검색어 :", an input field containing "HTML5", and a "검색" button.

# 폼 요소의 그룹핑

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>폼 입력 그룹으로 묶기</title>
</head>
<body>
<h3>회원 정보를 입력해주세요.</h3>
<hr>
<form>
  <fieldset>
    <legend>회원정보</legend>
    이메일 : <input type="email"><br>
    홈페이지 : <input type="url"><br>
    전화번호 : <input type="tel">
  </fieldset>
  <small>질문 : Tel. 010-111-1111</small>
</form>
</body>
</html>
```

폼 입력 그룹으로 묶기 -...

localhost/3/ex3-19.html

회원 정보를 입력해주세요.

회원정보

이메일 :

홈페이지 :

전화번호 :

질문 : Tel. 010-111-1111

그룹 박스

# Q&A

