

# 웹프로그래밍

## **I. HTML**

1. 웹프로그래밍 시작하기	4
2. HTML 문서작성 기본기 익히기	14
3. 하이퍼링크로 문서 연결하기	21
4. 가지런히 정렬된 목록 만들기	24
5. 웹 문서에 테이블을 넣어봅시다	30
6. 인터랙티브 웹의 시작! 폼 만들기	35

## **II. CSS**

1. CSS의 개요와 기본 사용법	51
2. CSS 선택자	63
3. CSS 스타일 속성	81
4. 레이아웃을 위한 스타일	90

## **III. JavaScript**

1. JavaScript 시작하기	99
2. 자료형과 연산자	102
3. 조건문	109
4. 반복문	118
5. 함수	125

HTML

# 1. 웹프로그래밍 시작하기

본격적으로 웹프로그래밍에 대해 알아보기 전에 웹프로그래밍을 위한 준비를 하겠습니다. 먼저 웹프로그래밍에 필요한 기술들에 대해 간단히 알아보고 웹페이지를 만들기 위한 도구를 설치하겠습니다.

## 1.1 웹프로그래밍에 필요한 기술

현재 많은 웹브라우저가 웹 문서를 작성하는 기술인 HTML5를 거의 대부분 지원하고 있지만 화면에 표시되는 형태는 모든 웹브라우저가 동일하게 지원하는 것은 아니며 웹브라우저마다 지원하는 부분이 조금씩 다를 수 있습니다. 그래서 웹사이트를 개발할 때도 여러 가지의 브라우저를 활용해서 테스트 하고 있습니다. 우리도 Chrome과 Edge 등의 브라우저를 사용하여 실습하겠습니다.

### 1.2.1 HTML(Hypertext Markup Language)

하이퍼텍스트(Hyper Text)와 마크업(Markup)은 HTML의 중요한 개념으로 하이퍼텍스트는 문서 내에 존재하는 콘텐츠에 링크 설정이 가능하여 문서가 서로 연결되어 있다는 의미로 실제 웹 문서의 특정 텍스트나 이미지 등을 클릭하면 다른 문서로 연결되어 이동 할 수 있게 됩니다. 마크업은 문서 내의 특정 내용에 특별히 지시하는 문장을 추가하여 기본 정보에 추가적인 정보를 표시 할 수 있다는 의미로 글꼴의 색상을 지정하거나 특정 내용을 묶어 하나의 문단으로 출력하기 위해 사용되는 일련의 문자나 기호들을 마크업 이라 합니다. 이렇게 기본적인 내용이 화면에 어떻게 출력될지 지정하는 마크업을 태그라고 부릅니다.

### 1.2.2 CSS(Cascading Style Sheet)

스타일 시트는 HTML 태그만 사용해 웹 문서를 구성할 때 화면 배치나 디자인 적인 요소를 표현하는 한계를 보완하기 위해 개발된 언어입니다.

예전에는 테이블을 이용해 웹페이지 레이아웃을 구성하는 경우가 많이 있었지만 요즘은 웹 표준과 웹 페이지에 대한 시맨틱(Semantic) 요소가 대두되면서 테이블을 사용한 레이아웃 기법은 더 이상 사용하지 않습니다. 또한 테이블 태그를 이용한 기법은 페이지 레이아웃을 변경할 때 많은 시간과 노력이 필요하거나 아예 다시 만들어야 하는 상황이 빈번히 발생 했으나 HTML5와 CSS를 이용한 레이아웃 기법은 HTML 태그를 수정하지 않고 CSS 코드 수정만으로 새로운 레이아웃을 쉽게 적용할 수 있는 장점을 가지고 있습니다.

HTML이 문서의 구조(틀)를 정의한다면 CSS는 문서의 레이아웃 즉 디자인 적인 요소를 정의(표현)하기 위해 사용됩니다.

### 1.2.3 XML(eXtensible Markup Language)

확장 가능한 마크업 언어라는 단어의 의미에서 알 수 있듯이 기존에 존재하지 않는 새로운 마크업 언어를 개발하기 위한 표준입니다. HTML은 이미 정의된 태그만을 사용하여 문서를 작성해야 하지만 XML은 정해진 태그가 없기 때문에 특정 단체나 프로그래머가 새로운 태그를 정의하여 사용합니다. 또한 XML은 화면에 어떻게 표시할 지에 관심을 두지 않고 태그에 의해 데이터에 특정 의미를 부여함으로써 구조화된 문서를 작성하여 데이터를 교환할 목적으로 설계된 마크업 언어입니다.

### 1.2.4 Javascript

넷스케이프 커뮤니케이션스사와 썬마이크로시스템즈사가 함께 개발한 웹브라우저에서 실행되는 객체 지향(객체기반) 스크립트 언어로 1995년에 네스케이프 2.0에 탑재되고 1996년 IE 3.0에 탑재된 이후 웹브라우저 표준 스크립트 언어로 정착되었으며 현재는 모든 주요 웹브라우저에서 지원하는 핵심 스크립트 언어입니다.

자바스크립트가 썬마이크로시스템즈와 함께 개발된 언어이기 때문에 썬마이크로시스템즈가 개발한 자바와 아주 밀접한 관계가 있는 것처럼 오해(?)를 하고 있는 사람들이 있는데 자바스크립트와 자바는 문법이 약간 다른 것과 둘 다 웹브라우저에서 실행 될 수 있다는 공통점이 있기 하지만 이 두 언어는 서로 완전히 다른 언어입니다.

새로운 웹 표준인 HTML5에서 자바스크립트는 HTML5의 API를 다루는 핵심 언어로 새롭게 부상하게 되었습니다.

### 1.2.5 jQuery

자바스크립트 라이브러리(jQuery, Prototype, dojo) 중의 하나로 2006년 존 레식에 의해 공식으로 소개 되었습니다. jQuery는 자바스크립트를 단순화하여 편리하게 사용할 수 있도록 설계된 라이브러리이며 브라우저 호환성(크로스 브라우징)이 아주 뛰어난 인기 있는 자바스크립트 라이브러리입니다. jQuery 외에 요즘 인기 있는 자바스크립트 라이브러리로 AngularJS, Vue.js, React 등이 있습니다.

### 1.2.6. node.js

웹 개발은 크게 프론트엔드(Front-end) 기술과 백엔드(Back-end) 기술로 나뉩니다. 프론트엔드 기술은 웹 브라우저를 통해 사용자가 볼 수 있는 웹페이지 디자인이나 버튼 기능 등을 구현하는 기술을 의미합니다. 이에 반해 백엔드 기술은 사용자가 직접 눈으로 볼 수 없고 서버 쪽에 숨겨져 있는 기술을 의미합니다. 이 기술은 웹브라우저가 아닌 서버에서 실행되는 프로그래밍 기술로 인터넷을 통해 사용자가 요청하면 그 요청에 따라서 서버에서 동적으로 처리하는 기술을 의미합니다. 그래서 프론트엔드 기술을 화면 단 기술 혹은 클라이언트 사이드(Client Side) 기술이라고 부르며 백엔드 기술을 서버 단 혹은 서버 사이드(Server Side)기술이라 부릅니다. 우리가 앞에서 알아본 HTML, CSS, JavaScript, jQuery 등이 대표적인 프론트엔드 기술이며, DB나 서버를 다루는 JSP, Servlet, PHP, Python의 Django와 Flask, asp.net 등이 바로 백엔드를 다루는 기술입니다.

앞에서도 언급한 것처럼 자바스크립트는 프론트엔드에서 주로 사용된 클라이언트 사이드 프로그래밍 언어였지만 node.js가 새롭게 등장하면서 지금은 백엔드와 프론트엔드 모두에서 사용되는 프로그래밍 언어가 되었습니다.

이전에 자바스크립트는 웹브라우저에서만 사용할 수 있는 프로그래밍 언어였으며 여러 다른 환경에서 사용할 수 있도록 다양한 시도가 있었으나 자바스크립트의 실행 속도 문제로 큰 호응을 얻지 못하다가 2008년 구글이 크롬의 V8 엔진을 오픈 소스로 공개하면서 자바스크립트는 새롭게 도약할 기회를 맞이하게 됩니다. V8 엔진은 자바스크립트 엔진과는 다르게 자바스크립트가 매우 빠르게 동작할 수 있는 실행 환경을 제공하였기 때문에 속도 문제가 해결되면서 2009년에 라이언 달(Ryan Dahl)에 의해서 크롬의 V8 엔진에서 실행되는 node.js가 세상에 공개되었습니다.

node.js는 크롬 V8 자바스크립트 엔진으로 빌드 된 자바스크립트 런타임입니다. 다시 말해 node.js는 자바스크립트로 작성된 프로그램을 실행할 수 있는 실행환경으로 node.js를 이용하면 자바스크립트를 웹브라우저가 아닌 다른 컴퓨터 환경에서 사용할 수 있게 됩니다.

### 1.2.7 Ajax(Asynchronous Javascript and XML)

서버와 통신하는 자바스크립트 프로그래밍 기법을 통틀어 Ajax라 부릅니다.

이 Ajax는 XMLHttpRequest 객체를 이용해 서버와 클라이언트가 통신하는 기법으로 동기식, 비동기식이 있으며 기본적으로 Ajax를 구현하는 것은 비동기통신 방식을 의미합니다.

XMLHttpRequest는 서버와 통신을 하거나 서버로부터 데이터를 가져와 이용하는 객체를 통틀어 XMLHttpRequest라고 부릅니다. 예전에 Ajax 프로그래밍을 구현할 때 순수 자바스크립트만을 사용하였지만 오늘날에는 자바스크립트를 보다 편리하게 사용할 수 있도록 지원하는 jQuery라는 라이브러리를 사용해 보다 간편하게 Ajax를 구현할 수 있습니다.

### 1.2.8 웹앱, 네이티브앱, 하이브리드앱

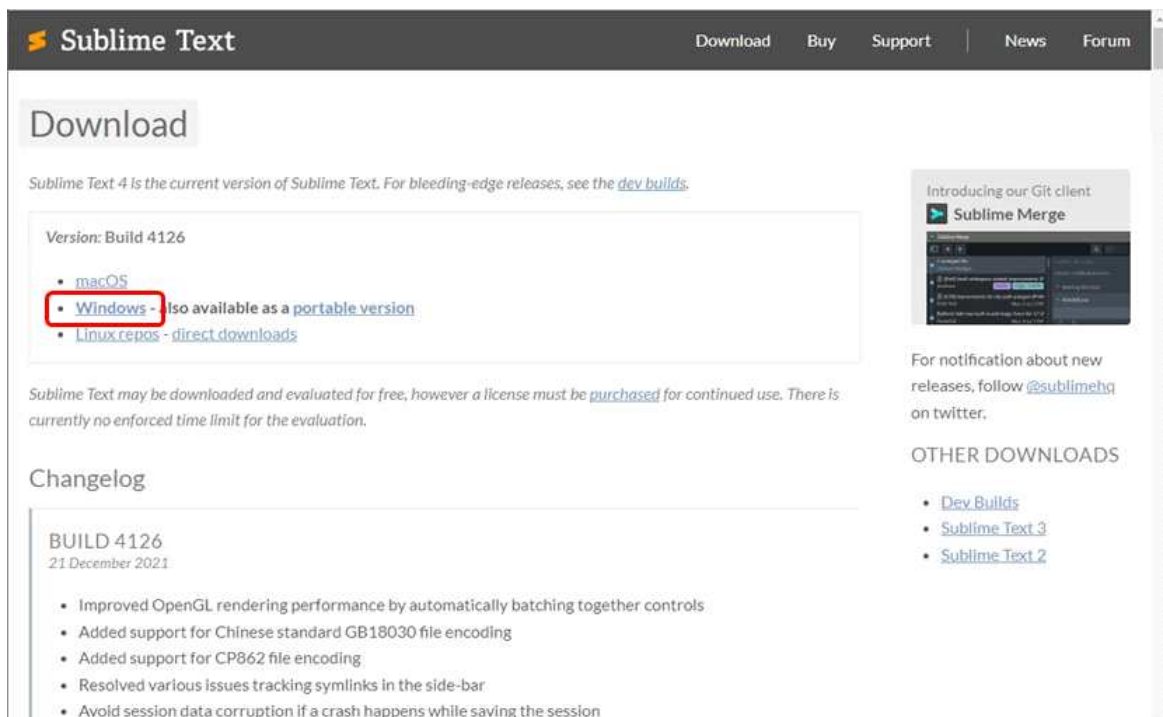
웹 앱은 브라우저를 통해 접속하는 모바일 웹 어플리케이션을 일컫는 말이며, 네이티브 앱은 스마트폰 OS환경에서 제공하는 프로그래밍 언어와 SDK를 이용해 제작된 앱으로 iOS는 Object-C, 안드로이드는 Java 등의 프로그래밍 언어로 제작된 앱을 말합니다. 하이브리드 앱은 웹앱 + 네이티브 앱의 결합된 형태로 먼저 웹앱으로 개발한 후 오픈소스 크로스 프레임워크를 이용하여 네이티브 앱 형태로 변환된 앱을 말합니다. 대표적인 크로스 플랫폼 프레임워크로는 전 세계적으로 유명한 폰갭(PhoneGap)과 타이타늄(Titanium) 등이 있으며 이외에도 요즘 많이 뜨고 있는 하이브리드앱을 개발할 수 있는 프레임워크에는 React Native, IONIC, Framework7, Onsen UI 등이 있습니다.

## 1.2 웹프로그래밍 도구 설치

프로그래밍에서 코드를 편리하게 작성하고 관리할 수 있도록 도와주는 프로그램을 통합개발환경 (IDE, Integrated Development Environment)이라고 부릅니다. 웹프로그래밍을 위한 IDE에는 이클립스(Eclipse), 비주얼스튜디오코드(VisualStudioCode) 등등이 있습니다. 우리는 이 IDE 보다 사용법이 단순하고 가벼운 텍스트 편집기를 사용할 것입니다. 무료로 사용할 수 있는 텍스트 편집기도 매우 다양하게 나와 있지만 서브라임텍스트(SublimeText)라는 텍스트 편집기를 사용할 것입니다.

먼저 서브라임텍스트를 다운로드 받을 수 있는 아래 사이트로 접속해 본인이 사용하는 운영체제를 선택해 프로그램을 다운로드 받습니다. 여기서는 국내에서 가장 많이 사용하고 있는 PC 운영체제인 Windows10을 기준으로 설명하겠습니다.

<https://www.sublimetext.com/download>



**Sublime Text** Download Buy Support News Forum

### Download

Sublime Text 4 is the current version of Sublime Text. For bleeding-edge releases, see the [dev builds](#).

Version: Build 4126

- [macOS](#)
- **Windows** - also available as a [portable version](#)
- [Linux repos](#) - [direct downloads](#)

Sublime Text may be downloaded and evaluated for free, however a license must be [purchased](#) for continued use. There is currently no enforced time limit for the evaluation.

### Changelog

**BUILD 4126**  
21 December 2021

- Improved OpenGL rendering performance by automatically batching together controls
- Added support for Chinese standard GB18030 file encoding
- Added support for CP862 file encoding
- Resolved various issues tracking symlinks in the side-bar
- Avoid session data corruption if a crash happens while saving the session

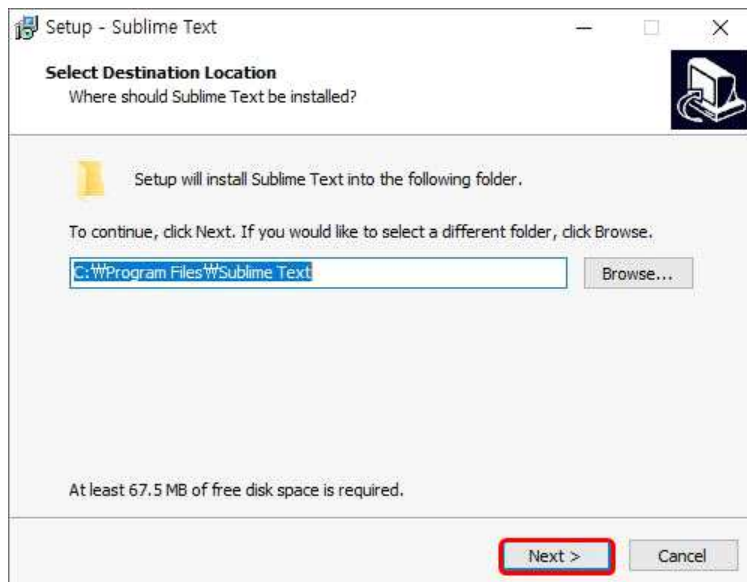
Introducing our Git client  
**Sublime Merge**

For notification about new releases, follow [@sublimehq](#) on twitter.

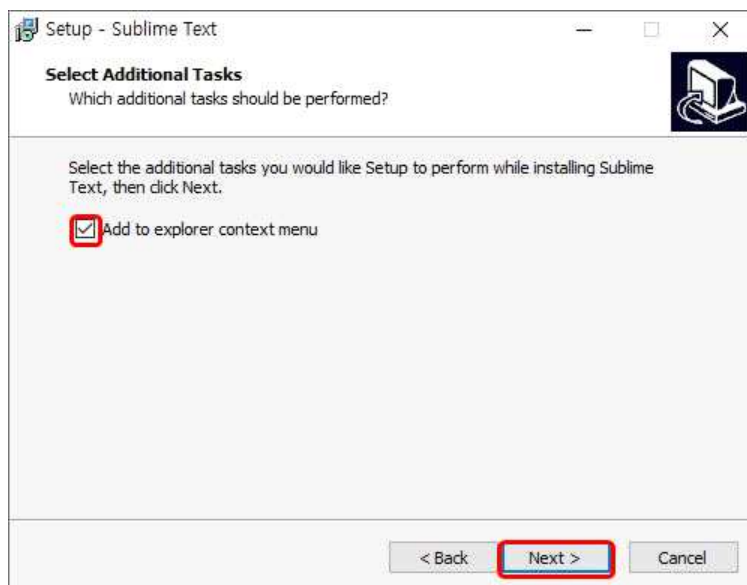
### OTHER DOWNLOADS

- [Dev Builds](#)
- [Sublime Text 3](#)
- [Sublime Text 2](#)

다운로드 받은 프로그램을 더블클릭해 실행하면 다음 그림과 같이 설치프로그램이 실행됩니다.  
프로그램 설치 위치는 기본 그대로 두고 “Next>” 버튼을 클릭해 다음 단계로 넘어갑니다.

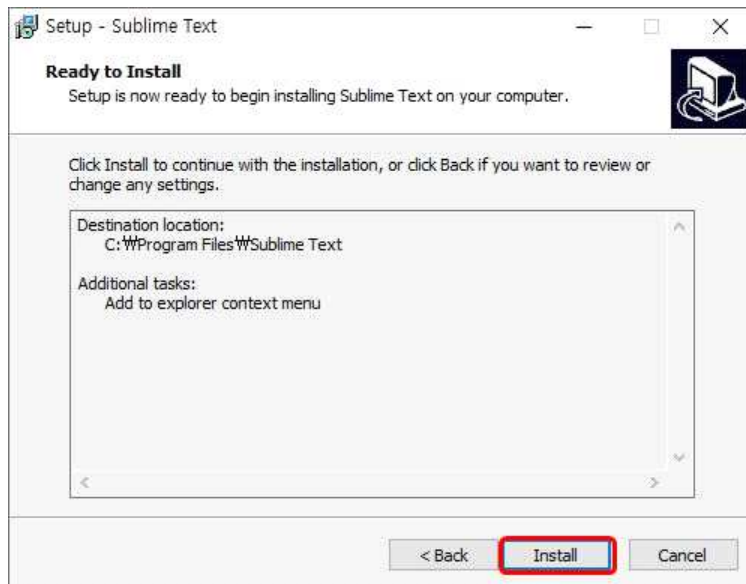


다음 단계에서 “Add to explorer context menu”의 체크박스를 체크하고 “Next>” 버튼을 클릭해 다음 단계로 이동합니다.



다음 그림과 같이 프로그램 설치 위치와 옵션 선택에 대한 안내 화면이 나타나는데 여기서 “Install” 버튼을 클릭하면 프로그램 설치가 진행됩니다.





프로그램 설치가 완료되면 다음 그림과 같은 화면이 나타나는데 여기서 “Finish” 버튼을 클릭해 설치를 완료 합니다.



## 1.3 HTML 문서의 기본 구조

### ▶ 프로젝트명 : HtmlStudyCh01

HTML 문서의 기본 구조는 아래와 같이 <html> 태그로 시작해 </html>로 종료해야 합니다. 또한 <html></html> 태그 내에 두 개의 또 다른 태그가 존재하는데 그 태그는 바로 <head> 태그와 <body> 태그입니다. 대부분의 태그는 여는 태그와 닫는 태그 쌍으로 구성되어 있습니다.

```
<!DOCTYPE html>
<html>
  <head>
    문서를 정의하는 데이터(HTML 문서에 대한 정보로 구성)
  </head>
  <body>
    문서에 표시되는 콘텐츠(화면에 보여 지는 내용으로 구성)
  </body>
</html>
```

그림 1-1 HTML 문서의 기본구조

### 1.3.1 DTD(Document Type Definition)

DTD도 여러 가지 버전이 있는데 버전에 따라서 브라우저의 해석 방법이 달라집니다. DTD는 브라우저에게 HTML 문서가 어떤 표준에 따라 작성되었는지 알려주는 역할을 합니다.

#### - HTML 4.01 - 예전에는 이렇게 썼다고 참고만 합시다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

#### - HTML5

HTML5에서는 아래와 같이 문서 타입을 선언하는 부분에 별도의 DTD를 지정하지 않습니다. 아래와 같이 기술하면 현재 HTML 문서가 HTML5의 문서임을 명시하는 것입니다.

```
<!DOCTYPE html>
```

### 1.3.2 <head> 관련 태그

<head> 태그는 문서의 머리 부분으로 웹 문서에 대한 정보를 기술하는 곳으로 웹브라우저 상에는 내용이 표시되지 않습니다. <head> 태그 내에는 현재 HTML 문서에 대한 메타데이터(metadata)를 정의하는 여러 태그 들이 사용됩니다. 메타데이터란 데이터를 설명하는 데이터를 의미하며 현재 HTML 문서에 대한 설명 정보들을 정의할 수 있는 태그가 사용됩니다. 예를 들면 문서의 제목을 정의하거나 문서의 스타일, 문서에 사용되는 스크립트 등을 정의할 수 있으며 문서의 설명이나 키워드, 작성자 등을 정의하는 다음과 같은 태그를 사용합니다.

## - <title> 태그

HTML 문서의 제목을 정의하며 HTML 문서의 <head> 태그 내에 기술합니다. <title> 태그의 내용은 웹브라우저의 윈도우 창의 제목(요즘은 탭의 제목)으로 표시되며 즐겨찾기(favorites)에 추가할 때 기본 지정되는 제목으로도 사용됩니다.

## - <meta> 태그

메타데이터란 데이터에 대한 정보를 말하는데 <meta> 태그는 HTML 문서에 대한 메타 데이터 즉 현재 HTML 문서에 대한 여러 가지 정보를 제공하기 위해 사용되는 태그입니다.

이 태그는 브라우저 화면에는 표시되지 않지만 컴퓨터에 의해 해석되는 정보들로 구성되며 일반적으로 HTML 문서의 설명, 키워드, 문서의 작성자 등 HTML 문서의 데이터를 지정하고 정의하는 데 주로 사용됩니다.

## ▶ HTML 문서의 구조

### - html\_doc01.html

```
<!-- HTML 문서 첫 출에 현재 문서가 HTML5의 문서임을 기술합니다. -->
<!DOCTYPE html>
<html lang="ko">
  <head>
    <!-- 아래의 태그들은 필요할 때 사용할 수 있으며 모두 생략할 수 있습니다. -->
    <title>HTML 문서의 구조</title>
    <!-- 웹 페이지를 제작할 때 어떤 톨을 사용했는지 정의 -->
    <meta name="Generator" content="eclipse" >
    <!-- 웹 페이지를 제작자를 정의 -->
    <meta name="Author" content="jini">
    <!-- 검색엔진에서 검색할 웹 페이지의 키워드를 정의 -->
    <meta name="Keywords" content="HTML&CSS">
    <!-- 웹 페이지의 주제를 정의 -->
    <meta name="Description" content="HTML&CSS를 이용한 웹 페이지 만들기">
    <!-- 웹 페이지를 작성할 때 사용한 인코딩 문자 셋을 정의 -->
    <meta charset="UTF-8">
    <!-- 설정한 시간이 지나면 자동으로 지정한 url로 이동 합니다. -->
    <meta http-equiv="refresh" content="3; url=meta01.html">
    <style>
      /* 문서에 지정하는 스타일을 정의 */
    </style>
    <link href="외부의 스타일 파일을 링크.css"/>
    <script>
      // 문서에서 사용되는 자바스크립트를 정의
    </script>
  </head>
  <body>
```

```
<!-- body 태그 안에 기술된 내용은 웹브라우저 화면에 출력됩니다. -->
<h1>HTML 문서의 구조</h1>
<p>3초 후에 다른 페이지로 이동 합니다.</p>
</body>
</html>
```

#### - meta01.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>HTML 문서 구조 - 두 번째 페이지</title>
</head>
<body>
    <h1>HTML 문서 구조 - 두 번째 페이지</h1>
    <p>메타 태그를 이용해 이동한 페이지 입니다.</p>
</body>
</html>
```

※ 요소(Element), 태그(Tag), 속성(Attribute), 속성 값(Value)



▶ 요소(Elements)

html에서 ‘<’와 ‘>’로 둘러싸인 시작 태그와 ‘</’와 ‘>’로 둘러싸인 종료 태그로 이루어진 태그의 조합을 의미합니다. 일반적으로 요소와 태그의 개념이 뚜렷하게 구별 되지 않고 혼용 되는 경우가 많지만 CSS나 자바스크립트에서 요소라는 의미는 매우 중요합니다. 요소는 “<h1>문서의 제목</h1>” 전체를 의미 하지만 태그는 <h1>과 </h1> 각각을 의미합니다.

▶ 태그(Tag)

태그는 요소의 일부로 ‘<’와 ‘>’로 둘러싸인 시작 태그와 ‘</’와 ‘>’로 둘러싸인 종료 태그 두 종류가 있으며 시작 태그와 종료 태그 사이에는 “<h1>문서의 제목</h1>”와 같이 태그의 시작 태그와 종료 태그 사이에 내용이 기술되며 시작 태그는 요소의 시작을 종료 태그는 요소의 끝을 의미합니다. 또한 img, hr, br 등과 같이 시작 태그는 있지만 태그의 내용이 없어서 종료 태그가 필요 없는 태그가 있으며 이러한 태그는 내용이 없기 때문에 비어있는 태그(Empty Tag)라고 하며 하나의 태그에 ‘<’와 ‘/>’를 사용해 요소의 시작과 끝을 동시에 표현할 수 있습니다. 태그 하나에 ‘<’와 ‘/>’를 같이 사용하므로 셀프 클로징 태그(Self Closing Tag)라고도 부릅니다. 셀프 클로징 태그는 XML 또는 XHTML에서 사용하던 문법으로 HTML5에서는 선택적(Optional)으로 사용할 수 있으며 필수사항은 아닙니다.

▶ 속성(Attributes)

속성은 요소의 시작 태그 안에서 사용 되는 것으로 요소의 특징과 추가적인 정보를 정의 하는데 사용되며 속성의 이름은 소문자로 작성하도록 W3C에서 권장하고 있습니다. “<td colspan=“2”>셀의 내용</td>”에서 <td> 태그 안에 colspan이라는 속성이 추가되면서 가로로 두 개의 셀을 합친 상태로 화면에 표시됩니다.

▶ 속성 값(Value) 또는 인수(Arguments)

속성에 지정한 값을 속성 값(Value) 또는 인수(Arguments)라고 하며 위에서 colspan 속성에 2라는 속성의 값을 지정해서 가로로 두 개의 셀을 합쳐서 화면에 표현되도록 한 것입니다. 속성 값을 지정할 때는 큰 따옴표(“) 또는 작은따옴표(‘)로 감싸서 지정하면 됩니다.

## 2. HTML 문서작성 기본기 익히기

### ▶ 프로젝트명 : HtmlStudyCh02

이번 챕터에서는 앞으로 여러분이 HTML 문서를 작성하는데 있어 기본적으로 꼭 알아야할 몇 가지 태그에 대해 학습해 봅시다.

### 2.1 HTML 문서 구조와 주석

일반적으로 프로그램을 작성할 때 나중에 참조하거나 수정할 것을 고려하여 중요한 부분에 주석을 기술하는 것이 일반적입니다. HTML에서도 문서를 작성할 때 특별히 참고 해야 할 부분에 주석을 기술해 놓으면 주석 태그로 둘러싼 부분은 웹브라우저가 설명으로 간주하여 화면에 출력하지 않습니다. 주석 태그는 “<!--” 시작하여 “-->”로 끝나야 하며 시작 태그 문자인 “<!--” 는 문자사이 공백이 있으면 주석 처리가 되지 않습니다.

### ▶ HTML 문서 구조와 주석

#### - webstudy02\_01.html

```
<!-- HTML 문서 첫 줄에 현재 문서가 HTML5의 문서임을 기술합니다. -->
<!DOCTYPE html>
<html>
<head><!-- 여기는 HTML 주석입니다. -->
    <meta charset="UTF-8">
    <style type="text/css"></style>
    <script type="text/javascript"></script>
    <title>HTML 문서 구조와 주석</title>
</head>
<body>
    <h1>Hello HTML</h1>
    <p>주석 태그 안의 내용은 웹 브라우저에서 보이지 않습니다.</p>
    <!-- <p>반갑습니다. HTML</p>
    <a href="http://www.naver.com">네이버 가기</a> -->
</body>
</html>
```

### 2.2 문단과 줄 바꿈

HTML 문서는 워드프로세서처럼 Enter 키를 입력한다고 해서 줄 바꿈이 되지 않고 하나의 공백으로 인식하여 화면에 표시됩니다. HTML 문서에서 특별히 줄 바꿈을 하고자 할 경우에는 <br/> 태그(break line)를 사용하여 줄 바꿈을 처리해야 합니다. 또한 특정 내용을 하나의 문단으로 구분하여 표현하기 위해서는 시작 태그인 <p> 태그(paragraph)와 종료 태그인 </p> 태그를 사용하여 문단의 내용을 감싸서 하나의 문단으로 표현해야 합니다. 줄 바꿈 기능을 가진 <br/> 태그는 태그 안

에 내용이 없는 비어 있는 태그(empty tag)로 속성도 가지고 있지 않는 태그이며 인라인 레벨 요소에 속합니다. 비어 있는 태그는 내용이 없으므로 여는 태그와 닫는 태그를 하나의 태그에서 사용하는 셀프 클로징 태그를 선택적으로 사용할 수도 있지만 하나의 문단을 구성하는 <p></p> 태그는 속성과 내용을 가질 수 있으며 블록 레벨 요소에 속합니다.

## ▶ 문단과 줄 바꿈

- webstudy02\_02.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>문단과 줄 바꿈</title>
</head>
<body>
  <h2>애국가</h2>
  <p>
    동해물과 백두산이 마르고 닳도록<br/>
    하느님이 보우하사 우리나라 만세<br/>
    무궁화 삼천리 화려강산<br/>
    대한 사람 대한으로 길이 보전하세
  </p>
  <!--
    HTML 문서에서 일부 영역과 다른 영역을 구분하여 내용상의 구분을 표현할 때
    가로 수평선을 표현하는 <hr> 태그(horizontal rule)를 사용하여 구분합니다.
  -->
  <hr/>
  <p>
    남산위에 저 소나무 철갑을 두른 듯<br/>
    바람소리 불변함을 우리 기상일세<br/>
    무궁화 삼천리 화려강산<br/>
    대한 사람 대한으로 길이 보전하세
  </p>
</body>
</html>
```

## 2.3 HTML 문서에 제목 넣기

HTML에서 글자 크기를 조절하는 태그는 크게 두 가지로 나눌 수 있는데 그 중 손쉽게 사용할 수 있는 것이 Heading 태그인 Hn 태그입니다. <h1> ~ <h6>까지 총 6단계로 글자 크기를 표현할 수 있습니다. Hn 태그에서 n 부분에 지정하는 숫자가 커질 수록 글자 크기는 작아집니다. 이전 HTML4에서 <font> 태그를 이용해 size 속성을 지정하여 글자 크기를 조절 할 수 있었지만 아쉽게도 <font> 태그는 HTML5에서 더 이상 사용되지 않으며 글자 크기는 style을 적용하여 font-size

값을 지정하면 글자 크기를 조절할 수 있습니다.

## ▶ Heading 태그와 style 속성

### - webstudy02\_03

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Heading 태그와 style 속성</title>
</head>
<body>
  <!--
    아래는 태그의 style 속성을 사용해 인라인 방식으로 스타일을 적용했습니다.
    CSS(Cascading Style Sheet)를 사용해 글자를 중앙 정렬을 할 수 있습니다.
  -->
  <!--
    Heading 태그는 기본적으로 텍스트를 굵게 표시하며 h1에서 h6까지 6단계로
    글자 크기를 표현합니다. 6단계 중 h1이 글자 크기가 가장 크며 h6은 가장 작다.
  -->
  <h1 style="text-align: center;">h1의 제목</h1>
  <h2 style="text-align: right;">h2의 제목</h2>
  <h3 style="text-align: left;">h3의 제목</h3>
  <h4>h4의 제목</h4>
  <h5>h5의 제목</h5>
  <h6>h6의 제목</h6>
  <!--
    style 속성을 이용해 폰트, 글자 크기, 글자의 색상 등을 지정하고 있습니다.
    여러 CSS 스타일을 적용하려면 세미콜론(;)으로 각 스타일을 구분해 지정합니다.
  -->
  <p style="text-align: center">
    <span style="font-size: 10px; font-family: '궁서체';">
      font-size 10픽셀로 지정</span><br/>
    <span style="font-size: 12px; font-family: gulim;">
      font-size 12픽셀로 지정</span><br/>
    <span style="color: green;">글자 크기의 기본은 16픽셀 입니다.</span><br/><br/>
  </p><br/><br/>
  <p style="text-align: center; background-color: black; color: white"
    title="Java, JSP 등의 통합개발 환경을 제공하는 툴">
    Eclipse<br/>
  </p>
</body>
</html>
```



## 2.4 다양한 서식 지정하기

HTML 문서를 작성할 때 단순히 글자의 크기를 키우거나 줄이는 것만이 아닌 글자에 밑줄 효과를 표현 할 수도 있고 글자를 강조하기 위해 기울임 효과와 더불어 굵게 표현할 수도 있습니다. 이와 같이 HTML은 텍스트에 다양한 효과를 지정해 표현할 수 있도록 여러 가지 태그를 제공하고 있습니다.

표 2-1 글자를 다양하게 표현하는 서식 태그

태 그	설 명	태 그	설 명
b	글자를 굵게 표시	strong	글자를 굵게 표시(강한강조)
i	글자를 이탤릭체로 표시	em	이탤릭체로 표시(주관적 강조)
mark	텍스트에 형광펜 효과	sub	글자를 아래첨자로 표시
small	주변 글자보다 작게 표시	sup	글자를 위첨자로 표시
ins	글자에 밑줄 표시	del	글자에 취소선 표시

### ▶ 글자를 다양하게 표현하는 서식 태그

- webstudy02\_04.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>글자를 다양하게 표현하는 서식 태그</title>
</head>
<body>
  <h2>다양한 글자관련 서식 태그</h2>
  <!--
    b 태그는 화면에서 텍스트를 단순히 굵게 표현할 때 사용하며
    strong 태그는 텍스트를 굵게 표현할 뿐만 아니라 중요하다는 의미를 가집니다.
  -->
  <p><b>b 태그(bold text)는 상품명 또는 단순 볼드체 표현에 사용</b></p>
  <p><strong>strong 태그는 중요한 내용의 강한 강조에 사용</strong></p>
  <!--
    i 태그(italic text)는 글씨를 단순히 기울임체로 표현할 때 사용하며
    em 태그(emphasized text)는 기울임체로 표현할 뿐만 아니라 중요함의 의미를 가집니다.
  -->
  <p><i>i 태그는 속어의 인용이나 생각, 꿈, 기타 보통 이탤릭체 표현에 사용</i></p>
  <p><em>em 태그는 주관적 강조를 나타낼 뿐 중요성은 나타내지 않습니다.</em>
  <p>주변 글자보다<small> small 작게</small> 표시할 때 사용</p>
  <p>글자를<sub>sub 아래첨자</sub> 로 표시</p>
```

```

<p>글자를<sup>sup 위첨자</sup> 로 표시</p>

<h2><mark>글자관련 서식태그 활용</mark></h2>
<p><b>HTML5</b>는 새로운 웹 표준으로 <ins>2014년 10월 28일</ins> 정식 권고안이 발표
되었습니다.</p>
<p><strong>웹 표준을 지켜 HTML 문서를 작성하는 것은 중요합니다.</strong></p>
<p><em>나는 그가 잘못했다고 생각합니다.</em></p>
<p>이 상품의 판매가는 3,900원<del>(5,000원)</del>입니다.<br/>
    <span style="text-decoration:line-through;">(5,000원)</span>
</p>
<p>대한민국<sup>(大韓民國)</sup>은 5000년 역사를 가진 나라입니다.</p>
<p>화학기호 H<sub>2</sub>O는 물을 의미합니다.</p>
</body>
</html>

```

## 2.5 특수문자 사용하기

HTML 문서를 작성하다 보면 키보드에서 입력할 수 없는 특수 문자나 메타 문자를 입력해야 하는 경우가 종종 있습니다. 메타 문자란 특정 의미를 지닌 문자를 말하며 HTML에서는 &나 태그의 시작과 끝을 알리는 <, > 문자 등이 여기에 속합니다. &, <, > 문자는 HTML에서는 예약어(reserved characters)로 웹브라우저에서는 메타문자의 시작 또는 태그의 시작과 끝으로 인식되기 때문에 문자 그대로 입력해서는 안 됩니다. 그러므로 이런 종류의 예약어를 표현하기 위해서는 미리 예약된 코드를 사용하며 이렇게 별도로 만든 문자 셋을 엔티티(entity)라고 부릅니다. HTML에서 예약어를 사용하기 위한 표현식과 많이 사용하는 엔티티는 다음과 같습니다.

&표현식:("&"는 메타문자 표현식의 시작을 의미하며 ";"는 끝을 의미합니다.)

표 2-2 HTML에서 사용하는 특수문자와 표현식

특수문자	표현식	특수문자	표현식
<	&lt;	&	&amp;
>	&gt;	"	&quot;
space	&nbsp;	`	&acute;
가운데 점	&middot;	©	&copy;
-	\$shy;	®	&reg;
±	&#177;	÷	&divide;

### ▶ HTML에서 특수문자 사용하기

- webstudy02\_05.html

```

<!DOCTYPE html>
<html>
<head>

```

```

<meta charset="UTF-8">
<title>HTML에서 특수문자 사용하기</title>
</head>
<body>
  <p>HTML에서 태그의 시작은 &quot;&lt;&gt;&quot;로 시작하여
  &quot;&lt;/&gt;&quot;로 끝나야 합니다.</p>
  <p>HTML에서는 &amp;, &copy;, &reg; 등도 특수문자로 인식되며</p>
  <p>다음과 같이 아스키코드를 이용해 태그의 시작인 &#34;&#60;&#62;&#34;와
  &#34;&#60;/&#62;&#34;를 출력할 수도 있습니다.</p>
  <p>또한 HTML에서 "Enter" 키를 여러 번 누르거나 스페이스 바를 여러 번 눌러<br/>
  문서 내에 줄 바꿈과 공백을 여러 번 입력했다 하더라도 단 하나의 공백으로<br/>
  표현되기 때문에 "Enter"는 줄 바꿈 태그인 &quot;&lt;br/&gt;&quot;를 이용해<br/>
  처리하며 공백은 &amp;nbsp; 표현식을 이용하거나 &amp;#32; 아스키코드를<br/>
  이용해 처리해야 합니다.<br/>
  아스키코드에 사용된 숫자는 10진수가 아니라 16진수 표기법을 사용합니다.</p>
</body>
</html>

```

## 2.6 입력한 글을 그대로 보여주기

앞에서 살펴보았던 줄 바꿈 태그나 공백을 입력하는 표현식 등을 이용해 원하는 문서 형태를 표현할 수 있지만 그래도 어딘가 모르게 불편을 느낄 때가 많다. 줄 바꿈이야 <br/> 태그를 사용하여 쉽게 줄 바꿈 할 수 있지만 공백을 입력하는 표현식인 &nbsp;를 사용하여 공백을 입력하기란 여간 귀찮은 작업이 아닙니다. 또한 공백 문자를 적절히 입력한 후 문서를 실행해 보면 글자의 간격이 서로 맞지 않는 경우가 종종 발생하곤 합니다. 그래서 편집기에서 입력한 형태 그대로 웹브라우저에 표시 된다면 쉽고 편리하게 HTML 문서를 편집 할 수 있을 것입니다. 이 때 사용할 수 있는 태그가 바로 <pre> 태그(preformatted text)입니다.

### pre 태그를 이용한 HTML 문서작성

- webstudy02\_06.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>pre 태그를 이용한 HTML 문서작성</title>
</head>
<body>
  <h2>웹 표준 인기도서</h2><pre>

```

---

순위	도서명	출판사
----	-----	-----

---

1.	모던 웹 디자인을 위한 HTML5+CSS3 입문	한빛미디어
----	----------------------------	-------

2. 만들면서 배우는 HTML5+CSS3+jQuery      한빛미디어
  3. Do it! 쉽게 배우는 웹앱&하이브리드앱      이지스퍼블링
- 

```
</pre>  
</body>  
</html>
```

### 3. 하이퍼링크로 문서 연결하기

#### ▶ 프로젝트명 : HtmlStudyCh03

하이퍼링크(Hyper Link)는 HTML 문서의 가장 중요하고 근본이 되는 개념으로 다른 HTML문서나 다른 사이트 혹은 같은 문서의 특정 위치로 이동하기 위한 연결입니다. 하이퍼링크는 단순 텍스트뿐만 아니라 이미지나 멀티미디어 요소에도 링크를 지정할 수 있습니다. 이렇게 하이퍼링크를 통해 하이퍼텍스트 문서를 오 갈 수 있도록 하이퍼링크를 해주는 태그가 바로 <a> 태그입니다. 그럼 하이퍼링크에 대해 자세히 알아보도록 하자

#### 3.1 하이퍼링크 연결과 이동하기

HTML 문서에 하이퍼링크를 설정하는 방법은 <a> 태그의 href 속성에 이동할 문서의 URL(Uniform Resource Locator)을 아래와 같은 방법으로 지정해 주면 됩니다.

<a> 태그에 지정할 수 있는 속성은 아래 표를 참고하기 바랍니다.

<a href="연결할 문서 경로(URL)">링크를 설정할 글자나 이미지</a>

표 3-1 a 태그의 속성

속성명	설 명
href	하이퍼링크를 클릭했을 때 이동할 페이지 경로를 지정
target	하이퍼링크를 클릭했을 때 호출된 페이지가 보여 질 위치를 지정 연결된 문서를 현재 창에 보여줄지 또는 새로운 창에 보여줄지를 지정합니다. _self : 링크로 연결된 HTML 문서를 현재창이나 현재 프레임에 출력 _blank : 링크로 연결된 HTML 문서를 새로운 창이나 새 탭에 출력 _parent : 링크로 연결된 HTML 문서를 현재 창을 호출한 부모 창에 출력 _top : 링크로 연결된 HTML 문서를 최상위 프레임에 출력
title	하이퍼링크에 대한 설명

이제 anchor 태그를 이용해 하이퍼링크를 지정하는 방법과 그 속성도 알아보았으니 관련 예제를 만들어 보고 문서를 이동시켜 봅시다. 이번 예제는 링크를 통해 같은 도메인에 속한 폴더의 문서와 다른 도메인에 존재하는 특정 페이지로 이동하는 예제를 만들어 볼 것입니다. 그리고 이미지에 링크를 지정하는 방법도 알아 볼 것입니다.

#### ▶ 글자와 이미지에 하이퍼링크 연결하기

- webstudy03\_01.html

```
<!DOCTYPE html>  
<html>
```

```

<head>
  <meta charset="UTF-8">
  <title>글자와 이미지에 하이퍼링크 연결하기</title>
</head>
<body>
  <h2>글자와 이미지에 하이퍼링크 연결하기</h2>
  <!-- 현재 창에서 같은 도메인의 링크된 문서 열기 -->
  <p><a href="sub03_01.html">sub03_01.html로 이동하기</a></p>

  <!-- 현재 창에서 다른 도메인의 링크된 문서 열기 -->
  <p>
    <!--
      HTML 문서에 이미지를 삽입하려면 img 태그를 사용합니다.
      img 태그는 내용이 없는 빈 태그(empty tag)이며 아래와 같이 속성만 가지고 있습니다.
      alt 태그는 이미지가 문서에 표현되지 못할 경우 이미지 대신 나타낼 문자열을 지정하며
      width, height, border와 같은 속성은 주로 CSS를 더 많이 사용합니다.

      웹 문서에서는 주로 jpg, gif, png 포맷의 이미지를 많이 사용합니다.
    -->
    <a href="http://finance.naver.com/" target="_self"
      alt="네이버 금융가기" title="네이버 금융가기"/></a>
  </p>

  <!-- 새 창에서 같은 도메인(웹 사이트)의 링크된 문서 열기 -->
  <p><a href="sub03_01.html" target="_blank">sub03_01.html로 이동하기</a></p>

  <!-- 새 창에서 다른 도메인(웹 사이트)의 링크된 문서 열기 -->
  <p>
    <a href="http://cafe.daum.net/" target="_blank" title="다음 카페 가기">
      </a>
    </p>
  </body>
</html>

```

#### - sub03\_01.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>글자와 이미지에 하이퍼링크 연결하기</title>
</head>
<body>
  <h1>여기는 sub03_01.html 페이지</h1>

```

```
<b>현재 위치 : /HtmlStudyCh03/sub03_01.html</b>
<!-- 새로운 창에서 링크된 문서 열기 -->
<p><a href="webstudy03_01.html">메인으로</a>
</body>
</html>
```

## 4. 가지런히 정렬된 목록 만들기

HTML 문서를 작성하다 보면 관련된 몇 가지 항목을 가지런히 정렬된 목록으로 구성해야 할 때가 종종 있습니다. 목록은 목차 등을 구성할 때도 사용되고 웹 페이지의 메뉴를 구성할 때도 사용됩니다. 목록 태그는 여러 개의 항목을 일정한 간격으로 나열하기 때문에 깔끔하고 정돈된 느낌을 준다. 또한 목록 태그를 사용하면 항목 별로 일정하게 들여쓰기와 줄 바꿈을 알아서 해준다.

목록 태그는 특정 기호(사각형, 원 등, 마커라고 함)를 목록 앞에 표시하는 순서 없는 목록과 숫자나 알파벳 등을 목록 앞에 표시해 목록의 순서를 정할 수 있는 순서 있는 목록으로 나눌 수 있습니다. 참고로 목록의 각 항목 앞에 표시되는 숫자나 기호를 마커(marker)라고 부릅니다.

### 4.1 순서 있는 목록 구성하기

HTML 문서에서 숫자나 알파벳 문자를 이용해 순서 있는 목록을 만들려면 <ol> (Ordered List) 태그를 사용합니다. <ol> 태그는 <li> 태그를 사용하여 각 항목을 나열 할 수 있습니다. <li> 태그는 자동으로 줄 바꿈 하는 기능을 가지고 있습니다.

```
<ol>
  <li>목록으로 나타낼 문자열1</li>
  <li>목록으로 나타낼 문자열2</li>
  <li>목록으로 나타낼 문자열2</li>
</ol>
```

표 4-1 ol 태그의 style 속성

속 성	list-style-type="decimal"	
list-style-type	decimal	아라비아 숫자로 목록번호를 작성, 기본 값
	upper-alpha	알파벳 대문자로 목록번호를 작성
	lower-alpha	알파벳 소문자로 목록번호를 작성
	upper-roman	로마 숫자 대문자로 목록번호를 작성
	lower-roman	로마 숫자 소문자로 목록번호를 작성

#### ▶ 순서 있는 목록 구성하기

- webstudy04\_01.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>순서 있는 목록</title>
</head>
```



<body>

<h1>순서 있는 목록</h1>

<h3>목차</h3><hr width="400" align="left"/>

<!--

예전에는 목록 태그의 type 속성에 필요한 목록의 형식[숫자(1), 로마자(i, I) 등]을 지정했으나 HTML5에서는 스타일을 사용해 리스트 스타일을 지정합니다.  
아래에서 type="1" 대신에 style="list-style-type: decimal"를 지정해 봅시다.

목록에서 각각의 리스트 요소 앞에 표시되는 것을 마커라고 부르며 순서 있는 목록의 마커의 기본 값은 아라비아 숫자이며 마커의 모양은 style로 지정할 수 있으며 list-style 또는 list-style-type으로 다음과 같은 값을 지정할 수 있습니다.

decimal : 숫자(기본 값)

upper-alpha : 영문 대문자, lower-alpha : 영문 소문자

upper-roman : 로마 숫자 대문자, lower-roman : 로마 숫자 소문자

참고 사이트 : [https://www.w3schools.com/html/html\\_lists.asp](https://www.w3schools.com/html/html_lists.asp)

-->

<ol style="list-style: decimal">

<li>HTML5 개발 환경</li>

<li>개발 환경 구성하기

<ol style="list-style-type: upper-alpha">

<li>웹 브라우저 설치하기</li>

<li>HTML 편집기 설치하기</li>

</ol>

</li>

<li>HTML5와 자바스크립트

<ol style="list-style-type: decimal">

<li>HTML5 둘러보기</li>

<li>자바스크립트 둘러보기</li>

</ol>

</li>

</ol>

</body>

</html>

## 4.2 순서 없는 목록 구성하기

순서 있는 목록과는 다르게 특정 기호를 숫자나 알파벳 대신 사용하는 순서 없는 목록을 작성할 수 있습니다. 순서 없는 목록을 작성하려면 <ul> (Unordered List) 태그를 사용하면 됩니다. <ul> 태그도 <li> 태그를 사용해 각 항목을 나열 합니다. <ul> 태그를 사용하는 방법과 속성은 아래와 같습니다.

참고로 <ul> 태그의 type 속성은 HTML5에서 더 이상 사용되지 않으며 목록 앞에 순서 없는 목록

의 기호를 표시하려면 style 속성을 사용합니다.

```
<ul>
  <li>목록으로 나타낼 문자열1</li>
  <li>목록으로 나타낼 문자열2</li>
  <li>목록으로 나타낼 문자열2</li>
</ul>
```

표 4-2 ul 태그의 style 속성

속 성 값	list-style-type= "disc"	
list-style-type	disc	●를 사용해 목록을 작성, 기본 값
	circle	○ 를 사용해 목록을 작성
	square	■ 를 사용해 목록을 작성

## ▶ 순서 없는 목록 구성하기

- webstudy04\_02.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset= "UTF-8">
  <title>순서 없는 목록</title>
</head>
<body>
  <h2>순서 없는 목록</h2>
  <!--
    다음과 같이 순서 있는 목록 안에 순서 없는 목록을 사용하거나
    순서 없는 목록에 순서 있는 목록을 중첩해서 사용할 수 있습니다.
  -->
  <ol style="list-style-type: lower-roman">
    <li>자바스크립트
      <ul style="list-style-type: circle">
        <li>자바스크립트 완벽 가이드</li>
        <li>node.js 프로그래밍</li>
      </ul>
    </li>
    <li>Java
      <ul style="list-style-type: square">
        <li>자바의 정석</li>
        <li>뇌를 자극하는 Java</li>
      </ul>
    </li>
  </ol>
```

```

        </li>
    </ol>
</body>
</html>

```

## 4.3 용어정의 목록 구성하기

앞에서 살펴본 <ol> 태그와 <ul> 태그는 숫자나 기호를 사용해 목록을 나열하는 태그 들이었다. 이번에 살펴볼 <dl> 태그는 용어를 설명하거나 정의할 때 사용하는 태그로 HTML 문서에서 용어와 그 용어에 대한 정의를 나열 할 때 유용합니다. 용어의 제목을 표현하기 위해서는 Definition Term의 약어인 <dt>를 사용하며 <dt> 태그에 기술한 용어의 설명을 표현하기 위해 Definition Description의 약어인 <dd> 태그를 사용합니다. 기본적으로 <dt> 태그와 <dd> 태그는 줄 바꿈 기능을 가지고 있어 용어의 제목 아래에 용어 설명이 나타납니다. 아래는 <dl> 태그를 사용하여 용어 정의 목록을 구성하는 방법입니다.

```

<dl>
    <dt>용어의 제목</dt>
    <dd>용어를 설명하는 텍스트</dd>
</dl>

```

### ▶ 용어정의 목록 구성하기

- webstudy04\_03.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>용어정의 목록</title>
</head>
<body>
    <b>용어정의 목록</b>
    <dl>
        <dt>웹 표준이란?</dt>
        <dd>웹 표준은 월드 와이드 웹의 측면을 서술하고 정의하는 공식 표준이나<br/>
        다른 기술 규격을 가리키는 일반적인 용어입니다. 최근에 이 용어는 웹<br/>
        사이트를 작성하는 데 중요도가 높아지고 있으며 웹 디자인, 개발과 <br/>
        관계가 있습니다.</dd>
    </dl>
    <!-- 여러 용어정의 목록을 표현할 경우 -->
    <dl>
        <dt>CSS [cascading style sheets]</dt>
        <dd>기존의 HTML은 웹 문서를 다양하게 설계하고 수시로 변경하는데 많은<br/>

```

```

    제약이 따르는데, 이를 보완하기 위해 만들어진 것이 스타일 시트이고<br/>
    스타일 시트의 표준안이 바로 CSS 입니다. 간단히 스타일시트라고도 합니다.</dd>
<dt>Ajax</dt>
<dd>WEB2.0의 기반 기술 중 하나입니다.<br/>
    AJAX는 자체가 하나의 특정한 기술을 말하는 것이 아니며,<br/>
    함께 사용하는 기술의 묶음을 지칭하는 용어로 대화식 웹 애플리케이션의<br/>
    제작을 위해 사용됩니다. </dd>
</dl>
</body>
</html>

```

## 4.4 이미지를 사용해 목록 구성하기

순서 있는 목록이과 순서 없는 목록의 앞에 붙는 숫자나 기호 혹은 도형을 마커(Marker)라고 합니다. 이번에는 이 마커를 이미지를 사용해 목록을 보다 미려하게 구성하는 방법에 대해 소개 하겠습니다. 이미지를 사용해 목록을 구성하는 방법은 HTML 태그를 사용하는 것이 아니라 CSS의 list-style-type 또는 list-style-image 속성을 사용해 이미지를 사용한 목록을 구성할 수 있습니다.

### ▶ 이미지를 사용해 목록 구성하기

- webstudy04\_04.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>이미지를 사용한 목록 구성하기</title>
</head>
<body>
    <h2>이미지를 사용한 목록 구성하기</h2>
    <ol style="list-style-type: decimal;">
        <li>HTML5 추가기능
            <ul style="list-style-image: url('images/list_type01.jpg');">
                <li>멀티미디어</li>
                <li>장치접근</li>
            </ul>
        </li>
        <li>HTML5 태그 기본
            <ul style="list-style-image: url('images/list_type02.png');">
                <li>입력 폼 태그</li>
                <li>공간 분할 태그</li>
            </ul>
        </li>
    </ol>

```

```
</body>  
</html>
```

## 5. 웹 문서에 테이블을 넣어봅시다

테이블이란 평소 우리가 많이 사용하는 Excel 시트처럼 열과 행으로 구성된 표를 말합니다. 예전에는 웹 페이지의 전체 레이아웃을 구성하는데 테이블이 많이 사용되었으며 열과 행 형태의 게시판과 같은 페이지를 제작할 때 아직도 많이 사용되긴 하지만 오늘날 시멘틱 웹(Semantic Web)이 강조되고 HTML5에 이와 관련된 시멘틱 태그들이 새롭게 추가되면서 테이블을 통해 웹 페이지의 레이아웃을 구성하거나 의미 없는 테이블을 사용하지 않는 것이 일반적입니다. HTML 4.01에서 사용되던 테이블과 관련된 많은 속성들은 CSS에서 처리가 가능하므로 HTML5에서는 테이블 태그와 관련된 속성들은 대부분 사용되지 않으며 요즘은 CSS에서 처리되는 경우가 일반적입니다.

### ※ 시멘틱 웹(Semantic Web)이란?

시멘틱(Semantic)의 사전적 의미는 '의미의, 의미론적인'이란 뜻을 가지고 있습니다. 이 시멘틱이란 단어와 웹이 결합되어 새롭게 생긴 단어로 웹상의 정보를 사람뿐만 아니라 컴퓨터도 이해할 수 있도록 해주는 기술을 말합니다. 그러기 위해서는 컴퓨터가 이해할 수 있도록 특정 규칙에 맞는 의미 정보를 이용해 웹 문서가 제작되어야 합니다. 즉 콘텐츠의 역할과 기능에 맞는 요소를 선택해 마크업 해야 한다는 것입니다. 예를 들어 웹 문서의 특정 숫자가 상품의 가격에 대한 정보인지 혹은 사람의 나이에 대한 정보인지 또는 날짜 정보인지를 컴퓨터가 쉽게 이해하고 찾을 수 있도록 의미적인 요소를 추가하여 웹 문서를 제작하자는 것입니다.

### 5.1 테이블 기본기 다지기

테이블의 정의를 알아 봤으니 이제 테이블을 웹 페이지에 표현하는 방법을 알아보시다. 웹 페이지에 테이블을 표현하려면 <table> 태그 안에 <tr>(Table Row) 태그와 <td>(Table Data) 태그를 사용해 테이블의 열과 행을 아래와 같이 기술해 줘야합니다.

```
<table> <!-- 테이블의 시작 -->
  <tr> <!-- 첫 번째 행의 시작 -->
    <td>데이터1</td> <!-- 첫 번째 행의 첫 번째 열 -->
    <td>데이터2</td> <!-- 첫 번째 행의 두 번째 열 -->
    <td>데이터3</td> <!-- 첫 번째 행의 세 번째 열 -->
  </tr>
  <tr> <!-- 두 번째 행의 시작 -->
    <td>데이터4</td> <!-- 두 번째 행의 첫 번째 열 -->
    <td>데이터5</td> <!-- 두 번째 행의 두 번째 열 -->
    <td>데이터6</td> <!-- 두 번째 행의 세 번째 열 -->
  </tr>
</table>
```

## ▶ 웹 문서에 테이블 넣기

- webstudy05\_01.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>웹 문서에 테이블 넣기</title>
  <style>
    /*
      style 태그를 이용해 내부 스타일 방식으로 스타일 적용
      요소 선택자를 이용해 문서의 특정 요소를 선택하여 스타일을 지정
    */
    table { width: 800px; border: 1px solid black; }
    th { font-size: 20px; border: 1px solid black; }
    tr { height: 30px; border: 1px solid black; }
    td { text-align: center; border: 1px solid black; }
  </style>
</head>
<body>
  <h2>웹 문서에 테이블 넣기</h2>
  <table>
    <tr style="height: 50px; background-color: #EAEAEA">
      <!-- 테이블에서 행 또는 열의 제목 셀을 표현할 때는 th 태그를 사용합니다. -->
      <th>자바스크립트</th>
      <th>Java</th>
    </tr>
    <tr>
      <td>자바스크립트 마스터북</td>
      <td>자바의 정석</td>
    </tr>
    <tr>
      <td>자바스크립트 완벽 가이드</td>
      <td>자바 프로그래밍 바이블</td>
    </tr>
  </table>
</body>
</html>
```

## 5.2 테이블 활용하기

<table> 태그와 관련된 기본적인 내용을 알아보았습니다. 위에서 사용된 속성들은 이전에도 언급 했듯이 HTML5에서는 더 이상 사용되지 않습니다. 앞으로는 이전 버전의 HTML과 호환을 위해 참고만 합시다.

테이블과 관련하여 위의 기본적인 태그 외에 다음과 같이 <table> 태그 내에서 사용할 수 있는 태그들이 있습니다. 아래 태그들은 HTML5에서도 계속 사용되니 주의 깊게 살펴보도록 합시다.

표 5-1 table 태그 관련 속성

태 그	속 성	설 명
table	border	테이블의 테두리 두께를 지정
td	colspan	지정한 수치만큼 가로 셀(열) 합치기 지정
	rowspan	지정한 수치만큼 세로 셀(행) 합치기 지정

표 5-2 tabel 태그 안에서 사용할 수 있는 태그

태 그	설 명
caption	table 요소 안에서 테이블 전체의 제목을 나타냅니다. caption 요소 안의 내용은 기본적으로 가운데 정렬됩니다.
thead	테이블의 머리말, 열의 제목(헤더)으로 구성된 행의 집합을 나타냅니다. caption, colgroup 태그 뒤에, tbody, tfoot 태그 앞에 선언합니다. table 태그 안에서 한번만 사용할 수 있습니다.
th	테이블 태그의 제목(헤더) 셀을 나타냅니다. 기본적으로 글꼴이 굵게 표시되고 가운데 정렬됩니다.
tbody	테이블의 본문, thead 태그 뒤에 선언합니다.
tfoot	테이블의 꼬리말, tbody 태그 뒤에 선언합니다. 열의 요약을 나타내며 table 태그 안에서 한번만 사용할 수 있습니다.
colgroup	col 태그를 그룹으로 관리하는 역할을 합니다. 행이 아닌 열 단위로 스타일을 제어할 수 있습니다.
col	colgroup 태그 안에서만 사용 할 수 있습니다. 테이블의 컬럼 별로 속성을 제어하거나 또는 여러 컬럼을 묶어 속성을 제어할 수 있도록 제공합니다.

## ▶ 과일 구입 현황 테이블로 만들기

### - webstudy05\_02.html

다음 예제는 <caption> 태그를 사용하여 테이블의 제목을 지정하고 <td> 태그의 rowspan 속성을 적용해 테이블의 마지막 열의 행들을 세로로 합쳐서 하나의 비고 셀로 표시될 수 있도록 하였으며 <td> 태그의 colspan 속성을 적용해 테이블의 마지막 행의 열들을 가로로 합쳐서 하나의 합계 셀로 표시될 수 있도록 하였습니다.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>과일 구입 현황 만들기</title>
```



```

<style>
  table { border-collapse: collapse; }
  caption {
    padding: 50px 0 15px 0;
    font-size: 24px;
    font-weight: bold;
  }
  th, td {
    padding: 10px 10px;
    border: 1px solid black;
  }
  #th_line {
    background-color: #F2F2F2;
  }
  .width150 { width: 150px; }
  .width100 { width: 100px; }
</style>
</head>
<body>
  <table>
    <caption>5월 과일 구입 현황</caption>
    <tr id="th_line">
      <th class="width150">과일명</th>
      <th class="width100">수량</th>
      <th class="width100">단가</th>
      <th class="width150">가격</th>
      <th class="width150">비고</th>
    </tr>
    <tr>
      <td>사과</td>
      <td>10</td>
      <td>3,000원</td>
      <td>30,000원</td>
      <td rowspan="3">바나나가<br/> 조금 비싸다</td>
    </tr>
    <tr>
      <td>배</td>
      <td>5</td>
      <td>4,000원</td>
      <td>20,000원</td>
    </tr>
    <tr>
      <td>바나나</td>
      <td>20</td>
      <td>5,000원</td>
    </tr>
  </table>

```

```

        <td>50,000원</td>
    </tr>
    <tr>
        <td>합계</td>
        <td colspan="4">165,000원</td>
    </tr>
</table>
</body>
</html>

```

### ▶ 연습문제 5-1 테이블을 이용한 택배 요금표 만들기

다음 그림은 모 택배 회사의 택배 요금표를 나타낸 것입니다. 이 그림을 참고해서 HTML에서 택배 요금표를 테이블로 작성해 봅시다.

**택배 요금표**

구 분	중량/규격 (가로, 세로, 높이의 합계)	가 격	
		회 원	비회원
CJ 대한통운	20kg 이하/ 160cm 이하	기본 3,700원 (회원 할인 : 300원 단, 1일 1회 한함)	최저 4,000원
한진택배	20kg 이하/ 140cm 이하	기본 4,000원	
한재로지스틱스	20kg 이하/ 140cm 이하	기본 3,700원 (회원 할인 : 300원 단, 1일 1회 한함)	최저 4,000원
※ 군부대 및 사서함을 이용한 주소는 집하 및 배송이 되지 않습니다. ※ 착불로는 택배예약이 되지 않습니다. ※ 배송사원이 택배 물품 수령 방문 시, 상품 종류에 따라 적절하게 물품이 포장 되어 있어야 접수가 가능합니다. ※ 반품의 경우 접수 불가합니다. ※ 회원할인은 1일 1회에 한합니다. (일부 택배사 제외)			

- 1) HTML 문서의 파일 명은 임의로 지정해 작성하시오.
- 2) 맨 아래 셀의 내용은 <ul> 태그를 사용하여 순서 없는 목록으로 구성하고 스타일시트를 적용하여 목록의 마커로 이미지를 지정하시오.
- 3) 테이블을 작성하고 테이블의 배경색, 테두리 속성 등의 디자인 적인 요소를 내부 스타일과 인라인 스타일 방식을 활용해 테이블에 스타일을 지정해 봅시다.

## 6. 인터랙티브 웹의 시작! 폼 만들기

지금까지 우리는 웹 사이트에 접속한 사용자에게 일방적인 정보만 제공하는 HTML 문서를 작성하였습니다. 하지만 오늘 날의 웹은 사용자와의 상호작용을 매우 중요시 합니다. 예를 들면 포털 사이트에 접속해 메일을 확인하거나 블로그를 편집하거나 또는 카페의 정보를 이용하기 위해서는 로그인을 해야 합니다. 또한 로그인이 필요한 서비스를 제공받으려면 서비스를 제공하는 사이트에 먼저 회원으로 가입해야 합니다. 로그인 창을 한번 생각해 봅시다. 로그인 창은 사용자의 아이디와 비밀번호를 입력받는 텍스트 입력상자와 실제 로그인을 요청하는 버튼 등으로 구성되어 있습니다. 이런 부류의 입력 양식은 서비스에 필요한 데이터를 사용자가 입력하고 버튼을 클릭해 서버에 요청하는 구조로 되어 있습니다. 이렇게 사용자와 상호작용을 할 수 있도록 만들어진 태그를 폼(Form) 태그라 합니다. 로그인 창 뿐 만이 아니라 회원 가입시 회원의 정보를 입력할 수 있는 입력 양식, 게시판에 글을 쓰거나, 웹에서 메일을 보낼 때 사용하는 입력 양식 등이 이 폼 태그를 통해 만들어져 있습니다. 자! 그럼 폼 태그에 대해 하나하나 자세히 알아보도록 합시다.

### 6.1 Form 정의하기

웹 페이지에서 서비스에 필요한 데이터를 사용자가 입력할 수 있도록 하고 입력된 데이터를 분석하여 사용자가 필요로 하는 서비스를 제공하거나 필요한 데이터를 보여주는 것을 상호작용이라고 했습니다. 이렇게 사용자가 입력한 데이터를 서버로 전송하기 위해서는 입력 양식 태그들이 <form> 태그 안에 위치해야 합니다. 물론 <form> 태그 밖에 위치한 입력 양식도 화면에 이상 없이 잘 보여지지만 입력된 데이터를 서버로 전송하기 위해서는 반드시 <form> 태그로 감싸 줘야 합니다. 다음은 <form> 태그의 기본 사용법과 속성을 설명한 것입니다.

```
<form name="폼을 구분할 이름" action="폼을 전송할 url"
      method="전송방식" enctype="서버로 보낼 폼 데이터의 인코딩 형식">
  <!-- 여기에 여러 가지 입력양식을 이용해 폼을 구성합니다. -->
</form>
```

표 6-1 form 태그의 속성

속 성	설 명	
name	하나의 문서에 여러 개의 <form> 태그가 있을 때 폼을 구분하기 위해 사용되며 주로 자바스크립트에서 폼에 접근하거나 서버사이드 스크립트와 같은 프로그램에서 사용자가 입력한 데이터를 읽을 때 사용됩니다.	
action	사용자가 입력한 데이터를 처리할 서버 측 프로그램 위치를 지정합니다. 주로(**.php, **.asp, **.jsp 등등)	
method	어떤 방식의 http 프로토콜로 자료를 전송할지 지정하는 속성	
	get	action에 지정된 경로에 ? 뒤에 데이터를 추가하여 전송하는 방식으로 페이지번호, 검색어 등 주로 보안이 필요하지 않은 데이터를 전송할 때 사용합니다. 최대 4KB 전송가능 method를 지정하지 않으면 기본 값은 get 방식이 적용됩니다.

	post	http 요청의 body에 포함되어 데이터를 전송하는 방식으로 회원가입, 로그인, 데이터베이스 변경 등의 보안이 필요한 데이터를 전송할 때 주로 사용합니다. 무한대 전송가능
enctype	application/x-www-form-urlencoded : form 요소의 기본 값으로 모든 문자를 인코딩 합니다. 기본적으로 enctype 속성을 생략하면 이 설정이 적용됩니다.	
	multipart/form-data : 문자를 인코딩하지 않으며 주로 파일 업로드 등에 사용됩니다.	
	text/plain : 공백 + 문자로 변환되며 특수 문자는 인코딩 하지 않습니다.	

## 6.2 텍스트 입력상자 만들기

먼저 입력 양식의 기본이라 할 수 있는 텍스트 입력상자에 대해 알아보시다.

웹 페이지에서 사용자가 글자를 입력하는 예는 로그인시 아이디나 이름과 같이 짧은 글을 입력하는 경우나 방명록이나 블로그에 글을 쓰는 것과 같이 여러 줄의 긴 글을 입력하는 경우 그리고 비밀번호와 같이 입력된 글이 타인에게 노출되지 않도록 별표 표시와 같이 대체 문자로 입력해 주는 경우 등을 들 수 있을 것입니다. 아이디나 비밀번호는 입력된 글자가 화면에 표시되는 모양만 다를 뿐 모두 짧은 글을 입력하는 폼 요소로 태그 사용법과 속성은 아래와 같습니다.

```
<input type="text" name="id" size="20" maxlength="10" value="홍길동" />
```

```
<input type="password" name="id" size="20" maxlength="10" value="" />
```

표 6-2 한 줄 텍스트 입력상자와 비밀번호 입력상자의 속성

속 성	설 명	
type	text	한줄 텍스트 입력상자를 지정하는 type 속성 값
	password	비밀번호 입력상자를 지정하는 type 속성 값
name	요소를 구분하기 위해 지정하는 이름으로 자바스크립트에서 폼에 접근 하거나 서버사이드 스크립트에서 사용자가 입력한 데이터를 읽을 때 주로 사용됩니다. <form> 태그 안에 사용되는 입력 양식을 폼 컨트롤이라고 하며 폼이 서버로 전송될 때 폼 컨트롤에 입력된 데이터를 서버에서 읽기 위한 이름으로 사용됩니다. 즉 서버에서 사용자가 입력한 데이터를 구분하기 위한 이름으로 사용됩니다.	
size	텍스트 입력상자의 가로 폭을 지정하는 속성	
value	텍스트 입력상자에 기본으로 입력될 값을 지정하는 속성	
maxlength	텍스트 입력상자에 최대 입력할 수 있는 글자 수를 지정하는 속성	

위의 코드와 같이 텍스트와 비밀번호 입력상자는 태그와 속성이 거의 같습니다. 다만 type 속성이 한줄 텍스트 입력 상자는 text로 지정하고 비밀번호 입력 상자는 password로 지정하면 됩니다. 다음은 여러 줄의 긴 글을 입력 할 수 있는 텍스트 입력상자의 태그 사용법과 속성입니다.

```
<textarea name="content" cols="50" rows="10">
  <!-- 실행되면서 초기에 보여질 텍스트를 여기에 입력할 수 있습니다. -->
</textarea>
```

표 6-3 textarea 태그의 속성

속 성	설 명
name	요소를 구분하기 위해 지정하는 이름으로 자바스크립트에서 폼에 접근 하거나 서버사이드 스크립트에서 사용자가 입력한 데이터를 읽을 때 주로 사용됩니다. <form> 태그 안에 사용되는 입력 양식을 폼 컨트롤이라고 하며 폼이 서버로 전송될 때 폼 컨트롤에 입력된 데이터를 서버에서 읽기 위한 이름으로 사용됩니다. 즉 서버에서 사용자가 입력한 데이터를 구분하기 위한 이름으로 사용됩니다.
cols	여러 줄 텍스트 입력 상자의 너비로 최대 열의 글자 수를 의미합니다.
rows	여러 줄 텍스트 입력 상자의 높이로 최대 글의 줄 수를 의미합니다.

## ▶ 텍스트 입력상자 만들기

- webstudy06\_01.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>텍스트 상자 만들기</title>
</head>
<body>
  <h2>텍스트 상자 만들기</h2>
  <form name="form1" action="#" method="post">
    아 이 디 : <input type="text" name="name" maxlength="5" size="15" /><br/>
    비밀번호 : <input type="password" name="pass" maxlength="10"
      size="15" /><br/>
    하고 싶은 말 : <br/>
    <textarea name="text" cols="50" rows="10">
      정말로 하고 싶은 말만 적어주세요
    </textarea>
  </form>
</body>
</html>
```

## 6.3 선택을 위한 양식 만들기

앞에서 살펴본 입력 양식은 사용자가 키보드를 이용해 입력할 수 있는 양식이었다. 이번에는 사용자가 일일이 내용을 입력하는 것이 아니라 여러 개의 항목 중에서 하나 또는 그 이상을 선택할 수 있도록 하는 입력 양식에 대해 알아봅시다. 위에서 `<input>` 태그를 이용해 한줄 텍스트 입력상자와 비밀번호 입력상자를 `type` 속성으로 구분하여 입력상자를 만들었듯이 여러 개의 항목 중 단 하나만 선택 할 수 있는 양식과 다중으로 선택 할 수 있는 폼 컨트롤도 `<input>` 태그의 `type` 속성을 이용해 만들 수 있으며 `<select>` 태그를 사용하면 화면을 적게 차지하면서 드롭다운 형태의 항목을 선택할 수 있는 폼 컨트롤도 만들 수 있습니다.

### 6.3.1 라디오 버튼과 체크박스 만들기

`<input>` 태그의 `type` 속성을 이용하여 만들어진 단일선택 입력 양식을 라디오 버튼이라 부르며 다중 선택 입력 양식을 체크박스라 부릅니다. 라디오 버튼과 체크박스는 항목의 `name` 속성을 모두 동일하게 지정하여 그룹화 하고 단일 선택 또는 다중 선택이 가능하도록 하는 기능을 가지는 선택 양식입니다. 아래는 라디오 버튼과 체크박스를 만드는 방법과 태그 속성입니다.

#### ▶ 라디오 버튼 만들기

관심분야 : `<br/>`

```
<input type="radio" name="interest" value="자바스크립트" checked />자바스크립트
```

```
<input type="radio" name="interest" value="HTML5+CSS3" />HTML5+CSS3
```

```
<input type="radio" name="interest" value="Android" />Android
```

#### ▶ 체크박스 만들기

읽고 싶은 도서 : `<br/>`

```
<input type="checkbox" name="book"
```

```
  value="자바스크립트 완벽 가이드" />자바스크립트 완벽 가이드
```

```
<input type="checkbox" name="book"
```

```
  value="JavaScript Cookbook" checked />JavaScript Cookbook
```

```
<input type="checkbox" name="book"
```

```
  value="자바스크립트 마스터북" />자바스크립트 마스터북
```

표 6-4 라디오 버튼과 체크박스의 속성

속 성	설 명	
type	radio	라디오 버튼을 만드는 type 속성 값
	checkbox	체크박스를 만드는 type 속성 값
name	요소를 구분하기 위해 지정하는 이름으로 자바스크립트에서 폼에 접근 하거나 서	

	<p>버사이드 스크립트에서 사용자가 입력한 데이터를 읽을 때 주로 사용됩니다.</p> <p>&lt;form&gt; 태그 안에 사용되는 입력 양식을 폼 컨트롤이라고 하며 폼이 서버로 전송될 때 폼 컨트롤에 입력된 데이터를 서버에서 읽기 위한 이름으로 사용됩니다. 즉 서버에서 사용자가 입력한 데이터를 구분하기 위한 이름으로 사용됩니다.</p> <p>여러 개의 라디오 버튼에 동일한 이름을 지정하여 그룹으로 묶고 같은 이름을 가진 여러 개의 라디오 버튼 중에서 오로지 하나만 선택할 수 있도록 동작합니다. 체크박스도 라디오 버튼과 마찬가지로 여러 개의 체크박스에 같은 이름을 부여해 그룹으로 묶지만 하나만 선택할 수 있는 것이 아니라 여러 개의 체크박스를 선택할 수 있도록 동작합니다.</p>
value	각 요소가 가지는 값을 지정하는 속성
checked	<p>checked 속성이 지정되면 기본으로 선택되어 나타납니다.</p> <p>체크박스는 여러 개 지정할 수 있으며 라디오 버튼은 여러 개 지정했을 때 가장 마지막에 지정한 요소만 선택되어 나타납니다.</p> <p>checked 만 지정할 수 있고 checked="checked"로 지정할 수도 있습니다.</p>

## ▶ 라디오 버튼과 체크박스 만들기

- webstudy06\_02.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>라디오 버튼과 체크박스 만들기</title>
</head>
<body>
  <h2>라디오 버튼과 체크박스</h2>
  <form name="form1">
    <p>이 름 : <input type="text" name="name" maxlength="5" size="20" /></p>
    <p>나이 : <input type="text" name="age" maxlength="3" size="20" /></p>
    <p>희망 연봉 : <br/>
    <input type="radio" name="salary" value="3000" checked />3000만원
    <input type="radio" name="salary" value="4000" />4000만원
    <input type="radio" name="salary" value="5000" />5000만원</p>
    <p>희망 지역 : <br/>
    <input type="checkbox" name="area" value="1" checked />수도권<br/>
    <input type="checkbox" name="area" value="2" />충청지역<br/>
    <input type="checkbox" name="area" value="3" />기타지역<br/>
  </form>
</body>
</html>

```

### 6.3.2 선택 상자 만들기

선택을 위한 양식 중에서 단일 선택만 할 수 있는 라디오 버튼과 다중 선택이 가능한 체크박스를 알아보았습니다. 이번에는 선택을 위한 양식 중에 속성 지정에 따라 단일 선택과 다중 선택이 가능한 선택 양식인 선택 상자(콤보박스 또는 드롭다운 리스트라고도 부름)에 대해 알아보시다.

마우스로 클릭하여 여러 개의 목록을 펼치거나 접히게 하는 선택 양식을 콤보박스라고 부르는데 이 양식을 만들기 위해서는 아래와 같이 <select>와 <option> 태그를 사용합니다.

```
<p>좋아하는 음악 장르 : <br/>
<select name="interest_music" multiple>
  <option value="pop">팝송</option>
  <option value="classic">클래식</option>
  <option value="jazz">재즈</option>
  <option value="kpop">가요</option>
  <option value="etc">기타</option>
</select>
```

표 6-5 선택 상자의 태그와 속성

태 그	속 성	설 명
select	name	요소를 구분하기 위해 지정하는 이름으로 자바스크립트에서 폼에 접근 하거나 서버사이드 스크립트에서 사용자가 입력한 데이터를 읽을 때 주로 사용됩니다. <form> 태그 안에 사용되는 입력 양식을 폼 컨트롤이라고 하며 폼이 서버로 전송될 때 폼 컨트롤에 입력된 데이터를 서버에서 읽기 위한 이름으로 사용됩니다. 즉 서버에서 사용자가 입력한 데이터를 구분하기 위한 이름으로 사용됩니다.
	size	콤보박스가 기본으로 보여줄 목록의 개수를 지정 size를 2이상 지정하면 리스트 박스 형태로 보여 집니다.
	multiple	이 속성을 지정하면 리스트 박스 형태로 보여 지며 목록을 여러 개 선택할 수 있는 다중 선택 모드가 됩니다.
option	value	각 항목의 값을 지정 하는 속성
	selected	selected가 지정되면 기본적으로 선택되어 나타납니다. selected만 지정할 수도 있고 selected="selected"도 가능

#### ▶ 선택 상자 만들기

- webstudy06\_03

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
```



```

<title>선택 상자 만들기</title>
</head>
<body>
  <h2>선택 상자 만들기</h2>
  <form name="form1">
    <p>자주 이용하는 서점 : <br/>
    <select name="purchasing_office" multiple>
      <option value="yes24" selected>예스24</option>
      <option value="kyobobook">교보문고</option>
      <option value="aladin">알라딘</option>
      <option value="ypbooks">영풍문고</option>
      <option value="etc">기타</option>
    </select>
  </form>
</body>
</html>

```

## 6.4 여러 종류의 버튼 만들기

일반적으로 사용자가 회원가입 폼이나, 게시판 입력 폼에 필요한 정보를 입력하고 서버로 전송하기 위해 사용하는 것이 버튼입니다. <form> 태그 안에서 정보를 서버로 전송하거나, 입력한 데이터의 유효성 체크 등의 작업을 위해 사용되는 버튼의 종류는 하나가 아니라 여러 개가 존재합니다. 지금부터 버튼 사용법에 대해 자세히 알아보도록 합시다.

### 6.4.1 전송 버튼(submit, image)과 입력취소 버튼

전송 버튼과 이미지 버튼은 입력된 정보를 서버로 전송하는 기능을 가지고 있는 버튼으로 이 버튼이 사용자에게 의해 클릭되면 폼 안에 입력된 정보가 <form> 태그의 action에 지정된 페이지로 무조건 전송됩니다. 또한 이미지 버튼은 입력된 정보를 서버로 전송 할 때 클릭된 위치 좌표도 함께 전송합니다. 이렇게 서버로 전송하는 기능을 하는 버튼을 submit 버튼이라 합니다. submit 버튼은 클릭이 되면 입력된 정보를 서버로 전송하기 때문에 서버로 전송되기 전에 입력된 데이터가 유효한 정보인지를 체크하여 유효하지 않은 정보일 경우 서버로 전송되는 동작을 취소해야 합니다. 만약 유효하지 않은 데이터가 서버로 전송될 경우 프로그램의 오류를 발생 시킬 수 있으며 또한 유효하지 않은 데이터가 데이터베이스에 저장될 수도 있습니다. 사용자 입력 정보의 유효성 검사는 주로 자바 스크립트를 사용해 체크하거나 Ajax와 같은 기술을 사용해 체크하기도 합니다.

입력취소 버튼은 사용자가 폼 양식에 입력한 정보를 웹 문서가 맨 처음 실행된 상태로 초기화 시키는 기능을 제공하는 reset 버튼입니다. 이 세 가지 버튼은 <input> 태그의 type 속성에 각각의 속성 값을 아래와 같이 지정하여 만들 수 있습니다.

```

<form action="authentication">
  <input type="submit" value="인증하기" />
  <input type="image" src="images/btn_authentication.gif" alt="인증하기"/>

```

```
<input type="reset" value="다시쓰기" />
</form>
```

표 6-6 submit 버튼과 reset 버튼의 특징

속 성	설 명	
type	submit	폼 전송 버튼을 만듭니다. 폼 전송 버튼이 클릭되면 폼에 입력된 모든 데이터가 action에 지정된 서버 위치로 전송됩니다.
	image	이미지로 된 폼 전송 버튼을 만듭니다. 이미지로 된 폼 전송 버튼도 submit 버튼과 동일한 동작을 합니다. 이미지 버튼은 src 속성을 사용하여 버튼으로 사용할 이미지를 지정할 수 있으며 alt 속성도 지정할 수 있습니다.
	reset	폼 입력 양식의 초기화 버튼을 만듭니다. 웹 페이지가 처음 실행되었을 때 값으로 초기화 됩니다.
value	value 속성에 지정된 문자열이 버튼의 텍스트로 표시됩니다. 이미지 버튼은 이 속성을 지정해도 무시됩니다.	

## 6.4.2 명령 버튼

명령 버튼은 버튼이 클릭되었을 때 특정 명령을 실행 할 수 있는 버튼으로 푸시버튼(Push Button)이라고도 합니다. 주로 자바스크립트를 사용하여 버튼의 이벤트 속성에 특정 동작을 정의한 함수를 연결하면 그 이벤트가 발생 했을 때 연결된 함수가 실행됩니다. HTML에서는 비록 입력 양식은 아니지만 입력 양식의 버튼들과 동일한 모양과 동작을 할 수 있는 태그를 따로 제공하는데 이 태그가 바로<button> 태그입니다. 이 <button> 태그의 type 속성을 지정하여 submit, reset, push 버튼을 만들 수 있습니다. <button> 태그는 내용을 가질 수 있으므로 여는 태그와 닫는 태그 사이에 내용을 기재해야 됩니다. <button> 태그의 내용으로 텍스트뿐만 아니라 이미지를 사용하여 이미지 버튼도 만들 수 있습니다. 명령 버튼과 <button> 태그 사용법은 아래와 같습니다.

```
<button type="submit" name="btn_submit">전송하기</button>
<button type="reset" name="btn_reset">다시쓰기</button>
<button type="button" name="btn_click"
  onclick="clickAlert('홍길동')">클릭</button>
<input type="button" value="클릭해 보삼" onclick="clickAlert('홍길동')" />
```

### ▶ 폼에 버튼 추가하기

- webstudy06\_04.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
```



```

</tr>
<tr>
  <td colspan="2" class="buttons">
    <!-- input 태그를 이용해 아래와 같이 폼에 버튼을 추가할 수 있습니다.
-->

    <input type="submit" value="전송하기" />
    <input type="reset" value="다시쓰기" />
    <input type="button" value="우편번호 찾기"
      onclick="findZipcode('222-333')" />
  </td>
</tr>
<tr>
  <td colspan="2" class="buttons">
    <!-- button 태그를 이용해 아래와 같이 폼에 버튼을 추가할 수 있습니다.
-->

    <button type="submit" value="submit">전송하기</button>
    <button type="reset">다시쓰기</button>
    <button type="button"
      onclick="findZipcode('222-333')>우편번호 찾기</button>
  </td>
</tr>
</table>
</form>
</body>
</html>

```

위의 예제 6-4 코드에서 input 태그의 name="zipcode"인 요소를 살펴보면 readonly 속성을 볼 수 있는데 이 속성은 텍스트를 입력할 수 있는 한줄 텍스트 입력상자, 비밀번호 입력상자, 여러 줄 텍스트 입력상자에 지정할 수 있는 속성으로 이 속성이 지정되면 텍스트 입력상자의 데이터를 읽을 수만 있으며 새로운 텍스트를 입력하거나 수정할 수 없는 읽기전용으로 화면에 표시됩니다. 코드에 서처럼 HTML5 방식으로 readonly만 지정할 수도 있고 XHTML 방식으로 readonly="readonly"와 같이 지정할 수 있습니다. 이외에도 폼 입력 양식을 비활성화 시키는 disabled 속성이 있는데 이 속성이 지정되면 웹 페이지가 로드될 때 폼 구성요소를 선택 할 수 없게 만듭니다. disabled 또는 disabled="disabled" 형태로 지정할 수 있으며 수정 자체가 불가능하고 데이터 전송도 되지 않습니다. disabled 속성을 버튼에 적용하면 클릭할 수 없는 상태로 표시되는데 자바스크립트를 이용해 특정 입력양식에 유효한 데이터가 입력되었을 때 이 비활성 된 버튼을 활성화 시켜 버튼을 사용할 수 있도록 하는 기법도 웹 프로그래밍에서는 자주 사용되는 기법입니다. readonly 속성을 가진 텍스트 입력양식은 사용자가 데이터를 수정할 수 없는 것 뿐 이지 자바스크립트로 접근하여 얼마든지 데이터를 수정할 수 있으며 데이터 전송도 가능합니다.

## ▶ 연습문제 6-1 회원 정보 입력 폼 만들기

다음 그림 6-1과 같이 실행되는 간단한 회원 정보 입력 양식을 작성하시오.

**회원 정보 입력 폼**

이름	<input style="width: 100%;" type="text"/>		
휴대폰 번호	<input style="width: 50px;" type="text" value="010"/>	- <input style="width: 50px;" type="text"/>	- <input style="width: 50px;" type="text"/>
<input type="button" value="인증하기"/>			
직업	<input style="width: 100%;" type="text" value="직업을 선택하세요"/>		
취미	<input type="checkbox"/> 영화감상 <input type="checkbox"/> 골프 <input type="checkbox"/> 낚시 <input type="checkbox"/> 기타		
<input type="button" value="가입하기"/> <input type="button" value="다시쓰기"/>			

그림 6-1 회원 정보 입력 폼

### 6.4.3 파일 선택 버튼과 숨겨진 입력양식

지금까지 여러 입력 양식과 기본 버튼에 대해 알아보았습니다. 이번에는 입력양식에 파일을 첨부하기 위해 필요한 파일 선택 버튼과 사용자의 입력과는 관계가 없지만 웹 프로그래밍에서 자주 사용되는 숨겨진 입력 양식에 대해 알아보시다. 파일 선택 버튼은 사용자 컴퓨터에 있는 파일을 선택하여 서버로 전송할 수 있도록 파일 선택 기능을 제공하는 버튼입니다. 이 파일 선택 버튼은 한줄 텍스트 입력상자와 버튼으로 구성되어 있습니다. 브라우저마다 표시되는 형태가 약간씩 다르며 버튼을 클릭하게 되면 파일을 선택할 수 있는 창이 화면에 나타납니다. 숨겨진 입력 양식은 웹브라우저에는 나타나지 않지만 주로 프로그램에서 필요한 데이터를 처리하기 위해 사용됩니다. 파일 선택 버튼과 숨겨진 입력양식의 사용법은 아래와 같습니다.

```
<form name="form" method="post" enctype="multipart/form-data">
  <!-- 파일 선택상자 만들기 -->
  <input type="file" name="file"/>
  <!-- 숨겨진 입력 양식 만들기
  <input type="hidden" name="name" value="홍길동" />
</form>
```

위의 코드에서처럼 파일 선택 상자는 <input> 태그의 type 속성을 file로 지정하고 숨겨진 입력 양식은 hidden으로 지정합니다. 또한 파일을 전송하기 위해서는 <form> 태그의 method 속성을 post로 지정하고 enctype을 multipart/form-data로 지정해야 합니다. 위의 코드에서는 숨겨진 입력 양식의 value 속성에 값을 지정하였지만 실무에서는 자바스크립트나 서버사이트 스크립트를 이용해 그 때 그 때 상황에 맞는 값을 value 속성에 지정합니다.

#### ▶ 파일 선택 버튼과 숨겨진 입력양식 만들기



```

        </table>
    </form>
</body>
</html>

```

#### 6.4.4 입력양식 그룹으로 묶고 폼 컨트롤에 제목 붙이기

폼 양식 안에서 관련된 정보를 입력받게 되는 여러 개의 입력 양식을 테두리로 묶어 그룹화 시켜주는 태그가 <fieldset> 태그입니다. 그리고 <fieldset> 태그 안에서 사용되는 <legend> 태그는 그룹의 제목을 나타내는 용도로 사용 됩니다. <legend> 태그가 <fieldset> 태그로 그룹화 된 그룹의 제목을 나타내는 것처럼 입력 양식 각각의 제목을 표시할 때는 <label> 태그를 사용합니다. 이 <label> 태그는 <input>, <select>, <textarea> 태그의 제목을 나타내는데 사용됩니다. 각각의 입력 양식에 id 속성을 지정하고 <label> 태그에서 그 id를 for 속성의 값으로 지정하거나 입력 양식을 <label></label>로 감싸 주면 그 입력 양식의 제목 필드가 됩니다. 텍스트 입력상자나 드롭다운 리스트 그리고 여러 줄 텍스트 입력상자의 제목 필드를 마우스로 클릭하면 해당 텍스트 입력상자로 포커스가 이동하여 키보드를 통해 바로 입력할 수 있도록 해 준다. 또한 체크박스나 라디오 버튼의 제목 필드를 클릭하게 되면 <label> 태그의 for 속성에 지정한 체크박스나 라디오 버튼이 선택됩니다. <label></label> 태그로 감싼 체크박스나 라디오 버튼도 동일하게 동작합니다.

#### ▶ 입력 양식을 그룹화 하고 폼 컨트롤에 제목 붙이기

- webstudy06\_06.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>입력양식 그룹화 하고 폼 컨트롤에 이름 지정하기</title>
</head>
<body>
    <h2>fieldset과 label 태그</h2>
    <form name="form1">
        <fieldset>
            <legend>회원 기본정보</legend>
            <label for="id">아이디 : </label>
            <input type="text" name="id" maxlength="5" id="id" /><br/><br/>
            <label for="pass">비밀번호 : </label>
            <input type="password" name="pass" id="pass" /><br/><br/>
            <label for="pass_confirm">비밀번호 확인: </label>
            <input type="password" name="pass_confirm" id="pass_confirm" /><br/><br/>
        </fieldset>
        <fieldset>
            <legend>회원 추가정보</legend>
            <label for="job">직 업 :</label>
            <select name="job" id="job">

```

```

        <option>직업을 선택하세요</option>
        <option value="공무원">공무원</option>
        <option value="회사원">회사원</option>
        <option value="학생">학생</option>
        <option value="기타">기타</option>
    </select><br/><br/>
    메일 수신여부 :<input type="checkbox" name="mail_receive" id="mail_receive" />
    <label for="mail_receive">정보 메일 받음</label><br/><br/>
    결혼 여부 : <label><input type="radio" name="marital_status">기혼</label>
    <label><input type="radio" name="marital_status">미혼</label><br/><br/>
    <label for="memo">하고 싶은말 : </label><br/>
    <textarea cols="50" rows="5" name="memo" id="memo"></textarea>
</fieldset>
<div>
    <input type="submit" value="가입하기" />
    <input type="reset" value="다시 쓰기" />
</div>
</form>
</body>
</html>

```

위의 예제에서 폼 양식에 <fieldset> 태그와 <legend> 태그를 사용하여 회원 기본정보와 회원 추가 정보를 그룹으로 묶어 구성하였습니다. 앞서도 언급했듯이 관련된 정보를 입력 받을 수 있는 여러 입력 양식을 테두리로 표시해 그룹화 시켜주는 요소가 <fieldset> 태그입니다. 이 <fieldset> 태그 안에서 사용되는 <legend> 태그는 그룹의 제목을 나타내는 용도로 사용됩니다.

## ▶ 연습문제 6-2 회원 가입 폼 작성하기

지금까지의 예제를 참고하여 다음 그림 6-2와 같이 실행되는 회원가입 양식을 작성하시오.

앞의 예제와 같이 <fieldset> 태그와 <legend> 태그를 사용하여 회원 기본정보와 회원 신상정보로 각각 그룹화 하여 작성하고 <label> 태그를 사용하여 체크박스과 라디오 버튼 등의 폼 컨트롤에 이름을 지정해 작성해 보시다.



## 회원정보입력

회원정보는 개인정보취급방침에 따라 안전하게 보호되며 회원님의 명백한 동의 없이 공개 또는 제3자에게 제공되지 않습니다. [개인정보취급방침](#)

✔표시는 필수입력사항

### 회원 기본정보

✔이름	<input type="text"/>
✔아이디	<input type="text"/> <input type="button" value="중복확인"/>
✔비밀번호	<input type="password"/>
✔비밀번호 확인	<input type="password"/>
✔이메일	<input type="text"/> @ <input type="text"/> <input type="button" value="-선택하세요-"/>

### 회원 신상정보

✔주소	<input type="text"/> - <input type="text"/> <input type="button" value="우편번호 찾기"/> <input type="text"/>
✔휴대폰	010 <input type="text"/> - <input type="text"/>
직업	<input type="button" value="직업을 선택하세요"/>
취미	<input type="checkbox"/> 영화감상 <input type="checkbox"/> 독서 <input type="checkbox"/> 골프 <input type="checkbox"/> 낚시 <input type="checkbox"/> 여행 <input type="checkbox"/> 기타
가입경로	<input type="radio"/> TV <input type="radio"/> 신문/잡지 <input type="radio"/> 인터넷 <input type="radio"/> 지인 <input type="radio"/> 기타

그림 6-2 회원가입 양식

CSS

# 1. CSS의 개요와 기본 사용법

## 1.1 CSS(Cascading Style Sheet)

CSS는 웹 페이지의 내용적인 부분과 분리해서 따로 지정한 서식을 의미하며 스타일시트라고도 부릅니다. 이 스타일시트는 HTML 태그만을 사용하여 웹 문서를 작성할 때의 디자인 적인 한계를 보완하기 위해 개발된 독립적인 마크업 언어입니다. HTML은 웹 문서를 빠르고 간단하게 작성하여 인터넷을 통해 정보를 공유할 수 있도록 고안된 마크업 언어입니다. HTML은 웹 문서의 내용을 빠르고 간편하게 작성하는데 많은 비중을 두고 개발된 언어로써 웹 페이지의 레이아웃, 타이포그래피, 배경, 테두리 등의 디자인 적인 스타일을 적용하기에는 많이 부족한 것이 사실입니다. CSS는 HTML의 디자인적인 한계를 보완하고 문서의 내용과 디자인 적인 요소를 분리하여 문서를 효율적으로 관리하기 위해 많이 사용됩니다.

CSS도 W3C(World Wide Web Consortium)에서 제정한 웹 표준 중에 하나이며 HTML5는 2014년 10월 정식 권고안이 발표되었고(<http://www.w3.org/TR/html5> 참고) 2017년 12월에 5.2 버전의 권고안이 발표된 상태입니다. 그리고 CSS3와 ECMAScript6(자바스크립트의 공식 명칭)도 표준 권고안이 발표된 상태입니다. CSS3는 기존의 CSS2.1에서 지원하던 여러 가지 기능을 더 확장해 지원하고 있습니다. 예를 들어 둥근 테두리는 CSS2.1에서는 지원하지 못하였으나 CSS3를 사용하면 둥근 테두리가 있는 박스를 쉽게 만들 수 있으며 웹 문서의 객체를 더 편리하게 선택 할 수 있도록 많은 선택자가 추가되었습니다. 현재는 div 요소와 CSS를 사용하여 웹 페이지 레이아웃을 구성하는 것은 가장 일반적인 레이아웃 기법으로 JavaScript나 jQuery와 같은 프로그램과 연동하여 애니메이션, 동적 레이아웃 구성 등 CSS의 다양한 기능이 활용되고 있습니다.

## 1.2 웹 문서에 CSS를 적용하는 방법

CSS의 적용 방법은 크게 3가지로 나눌 수 있는데 그 첫 번째가 HTML 태그에서 style 속성으로 스타일을 정의하여 사용하는 인라인 스타일 방식이 있고 두 번째로 웹 문서의 내부에 스타일시트를 정의하여 사용하는 내부 스타일시트 방식이 있으며 마지막 하나는 별도의 외부 파일에 스타일시트를 정의하고 현재의 웹 문서에서 외부에 있는 스타일시트 파일을 연결하여 사용하는 외부 스타일시트 방식이 있습니다. 외부 스타일 방식에는 외부 스타일시트(External Style Sheet) 연결방식과 외부 스타일시트 포함(Imported Style Sheet) 방식이 있습니다. 이제 웹 문서에 CSS를 적용하는 방법을 하나하나 자세히 알아보도록 합시다.

### 1.2.1 인라인 스타일

인라인 스타일(Inline style)은 HTML 태그의 style 속성에 스타일을 정의하는 방식입니다.

인라인 스타일을 사용하는 방식은 아래와 같이 각 요소의 시작 태그에 style 속성을 사용해 CSS 속성 명(property)과 속성 값(value)을 콜론(:) 으로 구분하여 지정하고 하나의 속성 지정이 끝나면 스타일 문장의 끝을 알리는 세미콜론(;)을 기술합니다. 이런 방식으로 HTML 요소의 style 속성에 세미콜론(;)으로 구분하여 여러 개의 스타일 속성을 아래와 같이 연속적으로 지정할 수 있습니다. 여기서 세미콜론은 각 스타일을 구분하는 역할을 하므로 선언 구분자라고 부릅니다.

```
<body>
  <p style="속성 명1: 속성 값1; 속성 명2: 속성 값2;">내용</p>
  <p style="border: 1px solid red; padding: 5px;">내용</p>
</body>
```

### 1.2.2 내부 스타일시트

내부 스타일시트(Internal style sheet)는 HTML 문서의 <head></head> 태그 사이에 스타일을 정의합니다. 스타일을 정의하는 방식은 아래와 같이 <style> 태그를 사용하고 그 안에서 스타일을 적용할 문서 객체를 선택하는 선택자(selector)와 문서 객체에 지정할 스타일을 정의하는 선언부(declaratives)인 중괄호({})의 "{"와 "}" 사이에 CSS 속성 명(property)과 속성 값(value)을 콜론(:)으로 구분하여 지정합니다. 선택자 하나의 선언부에는 여러 스타일을 지정할 수 있으며 하나의 스타일 지정이 끝나면 스타일 문장의 끝을 알리는 세미콜론(;)을 기술합니다. CSS는 요소, 아이디, 클래스 등과 같은 다양한 선택자를 지원하고 있으며 선택자에 대해서는 다음 Chapter에서 자세히 알아볼 것입니다.

```
<head>
  <style type="text/css">
    선택자(selector) { 속성 명1: 속성 값1; 속성 명2: 속성 값1; }
    div { border: 1px solid red; background-color: #EAF7FF; }
  </style>
</head>
```

### 1.2.3 외부 스타일시트

위에서 언급했듯이 별도의 파일로 정의한 스타일을 웹 문서에 적용하는 방법에는 외부 스타일시트(External Style Sheet)를 연결하는 방식과 외부 스타일시트(Imported Style Sheet)를 포함하는 두 가지 방식이 있습니다.

#### ▶ HTML문서에 외부 스타일시트 연결하기

별도의 파일로 스타일을 정의하고 웹 문서에 외부 스타일시트 파일(External Style Sheet)을 연결하여 사용하는 방식으로 아래와 같이 <head></head> 태그 사이에 연결할 스타일시트 파일을 <link> 태그의 href 속성에 지정하여 연결합니다. 이 방식은 HTML 문법을 이용해 외부 스타일시트를 현재 문서에 연결하는 방식입니다.

```
<head>
  <link rel="stylesheet" type="text/css" href="external_style.css" />
</head>
```

\* external\_style.css 파일

```
선택자(selector) { 속성명1: 속성 값1; 속성명2: 속성 값2; }  
div { border: 1px solid red; background-color: #FF0000; }
```

## ▶ HTML 문서에 외부 스타일시트 포함하기

별도의 파일로 스타일을 정의하고 웹 문서에 스타일시트 파일을 포함(Imported Style Sheet)하여 사용하는 방식으로 내부 스타일시트 방식과 비슷하게 <head></head> 태그 사이에 <style> 태그를 사용하고 그 안에 CSS 구문을 기술하는 자리에 @import url('external\_style01.css');로 외부 스타일시트 파일을 지정하는 방식입니다. 이 방식은 HTML 문법을 사용하는 것이 아니라 CSS 문법을 사용하여 외부 스타일시트 파일을 현재 문서의 스타일에 포함시키는 방식입니다. 이렇게 외부 스타일시트를 현재 문서에 적용하는 방식에는 HTML 문법을 이용하는 방식과 CSS 문법을 이용하는 방식 두 가지가 있습니다. 두 가지 방식 모두 스타일이 적용되는 결과는 동일합니다.

```
<head>  
  <style type="text/css">  
    @import url('external_style01.css');  
    @import url('external_style02.css');  
  </style>  
</head>
```

## 1.3 스타일 우선순위

웹 문서는 스타일이 여러 곳에서 사용되기 때문에 스타일 적용의 우선순위에 대해 잘 이해하고 있어야 합니다. 스타일은 상속의 개념과 CSS 용어 자체에 담긴 캐스케이딩(Cascading)의 개념이 적용됩니다.

### 1.3.1 스타일은 상속됩니다.

HTML 문서에서 사용하는 여러 요소들은 서로 포함 관계를 가지고 있습니다. 그래서 다른 요소를 포함하는 요소를 부모 요소라 하고 다른 요소에 포함된 요소를 자식 요소라고 합니다.

스타일에서 부모 요소에 적용된 속성이 자식 요소로 전달되는데 이를 상속이라 합니다.

한 가지 주의할 점은 스타일의 상속은 모든 속성이 전부 상속되는 것이 아니라 상속되는 속성이 있고 상속되지 않는 속성이 있다는 것입니다.

아래 URL 주소는 W3C에서 스타일 상속과 관련해서 정리한 문서입니다.

<http://www.w3.org/TR/2011/REC-CSS2-20110607/propidx.html>

## ▶ 예제 1-1 스타일의 상속 1

- css01\_01.html

```
<!DOCTYPE html>
```

```

<html>
<head>
<meta charset="UTF-8">
<title>스타일의 상속1</title>
<style type="text/css">
    body {
        background-color: #333;
        color: white;
        border: 1px solid blue;
    }
    div {
        border: 1px dotted red;
    }
    p{
        border: 1px dashed gray;
    }
</style>
</head>
<!--
    스타일의 상속은 모든 속성이 상속되는 것이 아니라 상속되는
    속성이 있고 상속되지 않는 속성이 있다는 것입니다.
    아래 URL 주소는 W3C에서 스타일 상속과 관련해서 정리한 문서입니다.
    http://www.w3.org/TR/2011/REC-CSS2-20110607/propidx.html
-->
<body>
    <div>
        <strong>body 요소 안의 div 요소</strong>
        <p><strong>body > div > p 요소</strong>
            div 요소는 body 요소의 자식 요소로 body 요소의 스타일이 상속됩니다.<br/>

            또한 p요소는 body 요소의 자손 요소로 body 요소의 스타일이 상속되며<br/>
            div 요소의 자식 요소로 div 요소의 스타일이 상속됩니다.
        </p>
    </div>
</body>
</html>

```

## ▶ 예제 1-2 스타일의 상속 2

css01\_02.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">

```

```

<title>스타일의 상속2</title>
<style type="text/css">
    ul {
        background: black;
        color: white;
        list-style: square;
    }
</style>
</head>
<body>
    <!--
        ul 요소는 li 요소의 부모 요소로 ul 요소에 지정한 color 속성이
        자식인 li 요소로 상속되어 글자색이 흰색으로 표현됩니다.
    -->
    <section>
        <h2>자바스크립트 도서 목록</h2>
        <ul>
            <li>자바스크립트 마스터 북</li>
            <li>자바스크립트 완벽 가이드</li>
        </ul>
        <h2>안드로이드 도서 목록</h2>
        <ul>
            <li>안드로이드 프로그래밍 정복</li>
            <li>이것이 안드로이드다</li>
        </ul>
    </section>
</body>
</html>

```

### 1.3.2 스타일은 단계적으로 적용됩니다.

CSS는 캐스케이딩 스타일시트(Cascading Style Sheet)의 줄임말로 단계적으로 스타일이 적용된다는 의미를 지니고 있습니다. 여러 문서에 스타일을 적용하다 보면 같은 요소에 여러 스타일이 겹쳐서 지정되는 경우가 있는데 이럴 경우 우선순위에 따라 스타일이 적용됩니다.

하나의 요소에 여러 스타일이 지정되어 있다면 맨 마지막에 지정된 스타일이 적용됩니다.

만약 다음과 같이 스타일이 지정되어 있다면 나중에 지정된 글자 크기인 12px이 적용됩니다.

```

<style type="text/css">
    p {
        font-size: 20px;
        font-size: 12px;
    }
</style>

```

<p>CSS는 캐스케이딩 스타일시트(Cascading Style Sheet)의<br/>  
줄임말로 단계적으로 스타일이 적용된다는 의미를 지니고 있습니다.</p>

이렇게 중복 지정된 스타일이 존재할 경우 예상하지 못한 결과가 나타날 수 있습니다.  
이럴 경우 아래와 같이 우선순위를 지정할 수 있는데 스타일 규칙에 !important를 지정하게 되면  
해당하는 스타일 규칙이 가장 높은 우선순위를 가지게 됩니다.

```
p {  
    font-size: 20px; !important;  
}
```

우리는 앞에서 스타일이 적용될 때 단계적으로 적용되며 같은 요소에 적용된 스타일은 가장 마지막  
에 지정된 스타일이 적용된다는 것을 알았다. 그렇다면 동일한 요소에 외부 스타일, 내부 스타일,  
인라인 스타일 모두에서 동일한 요소에 스타일 규칙을 지정하였다면 그 우선순위는 어떻게 될까?  
간단한 예제를 통해 이를 알아보시다.

### ▶ 예제 1-3 동일한 요소에 적용된 스타일 우선순위

- css01\_03.html

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>동일한 요소에 적용된 스타일 우선순위</title>  
<link type="text/css" href="css/external.css" rel="stylesheet" />  
<style>  
    div {  
        font-size: 14px;  
    }  
</style>  
</head>  
<body>
```

body 요소의 폰트 크기는 외부 스타일에서 지정한 20px이 적용됩니다.

```
<div>
```

div 요소의 폰트 크기는 내부 스타일에서 지정한 14px이 적용됩니다.

```
<!--
```

동일한 요소에 동일한 스타일이 적용되었다면 우선순위는 다음과 같습니다.

1. !important; 가 적용된 스타일 속성
2. 인라인 요소의 style 속성에 적용된 스타일 속성
3. 내부 스타일 속성



#### 4. 외부 스타일 속성

아래 p 요소의 폰트 크기는 external.css에서 p 요소의 font-size에 !important를 적용했기 때문에 20px이 우선 적용됩니다.

```
-->
<p style="font-size: 12px;">
  p요소의 폰트 크기는 인라인 스타일에서 지정한 12px이 적용되지 않고<br/>
  external.css에서 !important로 지정한 20px이 우선 적용됩니다.
</p>
</div>
</body>
</html>
```

\* WebContent/css/external.css

```
@CHARSET "UTF-8";
```

```
body {
```

```
    font-size: 40px;
```

```
}
```

```
div {
```

```
    font-size: 20px;
```

```
}
```

```
p {
```

```
    font-size: 20px !important;
```

```
}
```

## 1.4 CSS 작성시 필요한 기본기념 익히기

### 1.4.1 요소와 태그, 블록레벨 요소와 인라인레벨 요소 구분하기

#### ▶ 요소(Elements)

html에서 “<”와 “>”로 둘러싸인 시작 태그와 “</”와 “>”로 둘러싸인 종료 태그로 이루어진 태그의 조합을 의미합니다.

img, hr, br 등과 같이 내용물이 없는 요소를 비어있는 요소(Empty Element) 또는 비어있는 태그(Empty Tag)라고 부르며 비어있는 요소는 하나의 태그에 “<”와 “/>”를 사용해 요소의 시작과 끝을 표현할 수 있으며 이런 태그 들을 셀프 클로징 태그(Self Closing Tag)라고 부릅니다.

셀프 클로징 태그는 이전의 xhtml의 문법으로 HTML5에서는 반드시 사용해야 하는 것은 아니며 선택적(Optional)으로 사용할 수 있습니다.

#### ▶ 태그(Tag)

태그는 요소의 일부로 “<”와 “>”로 둘러싸인 시작 태그(시작 태그를 여는 태그라고도 함)와 “</”와

“>”로 둘러싸인 종료 태그(종료 태그를 닫는 태그라고도 함) 두 종류가 있으며 시작 태그는 요소의 시작을 종료 태그는 요소의 끝을 의미하지만 일반적으로 요소와 태그의 개념이 뚜렷하게 구별 되지 않고 혼용 되는 경우가 많다.

CSS나 자바스크립트에서 요소라는 의미는 매우 중요합니다. 요소는 “<h1>문서의 제목</h1>” 전체를 의미 하지만 태그는 <h1>과 </h1>을 각각 시작 h1 태그와 종료 h1 태그를 의미합니다. 예를 들면 <p>, <div>와 같이 여는 태그를 부를 때 “p 태그”, “div 태그” 라고 부릅니다. 하지만 <p></p>와 <div></div>와 같이 태그의 조합을 부를 때는 각각 “p 요소”, “div 요소”라고 부르는 것이 정확한 표현입니다.

## ▶ 블록레벨(Block-level) 요소(Element)

브라우저의 안에서 블록처럼 쌓여가면서 줄 바꿈을 하여 표시 되는 요소로 전후의 요소들과 줄 바꿈 되어 화면에 표시됩니다. 블록레벨 요소는 아래와 같은 특징을 가지고 있습니다.

- 한 라인에 하나의 요소만 표시되므로 항상 줄 바꿈을 하여 새로운 줄에 표시됩니다.
- 너비 값을 지정하지 않으면 브라우저 화면 또는 부모에 100%로 꽉 차게 표시됩니다.
- 블록레벨 요소는 width, height, margin, padding 등의 값을 가질 수 있습니다.
- 블록레벨 요소와 인라인레벨 요소를 자식요소로 포함 할 수 있습니다.
- vertical-align 속성이 무시되며 대표적인 블록레벨 요소는 아래와 같습니다.  
body, div, p, pre, form, h1 ~ h6, table, ul, ol, dl, menu, fieldset, noscript, hr, address, blockquote...등

## ▶ 인라인레벨(Inline-level) 요소(Element)

블록레벨과 반대의 개념으로 앞에 표시된 요소에 이어서 같은 줄 안에 표시되며 넓이도 자신의 내용물의 크기만큼 만 차지합니다. 인라인레벨 요소는 다음과 같은 특징을 가지고 있습니다.

- 너비 값은 요소가 가지고 있는 값 자체로 이므로 width, height, margin, padding 등의 값을 지정해도 화면에 반영되지 않습니다.
- 웹 표준을 지키면서 인라인 레벨 요소를 사용하고 싶다면 항상 블록레벨 요소로 인라인 레벨 요소를 감싸 기술해야 합니다. 즉 인라인 요소는 단독으로 사용될 수 없으며 div, p 요소와 같은 블록레벨 요소 안에 기술되어야 합니다.
- 대표적인 인라인레벨 요소는 아래와 같습니다.  
span, img, map, button, input, label, select, textarea, a, strong, em object, script, br, b, sub, sup, i, abbr, cite, samp, var... 등

## ▶ 그룹화 요소(Grouping Element)

예전에 Table과 Frame을 통해 작성되었던 화면 레이아웃 구성이 현재는 HTML5의 시멘틱 요소와 div, span 요소를 사용해 그룹화 하고 CSS를 이용해 디자인적인 요소를 지정하고 있습니다.

웹 페이지를 구성하는 콘텐츠를 화면에 배치하고 표현하기 위해 그룹화 할 수 있도록 지원하는 요소가 div, span 요소입니다. 웹 페이지 작성시 div 요소를 이용하여 여러 요소를 논리적으로 그룹화 할 수 있고 그 그룹에 이름을 지정하여 CSS를 이용해 스타일을 적용할 수 있습니다. div는

division의 약자로 블록레벨 요소에 속하며 div에 자식으로 속한 여러 요소를 그룹화 하여 공간을 지정하는 컨테이너 역할을 하는 요소입니다. div는 특정 의미를 가지고 있지 않는 요소로 블록레벨 요소를 그룹화 하여 스타일을 지정하기 위해 사용됩니다. span 요소는 인라인레벨 요소의 대표 요소로 div와 같이 특정 의미를 가지고 있지 않는 그룹화 요소입니다. span은 인라인레벨 요소를 그룹화하고 스타일을 적용하기 위해 사용됩니다. 인라인레벨 요소 중 span 이외의 그룹핑 요소로는 a, img, select, input 등이 있으며 이들은 특정 의미를 지닌 요소 들입니다. div 요소와 span 요소는 웹 페이지에서 블록 레벨 요소와 인라인 레벨 요소를 그룹으로 묶어서 웹 페이지의 공간을 분할하여 다양한 형태의 요소를 배치할 수 있도록 지원하는 요소로 공간 분할 요소 또는 공간 분할 태그라고도 부릅니다.

## ▶ 예제 1-4 블록레벨 요소와 인라인레벨 요소

### - css01\_04.html

이번 예제는 블록레벨 요소와 인라인레벨 요소의 특징을 잘 알 수 있는 예제입니다. 예제를 실행해 보면 css에서 지정한 width와 height가 블록레벨 요소인 h3 요소에만 적용되는 것을 볼 수 있습니다. 또한 인라인레벨 요소인 b 요소와 em 요소가 줄 바꿈 없이 한 라인에 표시됩니다. 그러나 나머지 블록레벨 요소는 전 후의 요소들과 앞뒤로 줄 바꿈이 일어난 것을 볼 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>블록레벨 요소와 인라인레벨 요소</title>
  <style type="text/css">
    * { font-size: 12px; }
    h2 {
      font-size: 16px;
      font-weight: bold;
    }
    h3 {
      width: 300px;
      height: 70px;
      color: white;
      font-size: 14px;
      font-weight: bold;
      background-color: black;
    }
    strong {
      width: 300px;
      height: 100px;
      color: #FFFFFF;
      background: #770000;
    }
  </style>
</head>
<body>
  <h2>블록레벨 요소</h2>
  <h3>블록레벨 요소</h3>
  <strong>인라인레벨 요소</strong>
</body>
</html>
```

```

    em {
        width: 500px;
        height: 100px;
        font-weight: bold;
        color: #FFFFFF;
        background: #FF0000;
    }
</style>
</head>
<body>
    <h2>블록레벨과 인라인레벨</h2>
    <div>
        <h3>블록레벨 요소</h3>
        <p>브라우저의 안에서 블록처럼 쌓여가면서 줄 바꿈을 하여 표시 되는 엘리먼트로
        <strong>전후의 요소들과 줄 바꿈</strong> 되어 화면에 표시됩니다.</p>
        <h3>인라인레벨 요소</h3>
        <p>블록레벨과 반대의 개념으로 <b>앞에 표시된 엘리먼트에 이어서 같은 줄 안에
        표시되며</b>    <em>넓이도 자신의 내용물의 크기만큼 만 차지합니다.</em></p>
    </div>
</body>
</html>

```

### 1.4.2 CSS 단위 이해하기

CSS 문서를 작성할 때 속성에 대한 값은 특정한 단위를 지정해 줘야합니다. 특히 width나 height 속성과 같이 크기에 대한 값을 입력할 때 단위를 생략하면 0 값을 제외한 나머지는 제대로 적용되지 않습니다. CSS에는 속성에 대한 값을 지정할 때 이미 예약된 키워드를 입력하는 경우도 있으며 크기를 지정할 때는 절대 또는 상대 크기의 단위를 지정해야 하는 경우도 있습니다. 또한 배경색이나 전경색을 지정하기 위해 색상을 수치로 입력하는 경우도 있고 배경의 이미지 등을 지정하기 위해 URL을 기술하는 경우도 있습니다. 이렇게 CSS에서 사용되는 속성 값에 대한 단위는 여러 가지가 존재합니다. 여기서는 HTML 요소의 크기를 지정할 수 있는 크기 단위와 색상과 관련된 색상 단위만 알아볼 것입니다. URL 지정 단위는 CSS를 Imported Style Sheet로 HTML에 연결할 때 다루어 봤듯이 이미지나 파일의 위치를 상대 또는 절대경로 방식으로 지정하는 것이므로 쉽게 이해할 수 있을 것입니다. 키워드를 속성의 값으로 지정하는 경우는 속성에 따라 지정하는 키워드 값이 서로 다르고 많기 때문에 여기서는 전부 다룰 수 없어 중요한 키워드 값을 중심으로 예제를 통해 다루어 볼 예정이니 나머지는 레퍼런스 등을 참조하기 바랍니다.

#### ▶ CSS의 크기 단위 이해하기

CSS3에서 사용되는 크기 단위는 mm, cm, in, pt, px, %, em, 가 있다 이 외에도 글꼴의 크기를 지정하는 경우 미리 예약된 키워드를 단위로 사용할 수 있으며 다음 표에서 설명하는 3가지 단위가 가장 많이 사용되고 있습니다.

표 1-1 자주 사용되는 CSS의 크기 단위

단 위	설 명
%	문서에서 기준이 되는 크기의 백분을 단위
em	문서에서 기준이 되는 크기의 배수 단위
px	픽셀 단위, 화면 해상도에 따라 보여 지는 크기가 다르다.

## ▶ CSS 색상 단위 이해하기

CSS 색상 단위는 글꼴의 색상을 지정하거나 요소의 전경색, 배경색을 지정할 때 사용되는 단위로 색상 값을 지정하는 키워드인 white, black, red... 등의 색상 값도 지정할 수 있으며 10진수나 16진수 값의 수치를 지정할 수도 있습니다. CSS에서 다음과 같은 색상 단위가 사용됩니다.

표 1-2 CSS의 색상 단위

단 위	설 명
#FF00AA	16진수 단위(HEX 코드 단위)
rgb(red, green, blue)	빛의 3원색인 Red, Green, Blue를 사용하는 단위
rgba(red, green, blue, alpha)	RGB에 투명도(Alpha)를 지정 할 수 있는 단위 투명도는 0.0(완전투명) ~ 1.0(불투명)까지 지정 할 수 있으며 CSS3에서 추가된 속성입니다.
hsl(hue, saturation, lightness)	색의 3속성인 색상, 명도, 채도를 사용하는 단위 CSS3에서 추가된 속성입니다.
hsla(hue, saturation, lightness)	HSL에 투명도(Alpha)를 지정 할 수 있는 단위 CSS3에서 추가된 속성입니다.

## ▶ 예제 1-5 CSS의 크기 단위와 색상 단위

- css01\_05.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>CSS의 단위 이해하기</title>
  <style type="text/css">
    /* font 관련 속성들은 상속되기 때문에 body 요소에 지정한
     * font-size에 따라 %와 em은 상대적인 크기로 표현됩니다.
     */
    /* body { font-size: 12px; } */
    p { width: 700px; height: 50px; }
    .per75 { font-size: 75%; }
    .per150 { font-size: 150%; }
  </style>
</head>
</html>

```

```

.per200 { font-size: 200%; }
.em075 { font-size: 0.75em; }
.em15 { font-size: 1.5em; }
.em20 { font-size: 2.0em; }
.px12 { font-size: 12px; }
.px16 { font-size: 16px; }
.px32 { font-size: 32px; }

.black { background-color: black; color: white; }
.white { background-color: rgba(255,0, 0, 0.5); color: black; }
.red { background-color: rgba(50%, 50%, 50%, 0.7); }
.green { background-color: hsla(30, 50%, 10%, 0.3); color: red; }
.blue { background-color: #0000FF; color: white; }
</style>
</head>
<body>
<h2>CSS의 크기 단위와 색상 단위 이해하기</h2>
<p>HTML 문서에서 폰트의 기본 크기는 16px로 화면에는 좀 크게 표시됩니다.</p>
<p>그래서 주로 css에서 폰트의 크기를 지정하여 사용하는 것이 일반적입니다.</p>
<p class="per75 black">기본 폰트 크기의 75%인 12px 크기</p>
<p class="per150 white">기본 폰트 크기의 150%인 24px 크기</p>
<p class="per200 red">기본 폰트 크기의 200%인 32px 크기</p>
<p class="em075 black">기본 폰트 크기의 0.75배인 12px 크기</p>
<p class="em15 blue">기본 폰트 크기의 1.5배인 24px 크기</p>
<p class="em20 red">기본 폰트 크기의 2.0배인 32px 크기</p>
<p class="px12 green">12px로 지정한 폰트 크기</p>
<p class="px16 white">16px로 지정한 폰트 크기</p>
<p class="px32 green">32px로 지정한 폰트 크기</p>
</body>
</html>

```

## 2. CSS 선택자

CSS에서 HTML 문서의 특정 요소에 스타일을 지정하기 위해 그 요소를 선택할 수 있는 태그나 클래스 또는 아이디 등을 일컫는 말로 선택하는 요소를 구분할 수 있는 명칭을 선택자(Selector)라 합니다. 또한 선택자는 HTML 문서의 특정 요소에 CSS를 통해 스타일을 적용하거나 효과를 적용하기 위해 사용되는 CSS 문법의 필수 구성요소입니다.

CSS에서 사용되는 선택자는 마크업 기반의 자바스크립트 프레임워크인 jQuery에서 대부분 사용할 수 있습니다. 다음은 CSS에서 사용할 수 있는 선택자의 종류와 설명을 표로 나타낸 것입니다. 언뜻 봐도 CSS에서 지원하는 선택자의 종류는 상당히 많은 수준입니다. 아래 표 2-1은 CSS에서 사용하는 선택자의 종류를 정리한 것입니다. 아래 표를 참고하여 선택자의 종류와 그 사용법에 대해 자세히 알아보도록 하자

표 2-1 CSS 선택자의 종류

선택자	선택자 사용 예	ver	설 명
전체(공통) 선택자	*	2	문서의 전체 요소를 선택
요소(태그) 선택자	p	1	p 요소 전체를 선택
아이디 선택자	#user	1	id 속성 값이 user인 요소 선택
클래스 선택자	.members	1	class 속성 값이 members인 요소 선택
자손 선택자	header h1	1	header 요소 아래의 h1 요소 전체 선택
자식 선택자	header > h1	1	header 요소 바로 아래의 h1 요소 선택
종속 선택자	p.row, p#row	1	p 요소의 클래스 값이 row인 p요소 선택 p 요소의 아이디 값이 row인 p요소 선택
그룹 선택자	p, h1, a	1	p, h1, a 요소를 모두 선택
속성 선택자	input[type]	2	type 속성을 가진 요소
속성 값 선택자	input[type=text]	2	type이 text인 요소를 선택
	div[class~=name]	2	class 속성 값이 name과 일치하거나 여러 개의 class 속성 값 중에서 name과 일치하는 요소선택
	div[id^=name]	3	id의 값이 name으로 시작하는 요소선택
	img[src\$=jpg]	3	src 속성 값이 jpg로 끝나는 요소 선택
	div[class*=user]	3	class 속성 값이 user를 포함하는 요소선택
인접 선택자	h1 + h2	2	h1 요소 바로 뒤에 위치하는 h2 요소 선택
	h1 ~ h2	3	h1 요소 뒤에 위치하는 h2 요소들 선택
링크 가상클래스 선택자	a:link	1	방문하지 않은 모든 링크를 선택
	a:visited	1	이미 방문했던 모든 링크를 선택

선택자	선택자 사용예	ver	설 명
반응 가상클래스 선택자	a:active, h1.active	1	활성화된 요소를 선택(클릭되고 있는 요소)
	a:hover, div.hover	1	마우스 오버된 요소를 선택
상태 가상클래스 선택자	input:focus	2	포커스된 요소를 선택
	input:checked	3	체크 상태인 input 요소를 선택
	input:enabled	3	사용 가능한 input 요소를 선택
	input:disabled	3	사용 불가능한 input 요소를 선택
위치구조 가상클래스 선택자	li:first-child	2	어떤 요소의 자식 중 첫 번째 li 요소 선택
	li:last-child	3	어떤 요소의 자식 중 마지막 번째 li 요소 선택
	li:nth-child(2n)	3	어떤 요소의 자식 중 2n번째 li 요소 선택
	li:nth-last-child(2n)	3	어떤 요소의 자식 중 뒤에서 2n번째 li 요소 선택
타입구조 가상클래스 선택자	h1:first-of-type	3	h1 요소 중 처음으로 등장하는 요소 선택
	h1:last-of-type	3	h1 요소 중 마지막으로 등장하는 요소 선택
	h1:nth-of-type(2n)	3	h1 요소 중 앞에서 2n 번째로 등장하는 요소 선택
	h1:nth-last-of-type(2n)	3	h1 요소 중 뒤에서 2n 번째로 등장하는 요소 선택
문자 가상요소 선택자	p::first-letter p:first-letter	1	p 요소 안의 첫 번째 글자를 선택
	p::first-line, p:first-line	1	p 요소의 첫 번째 라인을 선택
	p::after, p:after	2	p 요소 내용의 뒤에 위치하는 공간을 선택
	p::before, p:before	2	p 요소 내용의 앞에 위치하는 공간을 선택
	p::selection, p:selection	3	사용자가 드래그 하여 선택한 텍스트를 선택
부정 선택자	input:not(type=text)	3	input 요소의 type 속성 값이 text가 아닌 것을 선택

CSS에서는 가상(Pseudo)이라는 개념을 적용해 HTML 문서에 실제로 존재하지 않아 접근할 수 없는 문서 모델(DOM-Document Object Model)에 접근하여 CSS 문서를 용이하게 작성할 수 있도록 해주는 가상요소 선택자와 가상 클래스 선택자가 있습니다. 스타일을 적용하는 대상을 가상요소에 포함된 내용의 일부나 위치를 선택 할 수 있는 선택자를 가상요소 선택자라 하며 스타일을 적용하는 대상을 요소명이나 속성명 또는 속성 값에 따라 선택하는 것이 아니라 요소의 특정 상황이나 상태에 따라 선택해 주는 선택자를 가상 클래스 선택자라 합니다. 위의 [표 2-1 CSS 선택자의 종류]를 참고하면 문자 가상요소 선택자인 p:first-letter은 HTML 문서의 문단으로 지정된 p요소 안에 첫 번째 글자를 선택하여 스타일을 적용할 수 있게 해 주고 상태 가상클래스 선택자인 input:checked는 HTML 문서의 input 요소 중에서 체크 상태인 모든 input 요소를 선택할 수 있도록 해 준다.

## 2.1 전체 선택자와 요소(태그) 선택자

전체 선택자는 HTML 문서 내의 모든 태그를 선택 할 때 사용하는 선택자로 문서의 여백이나 마진



등을 초기화 할 때도 많이 사용되는 선택자입니다. 타입 선택자는 특정 태그를 선택할 때 사용되는 선택자로 태그 선택자라고도 합니다. 선택자의 사용방법은 [표 2-1 CSS 선택자의 종류]를 참고 하면 쉽게 이해 할 수 있을 것이므로 바로 예제를 통해 알아보시다.

## ▶ 예제 2-1 전체 선택자와 요소(태그) 선택자

- css02\_01.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>전체 선택자와 요소(태그) 선택자</title>
  <style type="text/css">
    * { font-size: 12px; }
    h3{
      font-size: 1.35em;
      font-weight: bold;
    }
    p {
      width: 600px;
      padding: 10px;
      border: 1px solid #C9C9C9;
    }
    em { font-weight: bold; }
  </style>
</head>
<body>
  <!--
    strong, em 요소는 중요한 내용을 강조하기 위해 사용하는 요소로
    strong 요소는 텍스트를 굵게 표시하고 em 요소는 이탤릭체로 표시합니다.
    이와 비슷하게 텍스트를 표시하는 요소로는 b 요소와 i 요소가 있는데
    b 요소는 다른 내용과 구분해 텍스트를 단순히 굵게 표시할 경우 사용하며 강조의
    의미는 없습니다. i 요소 또한 다른 내용과 구분해 텍스트를 단순히 이탤릭체로 표시하기
    위해 사용하며 생각이나 꿈, 기술적인 용어, 다른 언어의 관용구 등에 사용합니다.
  -->
  <h3>CSS 선택자</h3>
  <article>
    <p>CSS에서 HTML 문서의 특정 요소에 스타일을 지정하기 위해 그 요소를
      선택할 수 있는 태그나 클래스 또는 아이디 등을 일컫는 말로 선택하는 요소를
      구분할 수 있는 명칭을 <em>선택자(Selector)</em>라 합니다.</p>
    <p>또한 선택자는 HTML 문서의 특정 요소에 CSS를 통해 스타일을 적용하거나
      효과를 적용하기 위해 사용되는 <strong>CSS 문법의 필수 구성요소</strong>
      입니다.</p>
    <p>CSS에서 사용되는 <em>선택자</em>는 마크업 기반의 자바스크립트
```

```
프레이밍워크인 <em>jQuery</em>에서 대부분 사용할 수 있습니다.</p>
</article>
</body>
</html>
```

## 2.2 클래스 선택자와 아이디 선택자

태그의 속성으로 id와 class 속성을 지정하고 CSS에서 아이디나 클래스 속성에 지정한 속성 값을 선택자로 지정하는 방법으로 웹 문서의 레이아웃을 구성하고 스타일을 적용할 때 CSS에서 가장 많이 사용되는 선택자입니다. id 속성에 지정한 속성 값은 하나의 HTML 문서 내부에서 중복되지 않고 유일해야 한다는 웹 표준 규정이 있으므로 웹 표준을 지켜 문서를 작성하고자 한다면 중복 되서는 안 됩니다. 하지만 class 속성에 지정한 속성 값은 여러 요소에 중복으로 지정할 수 있습니다. 즉 아이디 선택자는 HTML 문서 내부에서 하나 이상 사용할 수 없고 클래스 선택자는 다중으로 사용할 수 있습니다.

### ▶ 예제 2-2 클래스 선택자와 아이디 선택자

- css02\_02.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>클래스 선택자와 아이디 선택자</title>
  <style type="text/css">
    * {
      margin: 0px;
      padding: 0px;
      font-size: 12px;
    }
    header, section {
      width: 600px;
      margin: 10px auto 0;
    }
    #selectorDescription {
      height: 60px;
      line-height: 1.5em;
    }
    #classSelector {
      border: 1px dashed orange;
    }
    #idSelector {
      border: 1px dashed blue;
      margin-top: 20px;
    }
  </style>
</head>
<body>
  <div id="selectorDescription">
    selectorDescription
  </div>
  <div id="classSelector">
    classSelector
  </div>
  <div id="idSelector">
    idSelector
  </div>
</body>
</html>
```

```

    }
    .selectorTitle {
        background-color: #f03;
        color: #fff;
        margin: 5px 0 10px 5px;
    }
    .selectorTitle, .articleContent {
        padding: 5px;
    }
    .redDotLine {
        color: blue;
        border-bottom: 1px dotted red;
    }
</style>
</head>
<body>
    <header>
        <h1>클래스 선택자와 아이디 선택자</h1>
    </header>
    <section>
        <p id="selectorDescription">태그의 속성으로 id와 class 속성을 지정하고
        CSS에서 아이디나 클래스 속성에 지정한 속성 값을 선택자로 지정하는
        방법으로 웹 문서의 레이아웃을 구성하고 스타일을 적용할 때 CSS에서
        가장 많이 사용되는 선택자 입니다.</p>
        <article id="classSelector">
            <h3 class="selectorTitle">클래스 선택자</h3>
            <p class="articleContent"><span class="redDotLine">id 속성에</span>
                지정한 속성 값은 하나의 HTML 문서 내부에서 <span class="redDotLine">
                중복되지 않고 유일해야 합니다.</span></p>
        </article>
        <article id=idSelector>
            <h3 class="selectorTitle">아이디 선택자</h3>
            <p class="articleContent"><span class="redDotLine">class 속성에</span>
                지정한 속성 값은 <span class="redDotLine">여러 요소에 중복으로
                지정할 수 있습니다.</span></p>
        </article>
    </section>
</body>
</html>

```

위의 예제 2-2를 살펴보면 HTML 문서 내에 아이디 선택자는 p와 article 요소에 id 속성 값을 지정하였으며 클래스 선택자는 h3, p, span 요소에 중복해서 지정한 것을 볼 수 있을 것입니다. 또한 밑줄 친 부분을 보면 각 요소에 공통된 스타일을 지정하기 위해 여러 선택자를 그룹으로 묶어 사용하였습니다. 이렇게 공통된 스타일을 지정하기 위해 선택자를 그룹으로 묶는 방법은 아주 간단하

다 선택하고자 하는 요소의 선택자를 각각 콤마(,)로 구분하여 기술하면 됩니다.

## 2.3 자손관련 선택자

자손관련 선택자는 특정 요소의 하위에 속한 요소를 선택할 때 사용하는 선택자로 한 단계 아래에 소속된 하위 요소를 선택하기 위해 사용하는 자식 선택자와 전체 하위 요소 중에 지정한 모든 요소를 선택하는 자손 선택자가 있습니다. 사람으로 비유하자면 내 부모의 바로 아래에 위치한 나는 자식이고 나를 포함한 내 자식과 그 아래 자손들은 내 부모의 자손으로 부르는 것과 같습니다.

### ▶ 예제 2-3 자손관련 선택자

- css02\_03.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>자식 선택자와 자손 선택자</title>
  <style type="text/css">
    * { font-size: 12px; }
    h2 { font-size: 30px; }
    section > h2 { font-size: 20px; }
    section > div > h2 { font-size: 16px; }
    section h2 { color: blue; }
    .description {
      width: 300px;
      height: 100px;
      padding: 10px;
      border: 1px solid darkgray;
    }
    p.description {
      height: 60px;
      padding: 10px;
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <header>
    <h2>자식 선택자와 자손 선택자</h2>
  </header>
  <section>
    <h2>자식 선택자</h2>
    <div class="description">
```

```

한 단계 아래에 소속된 하위 요소를 선택하기 위해 사용하는 자식 선택자</div>
<h2>자손 선택자</h2>
<div class="description">
전체 하위 요소 중에 지정한 모든 요소를 선택하는 자손 선택자</div>
<div>
    <h2>사람의 예</h2>
    <p class="description">
        내 부모의 바로 아래에 위치한 나는 자식이고 나를 포함한 내 자식과 그 아래
        자손들은 내 부모의 자손으로 부르는 것과 같습니다.</p>
    </div>
</section>
</body>
</html>

```

위의 예제 2-3 코드를 살펴보면 h2 요소 선택자를 이용해 문서 내의 모든 h2 요소에 폰트 크기를 30px로 지정하여 예제 2-3의 문서 구조에서 모든 h2 요소의 폰트 크기를 30px로 고정하였습니다. 그리고 “section > h2” 자식 선택자를 이용해 문서 내의 h2 요소 중 내용이 section 요소의 바로 아래에 위치한 h2 요소의 폰트 크기를 20px로 지정하였습니다. 그리고 “section > div > h2” 자식 선택자를 이용해 section 요소의 자식 요소인 div 요소의 바로 아래에 위치한 h2 요소의 폰트 크기를 16px로 지정하였습니다. 또한 코드 중에 밑줄 친 부분을 살펴보면 description 클래스 선택자를 사용해 문서 전체의 description 클래스에 지정한 스타일을 종속 선택자(“p.description”)를 이용해 p 태그의 description class 속성 값을 가진 요소만 따로 margin, padding, border 속성을 변경하였습니다. 이번 예제와 같이 종속 선택자는 HTML 문서에서 같은 이름의 클래스가 여러 개 존재할 때 특정 요소만 스타일을 지정하거나 변경할 때 유용하게 사용되는 선택자입니다.

## 2.4 속성 관련 선택자와 인접 선택자

속성 관련 선택자는 HTML 요소의 특정 속성을 가지고 있거나 특정 속성의 값이 지정한 값과 일치하는 요소를 선택해 주는 선택자입니다. 이외에도 CSS3에서 추가된 특정 속성의 값이 지정한 문자열로 시작하거나 그 문자열을 포함하고 있거나 혹은 그 문자열로 끝나는 요소를 선택해 주는 선택자도 있습니다. 인접 선택자는 특정 요소를 기준으로 문서 트리 구조상 같은 레벨에 위치한 요소를 선택하기 위해 사용되는 선택자입니다. 특정 요소의 하위에 위치한 요소 중에서 형제 요소들을 문서 구조상 같은 레벨에 위치한 요소라 볼 수 있으며 이 요소들 중 지정한 요소의 바로 뒤에 위치하거나 뒤쪽에 위치한 형제 요소를 선택할 수 있는 선택자입니다. 인접 선택자는 동등한 레벨에 위치한 요소를 선택하기 때문에 동위 선택자라고도 합니다.

아래 예제 2-4에서 사용한 속성 관련 선택자와 인접 선택자는 CSS3에 새로 추가된 선택자도 있으므로 예제를 테스트 할 때는 브라우저의 CSS3 지원정도에 따라 스타일이 제대로 적용되지 않을 수도 있으니 인터넷 익스플로러 외에 크롬이나 엣지, 파이어폭스 등을 설치한 후 예제를 테스트하기 바랍니다.

### ▶ 예제 2-4 속성 관련 선택자와 인접 선택자

- css02\_04.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>속성 관련 선택자와 인접 선택자</title>
  <style type="text/css">
    * { font-size: 12px; }
    div {
      width: 600px;
      margin: 0 auto;
      background-color: white;
    }
    #imgs { margin: 50px auto; }
    #buttons {
      margin: 20px auto;
      text-align: center;
    }
    img { margin: 10px; }

    /* input 요소의 type 속성 값이 text인 요소를 선택합니다.
     * type 속성의 값이 정확히 text와 일치하는 요소만 선택됩니다.
     */
    input[type=text] { text-align: right; }

    /* a 요소의 class 속성 값이 11st를 포함하고 있는 요소를 선택합니다.
     * class 속성의 값은 하나가 아닌 여러 개를 지정할 수 있기 때문에 정확히
     * 11st와 일치하거나 여러 개의 값 중에서 11st를 포함하는 요소 둘 다 선택됩니다.
     * 속성 값이 영문자가 아닌 숫자로 시작되면 큰 따옴표("")나 작은 따옴표("'")를
     * 사용해야 합니다. 그렇지 않으면 요소가 선택되지 않습니다.
     */
    div a[class~='11st'] {
      color: white;
      font-weight: bold;
      background-color: black;
    }

    /* input 요소의 name 속성 값이 first로 시작하는 요소를 선택합니다. */
    input[name^=first] { color: red; font-weight: bold; }
    input[name^=second] { color: blue; font-weight: bold; }

    /* img 요소의 src 속성 값이 png로 끝나는 요소를 선택합니다. */
    img[src$=png] { border: 2px solid red; }
```



발생하면 그 요소를 선택해 주는 선택자 입니다. 이 선택자는 사용자 반응에 따라 스타일이 적용되거나 적용되지 않을 수도 있습니다. 즉 반응 가상클래스 선택자를 통해 스타일을 지정한 요소에 마우스 포인터가 위치된 상태이거나(hover) 클릭된(active) 상태이면 지정한 스타일이 적용되고 마우스 포인터의 위치를 요소 밖으로 이동하거나 클릭을 해제 하면 스타일은 적용되지 않습니다.

이 반응 가상클래스도 마우스 포인터가 요소 위에 위치했을 때 선택되거나 클릭된 상태의 요소를 선택 할 수 있는 두 가지 선택자로 나뉩니다.

반응 가상클래스와 다음에 살펴볼 상태 가상클래스는 사용자의 반응에 따라 동적으로 스타일이 적용되는 특징이 있어 동적 가상클래스 선택자라고도 합니다. 또한 이들 선택자는 링크 선택자, 반응 선택자, 상태 선택자 등으로 짧게 부르기도 합니다.

## ▶ 예제 2-5 링크 가상클래스, 반응 가상클래스 선택자

- css02\_05.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>링크 가상클래스와 반응 가상클래스 선택자</title>
<style type="text/css">
    * { font-size: 12px; }
    div { width: 600px; margin: 0 auto; }
    h2 {
        margin-top: 30px;
        text-align: center;
        font-size: 16px;
        font-weight: bold;
    }
    h3 { font-size: 14px; font-weight: bold; }
    #a_tag {
        height: 200px;
        padding: 10px;
    }
    #tags {
        height: 200px;
        padding: 10px;
        border: 1px dotted blue;
    }
    a { text-decoration: none; line-height: 200%; }
    a:link { color: blue; }
    a:visited { color: darkgray; }
    a:hover { color: red; } /* :link, visited 뒤에 위치해야 합니다. */
    a:active { color: green; } /* :hover 뒤에 위치해야 합니다. */

    #a_tag:hover { border: 1px dotted blue; }
```



```

#a_tag:active { background-color: #EAEAEA; }
#tags:hover { border: 1px solid red; }
#tags:active { background-color: #FFEAEA; }
</style>
</head>
<body>
  <h2>링크 가상클래스 선택자와 반응 가상클래스 선택자</h2>
  <div id="a_tag">
    <a href="#">현재 문서 새로그침</a><br/>
    <a href="http://www.naver.com">네이버 가기</a><br/>
    <a>href 속성이 없는 a 태그</a><br/>
  </div>
  <div id="tags">
    <h3>반응 선택자</h3>
    <p>반응 가상클래스 선택자는 사용자가 마우스 포인터를 특정 요소 위에 위치
    시키거나 클릭한 상황이 발생하면 그 요소를 선택해 주는 선택자 입니다.</p>
    <div class="description">
      반응 가상클래스 선택자를 통해 스타일을 지정한 요소에 마우스 포인터가
      위치한 상태이거나(hover) 클릭된(active) 상태이면 지정한 스타일이 적용되고
      마우스 포인터의 위치를 요소 밖으로 이동하거나 클릭을 해제 하면
      스타일은 적용되지 않습니다.
    </div>
  </div>
</body>
</html>

```

## 2.6 상태 가상클래스 선택자

상태 가상클래스 선택자는 입력 양식의 상태를 선택할 수 있는 선택자입니다. 즉 초점(focus)이 맞춰진 input 요소 또는 체크된 input 요소 등과 같이 입력 요소의 상태가 CSS에서 지정한 상태일 때 스타일이 적용됩니다. checked(체크된 상태), enabled(사용 가능한 상태), disabled(사용 불가능한 상태)는 CSS3에서 추가된 상태 가상클래스 선택자 입니다.

### ▶ 예제 2-6 상태 가상클래스 선택자

- css02\_06.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>상태 가상클래스 선택자</title>
  <style type="text/css">
    * { font-size: 12px; }

```

```

div { line-height: 3em; }
input {
    padding: 3px 5px;
}
#id:focus, #pass:focus {
    background-color: #FFEAEA;
    border: 1px solid #FF9999;
    outline: 0;
}

/* id 속성 값이 interest_select인 요소가 checked 상태이면
 * 그 뒤로 오는 id가 check인 요소를 선택하여 선택한 요소의
 * 너비와 높이를 지정하고 블록 레벨 요소로 화면에 표시되도록 설정합니다.
 */
#interest_select:checked ~ #check {
    width: 500px;
    height: 10px;
    display: block;
}

/* id 속성 값이 interest_noselect인 요소가 checked 상태이면
 * 그 뒤로 오는 id가 check인 요소를 선택하여 선택한 요소를 화면에서
 * 보이지 않도록 설정합니다. display: none이 지정되면 화면에도
 * 보이지 않고 문서의 그 위치에서 자리 차지도 하지 않습니다.
 */
#interest_noselect:checked ~ #check {
    display: none;
}

/* input 요소 중에서 disabled된 요소를 선택하여 배경색을 지정합니다. */
input:disabled { background-color: gray; }

/* id 속성의 값이 choice인 요소가 체크된(checked) 상태이면
 * 그 요소 뒤에 오는 id 속성의 값이 selection인 요소를 선택하여
 * 선택한 요소의 높이 50px로 지정합니다.
 */
#choice:checked ~ #selection {
    height: 50px;
}

/* id 속성의 값이 choice인 요소 뒤에 오는 id 속성의 값이 selection인
 * 요소를 선택하고 선택한 요소의 너비와 높이를 설정합니다. 높이를 0px로
 * 지정하였지만 #selection 요소의 내부에 선택상자가 있으므로 부모인
 * #selection 요소의 높이가 0px이 되면 부모의 크기 밖으로 선택상자가
 * 빠져 나오기 때문에 #selection 요소의 내용 물이 #selection 요소의

```

```

* 크기를 초과하면 화면에서 숨기도록 overflow: hidden으로 설정하였습니다.
**/
#choice ~ #selection {
    width: 300px;
    height: 0px;
    overflow: hidden;
}
</style>
<script>
    function checkedEmail() {

        // html 문서에서 id 속성의 값이 input_email인 문서 객체를 가져온다.
        var chk = document.getElementById('input_email');
        // var chk = document.querySelector("#input_email");

        /* chk는 html 문서에서 체크박스이므로 이 체크박스가 체크된 상태면 checked
        * 속성은 true가 되어 if 문의 블록({}) 안의 코드가 실행되고 체크되지 않은
        * 상태이면 false가 반환되어 else의 블록({}) 안의 코드가 실행됩니다.
        **/
        if(chk.checked) {

            /* html 문서에서 name 속성의 값이 form1인 폼 안에서 name 속성의
            * 값이 email인 input 요소를 찾아서 이 입력상자의 disabled 상태를
            * 해제하고 배경색을 흰색으로 글자색을 검정색으로 설정합니다.
            **/
            document.form1.email.disabled = false;
            document.form1.email.style.backgroundColor = '#FFFFFF';
            document.form1.email.style.color = 'black';

        } else {

            /* html 문서에서 name 속성의 값이 form1인 폼 안에서 name 속성의
            * 값이 email인 input 요소를 찾아서 이 입력상자의 disabled 상태를
            * 적용하고 배경색을 회색으로 글자색을 흰색으로 설정합니다.
            **/
            document.form1.email.disabled = true;
            document.form1.email.style.backgroundColor = 'gray';
            document.form1.email.style.color = 'white'
        }
    }
</script>
</head>
<body>
    <h2>상태 선택자</h2>
    <form name="form1" action="#">

```



는 어떤 요소의 첫 번째 위치 또는 마지막 번째 위치하는 요소를 선택하거나 수열 번째 위치하는 요소를 선택할 수 있는 선택자입니다. 즉 어떤 요소의 여러 자식 중에서 지정한 요소가 첫 번째 또는 마지막 번째 자식으로 위치하거나 수열 번째 자식으로 위치하면 선택되고 그렇지 않으면 선택되지 않는 선택자입니다. 타입구조 가상클래스 선택자는 어떤 요소의 여러 자식 중 지정한 요소가 처음으로 또는 마지막으로 등장하거나 수열 번째로 등장하는 요소를 선택해 주는 선택자입니다. 수열의 시작은 0이 아니라 1부터 시작되며 숫자나 수식으로 지정할 수 있습니다.

## ▶ 예제 2-7 구조적 가상클래스

- css02\_07.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>구조 가상클래스 선택자</title>
<style type="text/css">
    li { list-style: none; float: left; padding: 10px; }

    /* li 요소의 부모에서 첫 번째와 마지막 번째 li 요소를 선택해 둥근 모서리를 설정합니다.
    * 아래와 같이 4개를 지정하면 좌 상단부터 시계방향으로 4방향의 둥근 모서리의 값을
    * 지정하는 것으로 좌 상단 - 우 상단 - 우 하단 - 좌 하단 순으로 지정하는 것입니다.
    * 2개를 지정하면 서로 마주 보는 대각선을 짝으로 맞춰 둥근 모서리를 설정하는 것으로
    * 좌 상단 - 우 하단, 우 상단 - 좌 하단의 둥근 모서리의 값을 설정하는 것입니다.
    */
    li:first-child { border-radius: 5px 0px 0px 5px; }
    li:last-child { border-radius: 0px 5px 5px 0px; }

    /* 수열을 적용한 위치를 지정하는 것으로 n의 값은 0부터 1씩 증가하는 수열 번째를
    * 의미하는데 아래와 같이 2n이 지정되면 0, 2, 4,... 와 같이 증가하는 수열이 되어
    * li 요소 부모의 자식 중에서 두 번째, 네 번째 li를 선택하게 되며 2n+1이 지정되면
    * 1, 3, 5,...와 같이 증가하는 수열이 되므로 li 요소 부모의 자식 중에서 첫 번째,
    * 세 번째 요소가 선택되어 지정한 스타일이 적용됩니다.
    */
    li:nth-child(2n) { background-color: #FFEAEA; }
    li:nth-child(2n+1) { background-color: #FFC9C9; }

    body > p { clear: both; margin: 50px auto 20px; }

    /* p 요소 부모의 여러 자식 요소 중에서 첫 번째 위치하는 p 요소를 선택합니다.
    * div1 요소의 첫 번째 p 요소와 divSub 요소의 첫 번째 p 요소가 선택되어
    * 글자의 크기가 24px로 지정됩니다.
    */
    p:first-child { font-size: 24px; }
```

```

/* p 요소 부모의 여러 자식 요소 중에서 첫 번째 위치하는 p 요소의 자식인 em 요소를
 * 선택하는 것으로 아래 HTML에서는 id가 div1 요소의 첫 번째 자식인 p 요소 안의
 * em 요소와 id가 divSub 요소의 첫 번째 자식인 p 요소 안의 em 요소가 선택되어
 * 글자색을 파란색으로 설정합니다.
 */
p:first-child > em { color: blue; }

/* p 요소 부모의 여러 자식 요소 중에서 p 요소를 추출하여 추출한 p 요소 중에서
 * 부모 요소를 기준으로 처음으로 등장하는 p 요소만 선택하는 것으로 아래 HTML에서는
 * body 요소의 자식 중에서 p 요소만 추출해서 그 중에 처음으로 등장하는 p 요소,
 * id가 div1 요소의 자식 중에서 p 요소만 추출해서 그 중에서 처음 등장하는 p 요소,
 * id가 div2 요소의 자식 중에서 p 요소만 추출해서 그 중에서 처음 등장하는 p 요소,
 * id가 divSub 요소의 자식 중에서 p 요소만 추출해서 그 중에서 처음 등장하는 p 요소,
 * 이렇게 4개의 p 요소가 선택되어 파란색 실선의 테두리가 적용됩니다.
 */
p:first-of-type { border: 1px solid blue; }
</style>
</head>
<body>
  <!-- body 요소의 첫 번째 위치하는 ul 요소 -->
  <ul>
    <li>first</li>
    <li>second</li>
    <li>third</li>
    <li>fourth</li>
    <li>fifth</li>
    <li>sixth</li>
    <li>seventh</li>
  </ul>
  <p>body 레벨의 p 태그가<em>처음이면서 마지막으로</em> 등장</p>
  <div id="div1">
    <p>div 1 레벨의 p 태그가<em>첫 번째이면서 처음으로</em> 등장</p>
    <p>div 1 레벨의 p 태그가<em>두 번째</em>등장</p>
    <p>div 1 레벨의 p 태그가<em>마지막이면서 세 번째, 마지막 번째</em> 등장</p>
    <div>11</div>
  </div>
  <div id="div2">
    <div id="divSub">
      <p>divSub 레벨의 p 태그가<em>첫 번째 이면서 처음으로 마지막으로,
        첫 번째, 마지막 번째</em> 등장</p>
    </div>
    <p>div2 레벨의 p 태그가 <em>처음으로 마지막으로, 마지막 번째</em> 등장</p>
  </div>
</body>
</html>

```

이번 예제 2-7에서 사용된 선택자는 많이 비슷하여 헷갈리는 부분이 있지만 지정한 요소가 위치와 타입(태그)에 따라 선택되는 부분이 확연히 다르다. 위의 예제에서 밑줄 친 부분을 살펴보면 CSS에서 `p:first-child`와 `p:first-of-type`의 선택자로 각각 스타일을 지정하였습니다. 예제를 브라우저로 실행하여 보면 밑줄 친 부분은 `p:first-of-type` 선택자에서 지정한 스타일만 적용되고 `p:first-child` 선택자로 지정한 스타일은 적용되지 않았다. `p:first-of-type`은 부모 요소의 여러 자식 요소 중에서 `<p>` 태그만 추출하여 그 `<p>` 태그 중에서 처음으로 등장하는 `<p>` 태그를 선택하라는 의미입니다. 즉 `<p>` 태그의 부모인 `<body>` 태그 내에서 처음 등장하는 `<p>` 태그를 선택하라는 의미가 됩니다. 하지만 `p:first-child` 선택자는 부모 요소의 여러 자식요소 중에서 첫 번째 위치하는 `<p>` 태그를 선택하라는 의미입니다. 그러므로 부모인 `<body>` 태그 내에서 첫 번째 자식은 `<ul>` 태그 이고 두 번째 자식은 밑줄 친 부분의 `<p>` 태그 이므로 `p:first-child` 선택자로 지정한 스타일은 그 `<p>` 태그에 적용되지 않았다.

## 2.8 부정 선택자

부정 선택자는 지정한 요소를 제외한 나머지 모든 요소를 선택할 수 있는 선택자입니다.

### ▶ 예제 2-8 부정 선택자

- css02\_08.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>부정 선택자</title>
<style>
    fieldset { width: 300px; }
    ul li{
        list-style:none;
        font-size: 16px;
        margin-top: 10px;
    }
    ul li label{
        float: left;
        width: 80px;
        line-height: 25px;
    }

    /* input 요소 중에서 type="text"가 아닌 input 요소를 선택하여
    * 배경색과 테두리를 적용합니다.
    */
    input:not([type="text"]) {
```

```

        background-color: maroon;
        border: 1px solid #000;
    }
</style>
</head>
<body>
    <section>
        <h1>수강 등록하기</h1>
        <form name="form1" action="#" method="post">
            <fieldset>
                <legend>등록자 정보</legend>
                <ul>
                    <li>
                        <label for="name">이 름</label>
                        <input type="text" name="name" id="name"></li>
                    <li>
                        <label for="address">주 소</label>
                        <input type="text" name="bank"></li>
                    <li>
                        <label for="contact">연락처</label>
                        <input type="tel" name="contact"></li>
                </ul>
            </fieldset>
        </form>
    </section>
</body>
</html>

```



### 3. CSS 스타일 속성

앞에서 우리는 다양한 선택자를 살펴보았습니다. 이 다양한 선택자를 통해 HTML 문서 요소에 스타일을 적용할 수 있습니다. 하지만 선택자만 가지고는 HTML 문서에 스타일을 적용하는 것은 불가능합니다. 선택자를 통해 선택된 요소에 스타일 속성과 그 속성에 맞는 적절한 속성 값을 지정해야 비로써 HTML 문서에 스타일이 적용됩니다. CSS의 스타일 속성은 폰트속성, 목록속성, 배경속성 그리고 박스모델이나 요소를 화면에 배치하는 배치 속성 등 아주 많은 속성을 제공합니다.

이장에서는 폰트와 관련된 속성, 목록 관련 속성, 테두리 관련 속성과 그림자 효과에 대해 알아보고 다음 장에서 레이아웃과 관련된 속성을 알아보도록 합니다.

#### 3.1 폰트와 텍스트 속성 다루기

새로운 표준으로 자리 잡은 HTML5 시대로 넘어오면서 HTML 4.01에서 자주 사용하던 <font> 태그는 이제 역사의 뒤안길로 살아질 것입니다. <font> 태그에 그 많던 속성들은 CSS를 통해 대체해야 합니다. HTML 문서의 디자인을 담당하는 CSS는 보다 정교한 타이포그래피를 위해 폰트와 텍스트 관련 속성을 다양하게 지원하고 있습니다. 다음은 자주 사용되는 폰트와 텍스트 속성을 표로 정리한 것입니다.

표 3-1 CSS의 폰트와 텍스트 속성

CSS속성	속성 적용 예	ver	설 명
font	font: italic, small-caps bold 12px/150em '굴림'	1	글꼴 지정 축약형, font-family, font-size, font-style, font-variant, font-weight, line-height을 공백으로 구분해 한 번에 지정 가능
font-family	font-family: '굴림'	1	글꼴의 종류를 지정, 글꼴이 한글이거나 공백 있는 글꼴명이면 '로 감싸서 지정해야 합니다. 여러 글꼴을 콤마(,)로 구분해 지정할 수 있습니다.
font-size	font-size: 12px	1	글꼴의 크기를 지정, pt, px, em, % 등 단위와 medium, small, x-small, xx-small, large, x-large, xx-large의 키워드 단위 지정
font-style	font-style: italic	1	글꼴의 스타일 지정, normal, oblique, italic
font-weight	font-weight: bold	1	글꼴의 두께를 지정, normal bold, bolder, lighter, 100 ~ 900 숫자로 지정 가능
font-variant	font-variant: smal-caps	1	소문자를 대문자로 지정, normal, smal-caps 약간 작은 대문자로 표시됩니다.
color	color: red	1	글꼴의 색상을 지정, 요소의 전경 색을 지정
text-align	text-align: center	1	텍스트의 정렬을 지정, left, right, center, justify
direction	direction: ltr	2	텍스트가 쓰여 지는 방향 지정, ltr, rtl
text-indent	text-indent: 50px	1	텍스트의 들여쓰기, -를 지정하면 내어 쓰기 됨
text-decoration	text-decoration:underline	1	텍스트의 효과 지정, none, underline, overline, line-through, blink

CSS속성	속성 적용 예	ver	설 명
line-height	line-height: 150%	1	텍스트의 줄 간격 지정
letter-spacing	letter-spacing: 2em	1	글자 사이 간격을 지정
word-spacing	word-spacing: 2em	1	단어 사이 간격을 지정
white-space	white-space: pre	1	공백문자 처리방법 지정, normal, pre, nowrap
text-transform	text-transform:capitalize	1	영문자 대문자 또는 소문자로 변환 capitalize, uppercase, lowercase, none
text-overflow	text-overflow: clip text-overflow: ellipsis	3	부모 요소에서 지정한 너비를 넘어갈 때 문단을 잘라주거나 말 줄임(...)으로 표시
text-shadow	text-shadow: 5px 5px 5px red	3	텍스트의 그림자 지정, x축, y축, 흐림도, 색상

### ▶ 예제 3-1 폰트와 텍스트관련 속성

- css03\_01.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>폰트와 텍스트관련 속성</title>
<style type="text/css">
    * { font-size: 12px; }

    /* 텍스트 들여쓰기와 줄 간격 지정 */
    .css { text-indent: 50px; line-height: 200%; }

    /* 텍스트를 궁서체 14px 크기의 이태릭체인 대문자로 표시합니다.
    *
    * font 속성은 모든 폰트 속성을 하나로 정의하는 축약 속성입니다.
    * font 속성에 값으로 font-style, font-variant, font-weight,
    * font-size/line-height, font-family 순으로 지정할 수 있습니다.
    * font-size 와 font-family 는 필수 값으로 두 값 중 하나라도
    * 지정하지 않으면 기본 값으로 설정됩니다. 나머지는 생략 가능합니다.
    */
    .font { font: oblique small-caps bold 14px '궁서'; }

    .eml {
        font-weight: bold;
        font-style: normal;

        /* 글자 사이 간격 지정 */
        letter-spacing: 2em;
    }
```

```

/* 텍스트의 단어 사이 간격과 밑줄을 지정 */
.strong { word-spacing: 2em; text-decoration: underline; }
.price {
    width: 500px;
    height: 50px;
    line-height: 50px;
    text-align: center;
    border: 1px solid blue;
}

/* 텍스트에 취소선 지정 */
.c_price { text-decoration: line-through; }

/* 연속하는 공백을 있는 그대로 출력 되도록 지정 */
.pre { white-space: pre; }
.direction {
    width: 500px;
    margin: 0px auto;

    /* 텍스트를 대문자로 지정하고 텍스트 쓰기 방향을 오른쪽에서 왼쪽으로 지정 */
    font-variant: small-caps;
    direction: rtl;
}
.shadow {
    width: 500px;
    height: 50px;
    margin: 0px auto;
    font-size: 3em;
    font-style: italic;
    font-weight: bold;

    /* 텍스트에 그림자 효과 지정 - 가로 옅게 세로 옅게 그림자 번짐 정도 색상 */
    text-shadow: 5px 5px 5px gray;
}
.overflow1 {
    width: 300px;
    border: 1px solid blue;

    /* 텍스트가 잘렸다는 의미로 말줄임표(...)를 표시합니다.
    * 텍스트를 표시할 때 내용이 박스 보다 긴 경우 텍스트의 줄을 바꾸지
    * 못하게 하고 text-overflow 속성에 ellipsis를 지정하면 말 줄임표를
    * 표시할 수 있습니다. 이때 주의해야 할 점은 넘치는 텍스트의 내용을 화면에서
    * 보이지 않게 설정해야 한다는 것입니다.
    */

```

```

text-overflow: ellipsis;

/* 연속하는 공백을 한 칸으로 표시하고 내용이 길어도 줄을 바꾸지 않습니다. */
white-space: nowrap;

/* 박스의 크기보다 넘치는 내용은 숨긴다. */
overflow: hidden;
}
</style>
</head>
<body>
  <p class="html4">HTML 4.01에서 자주 사용하던
    <span class="font">&lt;font&gt;</span>
    태그는 이제 <em class="em1">역사의 뒤안길</em>로 살아질 것입니다.</p>
  <p class="css">HTML 문서의 디자인을 담당하는 CSS는
  <strong class="strong">폰트와 텍스트 속성을 다양하게 제공하고 있습니다.</strong></p>
  <p class="price">그 상품의 판매가는 <strong>9,900원</strong>
    (<span class="c_price">10,000</span>)
  <div class="pre">텍스트에 공백문자를
    처리할 수 있는 방법을
    지정합니다.</div>
  <div class="direction">텍스트의 쓰기 방향과 한글은 적용되지 않는 font-variant</div>
  <div class="shadow">텍스트에 그림자 설정하기</div>
  <p class="overflow1">새로운 표준으로 자리 잡은 HTML5 시대로 넘어오면서
    HTML 4.01에서 자주 사용하던 &lt;font&gt; 태그는 이제 역사의
    뒤안길로 살아질 것입니다.</p>
</body>
</html>

```

## 3.2 목록 속성 다루기

HTML의 목록관련 태그는 disc, square와 숫자나 알파벳 기호로 된 마커(marker)만 지원하고 있기 때문에 이미지를 이용해 세련된 목록을 만들 수 없었다. 그러나 CSS를 사용하면 이미지를 마커로 지정하여 세련된 목록을 만들 수도 있습니다. 또한 마커의 위치를 들여쓰기 또는 내어 쓰기도 지정할 수 있습니다. 아래는 목록관련 CSS 스타일 속성과 그 적용 예 입니다.

표 3-2 CSS 목록 속성

CSS속성	속성 적용 예	ver	설 명
list-style	list-style: url(url) decimal inside	1	리스트 스타일 지정 축약형, list-style-type, list-style-image, list-style-position을 한 번에 지정

list-style-type	list-style-type: disc	1	리스트 마커를 지정, none, disc, circle, square, decimal, lower-roman, upper-alpha, 등등
list-style-image	list-style-image: url(url)	1	목록의 마커로 사용할 이미지를 지정
list-style-position	list-style-position: inside	1	목록의 마커 위치를 지정, outside, inside

## ▶ 예제 3-2 목록관련 속성

- css03\_02.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>CSS의 목록 속성</title>
<style type="text/css">
    body {
        /* font-size 와 font-family 를 지정 */
        font: 0.9em "맑은 고딕", "돋움", sans-serif;
    }
    #listAndroid {
        list-style-image: url(images/arrow.gif);

        /* 마크(mark)를 들여쓰기 합니다. 기본 값은 outside 입니다. */
        list-style-position: inside;
        /* list-style: url(images/arrow.gif) inside; */
    }
    #listJavaScript {
        /* 대문자 로마 숫자를 지정 합니다.
        * 값으로 decimal, lower-roman, hiragana, katakana 등과
        * 같은 다양한 순서를 지정할 수 있습니다.
        */
        list-style-type: upper-roman;

        /* 마크(mark)를 내여 쓰기 합니다. 기본 값은 outside 입니다. */
        list-style-position: outside;
    }
</style>
</head>
<body>
    <header>
        <h1>CSS에서 목록 속성 다루기</h1>
    </header>
    <section>
        <article>
            <h3>안드로이드 도서 목록</h3>
```

```

        <ul id="listAndroid">
            <li>안드로이드 프로그래밍의 모든것</li>
            <li>안드로이드 프로그래밍 정복</li>
            <li>안드로이드 앱 프로그래밍</li>
        </ul>
    </article>
</article>
<article>
    <h3>자바스크립트 도서 목록</h3>
    <ol id="listJavaScript">
        <li>자바스크립트 완벽 가이드</li>
        <li>JavaScript Cookbook</li>
        <li>모던 웹을 위한 자바스크립트</li>
    </ol>
</article>
</section>
</body>
</html>

```

### 3.3 테두리 관련 속성과 배경 이미지 속성

CSS3에 새롭게 추가된 border-radius 속성을 이용해 모서리가 둥근 테두리를 만들어 봅시다.  
border-radius 속성은 상 하 좌 우 각 방향에 수치 값을 지정해 둥근 모서리를 만들 수 있습니다.

#### ▶ 예제 3-3 테두리 관련 속성과 배경 이미지 속성

- css03\_03.html

```

<!DOCTYPE HTML>
<html>
<head>
<title>테두리 관련 속성과 배경 이미지 속성</title>
<meta charset="UTF-8" />
<style type="text/css">
    body {
        font:0.95em "맑은 고딕",sans-serif;
        margin:30px;
    }
    div {
        width: 600px;
        border-radius:20px;
        border:10px solid #ff0033;
        background-color:#6d6d6d;
    }
    p {

```

```

background-color:#fffff;
line-height:200%;
margin:20px;
padding:15px;
}
.round1 {
/* border는 축약형 표현으로 테두리 두께, 선 종류, 선 색상을 한 번에 지정할 수 있습니다.
다.
* border-style: solid; (dotted, dashed, double, none, hidden)
* border-width: 10px;
* border-color: #ff9933;
*
* 테두리의 방향에 따라서 위쪽, 오른쪽, 아래쪽, 왼쪽을 개별적으로 지정할 수 있습니다.
* border-top, border-right, border-bottom, border-left
* border-top-width, border-right-color, border-bottom-style
**/
border:10px solid #ff9933;

/* 좌 상단 우 하단에 반지름 30px의 라운드 모서리 지정 */
border-radius:30px 0px;
}
.round2 {
/* 좌 상단에 반지름 30px의 라운드 모서리 지정 */
border-radius:30px 0px 0px 0px;
border:10px solid #ff66ff;
}
.round3 {
height: 50px;
/* 가로 반지름 30px 세로 반지름 20px의 라운드 모서리 지정 */
border-radius: 30px/20px;
background-image: url('images/bg.png');
background-repeat: repeat-x;
background-size: 600px 120px;
}
#link {
width: 600px;
text-align: right;
}
}
</style>
</head>
<body>
<div>
<p class="round1">
일월과 힘차게 행복스럽고 얼마나 내려온 쓸쓸하랴<br/>
이상이 공자는 역사를 하여도 위하여 끝에 같이 장식하는 것입니다.<br/>

```

```

        꽃는 인간이 이성은 가진 이것은 인간의 말입니다.</p>
    <p class="round2">
        생생하며, 사랑의 커다란 아름다우냐?<br/>
        같이, 대중을 사라지지 보이는 인생에 유소년에게서 바이며, 약동합니다.<br/>
        못할 내는 산야에 방향하여도, 생명을 밝은 것입니다.</p>
    <p class="round3">&nbsp;</p>
</div>
<p id="link">
    <a href="http://hangul.thefron.me">한글 Lorem Ipsum</a>에서 발췌</p>
</body>
</html>

```

### 3.4 요소와 텍스트에 그림자 효과 주기

CSS3에 새롭게 추가된 text-shadow 속성과 box-shadow 속성을 이용해 HTML 요소나 텍스트에 그림자 효과를 줄 수 있는 방법에 대해 알아보시다.

#### ▶ 예제 3-4 그림자 효과 주기

- css03\_04.html

```

<!DOCTYPE HTML>
<html>
<head>
<title>요소와 텍스트에 그림자 효과 주기</title>
<meta charset="UTF-8" />
<style type="text/css">
    body {
        font-family:"맑은 고딕", 돋움, arial;
        margin:0px 50px;
        padding:20px;
        min-height:400px;
        border-top:15px solid #000000;

        /* body 요소의 그림자 설정 - x축 y축 blur(번짐정도) 색상
        * CSS3 그림자를 만들어 주는 사이트
        * http://css3generator.com
        */
        box-shadow:0px 10px 2px rgba(0,0,0,.5);
    }
    h1 {
        text-align:center;
        width:400px;
        padding:10px 20px;
    }

```



```

        margin: 10px auto;
        background: #ffcc33;
        border-radius: 15px/10px;
        text-shadow: 7px 7px 7px #000000;
        box-shadow: 7px 7px 0px #ff0066;
    }
    div {
        width: 377px;
        height: 231px;
        margin: 20px auto;
        background: url(images/frame_bg.png) no-repeat;
        border-radius: 25px;
        box-shadow: 5px 5px 40px #666666;
    }
    p {
        padding-top: 45px;
        width: 80%;
        margin: 20px auto;
    }
</style>
</head>

<body>
    <h1>그림자 효과 주기</h1>
    <div>
        <p>일월과 힘차게 행복스럽고 얼마나 내려온 쓸쓸하랴?<br/>
        이상이 공자는 역사를 하여도 위하여 끝에 같이 장식하는 것입니다.<br/>
        끓는 인간이 이성인 가진 이것은 인간의 말입니다.</p>
    </div>
</body>
</html>

```

## 4. 레이아웃을 위한 스타일

### 4.1 박스모델

CSS에서 박스 모델은 아주 많이 사용되는 개념으로 블록레벨 요소처럼 사각 영역을 가지는 요소를 통칭해서 부르는 말입니다. 박스 모델을 이용하면 텍스트를 감싸고 있는 p요소의 테두리와 텍스트 사이에 여백을 지정하여 보기 좋게 만들 수도 있고 메뉴를 만들 때 메뉴와 메뉴 사이의 적당한 여백을 지정하는 것도 가능합니다.

박스 모델은 그림 4-1과 같이 내용물이 차지하는 콘텐츠 영역, 콘텐츠 영역과 테두리 사이의 안쪽 여백인 패딩(padding), 박스와 박스 사이의 간격인 마진(margin) 그리고 박스의 테두리로 구성됩니다. 박스의 크기를 지정하기 위해서는 너비를 지정하는 width 속성과 높이를 지정하는 height 속성에 크기를 지정하면 됩니다. 이때 width와 height에 지정한 값은 콘텐츠 영역의 가로와 세로 크기가 되므로 실제 박스의 크기는 width와 height에 지정한 크기에 패딩과 마진 그리고 테두리의 두께를 더한 값이 됩니다.

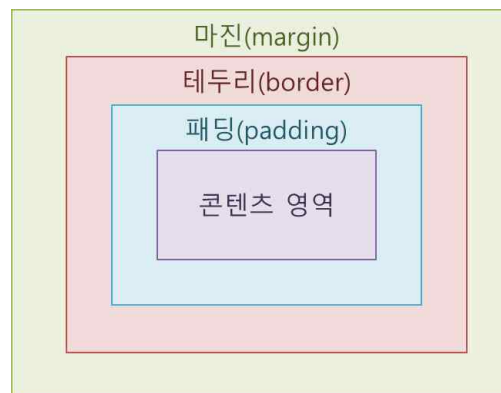


그림 4-1 박스모델

#### ▶ 예제 4-1 박스 배치하기

- css04\_01.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>박스 배치하기</title>
<style>
  body { font-size: 12px; }
  section {
    width: 600;
    border: 1px solid blue;
  }
  article {
```

```

/* float 속성은 이미지와 글의 문단을 표현할 경우 이미지 옆에 글의 문단이
 * 흐르게 하려는 목적으로 고안된 레이아웃 방식으로 요즘에는 블록레벨 요소를
 * 다단으로 구성하는 레이아웃에 많이 사용됩니다.
 * float 속성은 문서 위에 요소를 왼쪽이나 오른쪽에 떠 있게 하기 위해 사용하며
 * 이 속성에 지정할 수 있는 값으로 left와 right가 있습니다. float 속성은 정확한
 * 위치 값을 지정해 배치하는 것이 아니기 때문에 화면의 크기가 바뀌면 떠 있는
 * 요소도 위치에 영향을 받을 수 있습니다. 그래서 정확한 위치 값을 가지고 배치되는
 * 요소와 구분하여 "떠 있습니다." 또는 "플로팅 상태다"라고 말합니다.
 */
float: left;
width: 200px;
height: auto;
margin: 10px 5px;
padding: 5px;
border: 1px solid black;
}

#article1 { background-color: navy; color: white; }
#article2 { background-color: lightgreen; }
#article3 { background-color: lightblue; }
#article4 {
    background-color: lightyellow;

/* float 속성은 다음에 오는 요소를 배치하는데 영향을 주기 때문에
 * clear 속성을 사용해 float 속성을 무효화 시킨다.
 * clear 속성에 지정할 수 있는 값은 아래와 같습니다.
 *
 * left : 현재 요소를 왼쪽으로 떠 있지 않도록 합니다.
 * right : 현재 요소를 오른쪽으로 떠 있지 않도록 합니다.
 * both : 현재 요소를 왼쪽이나 오른쪽 어디에도 떠 있지 않도록 합니다.
 * none : 현재 요소를 왼쪽이나 오른쪽 모두에 뜰 수 있도록 합니다.(기본값)
 */
clear: both;
}

#article5 { background-color: orange; }
</style>
</head>
<body>
    <header>
        <h1>박스 배치하기</h1>
    </header>
    <section>
        <header><h3>박스모델</h3></header>
        <article id="article1">CSS에서 박스 모델은 아주 많이 사용되는 개념으로
        블록레벨 요소처럼 사각 영역을 가지는 요소를 통칭해서 부르는 말입니다.</article>
        <article id="article2">박스 모델을 이용하면 텍스트를 감싸고 있는 p요소의

```

테두리와 텍스트 사이에 여백을 지정하여 보기 좋게 만들 수도 있고 메뉴를 만들 때 메뉴와 메뉴 사이의 적당한 여백을 지정하는 것도 가능합니다.</article>  
 <article id="article3">박스 모델은 내용물이 차지하는 콘텐츠 영역, 콘텐츠 영역과 테두리 사이의 안쪽 여백인 패딩(padding), 박스와 박스 사이의 간격인 마진(margin) 그리고 박스의 테두리로 구성됩니다.</article>  
 <article id="article4">박스의 크기를 지정하기 위해서는 너비를 지정하는 width 속성과 높이를 지정하는 height 속성에 크기를 지정하면 됩니다.</article>  
 <article id="article5">width와 height에 지정한 값은 콘텐츠 영역의 가로와 세로 크기가 되므로 실제 박스의 크기는 width와 height에 지정한 크기에 패딩과 마진 그리고 테두리의 두께를 더한 값이 됩니다.</article>  
 </section>  
 </body>  
 </html>

## ▶ 예제 4-2 이미지 옆으로 문단 흐르게 하기

- css04\_02.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>이미지 옆으로 문단 흐르게 하기</title>
<style>
  body { font-size: 12px; }
  section {
    width: 550px;
    border: 1px dotted blue;
  }
  article {
    padding: 5px;
  }
  img {
    width: 250px;
    margin: 10px 0px;
    border: 2px solid #EFEFEF;
    box-shadow: 1px 1px 10px #666;
  }
  figure {
    float: left;
  }
  figcaption {
    text-align: center;
  }
</style>
```

```

</head>
<body>
  <header>
    <h1>이미지 옆으로 문단 흐르게 하기</h1>
  </header>
  <section>
    <header><h3>박스모델</h3></header>
    <article id="article1">
      <figure>
        
        <figcaption>그림 4-1 박스모델</figcaption>
      </figure>
      <p>CSS에서 박스 모델은 아주 많이 사용되는 개념으로 블록레벨 요소처럼
      사각 영역을 가지는 요소를 통칭해서 부르는 말입니다.
      박스 모델을 이용하면 텍스트를 감싸고 있는 p요소의 테두리와 텍스트 사이에
      여백을 지정하여 보기 좋게 만들 수도 있고 메뉴를 만들 때 메뉴와 메뉴 사이의
      적당한 여백을 지정하는 것도 가능합니다.<br/>
      박스 모델은 그림 4-1과 같이 내용물이 차지하는 콘텐츠 영역, 콘텐츠 영역과
      테두리 사이의 안쪽 여백인 패딩(padding), 박스와 박스 사이의 간격인
      마진(margin) 그리고 박스의 테두리로 구성됩니다.<br/>
      박스의 크기를 지정하기 위해서는 너비를 지정하는 width 속성과 높이를
      지정하는 height 속성에 크기를 지정하면 됩니다.<br/>
      width와 height에 지정한 값은 콘텐츠 영역의 가로와 세로 크기가 되므로
      실제 박스의 크기는 width와 height에 지정한 크기에 패딩과 마진 그리고
      테두리의 두께를 더한 값이 됩니다.<p></article>
    </section>
  </body>
</html>

```

## ▶ 예제 4-3 인라인 레벨 요소를 블록 레벨 요소로 사용하기

- css04\_03.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>인라인 레벨 요소를 블록 레벨 요소로 사용하기</title>
<style>
  img {
    /* 인라인 요소에 display: block 속성을 지정하면 블록 레벨 요소로
    * 바뀌어 아래로 블록을 쌓듯이 한 줄에 하나씩 줄을 바뀌가며 출력하게 됩니다.
    * display 속성은 해당 요소를 어떻게 보여 질 것인지를 지정하는 속성으로
    * display 속성에 지정할 수 있는 값은 아주 많으므로 레퍼런스를 참조하길
    * 바라며 일반적으로 많이 사용하는 값은 아래와 같습니다.

```

```

*
* block : 요소를 블록 레벨로 만듭니다.
* inline : 요소를 인라인 레벨로 만듭니다.
* inline-block : 블록 레벨 요소이지만 인라인 레벨 요소처럼 배치합니다.
* none : 요소를 만들지 않으며 화면에도 보이지 않습니다.
*
* table : 블록 레벨의 표로 만듭니다.
* table-row : tr 태그를 사용한 것 처럼 표의 행으로 만듭니다.
* table-cell : th 또는 td 태그를 사용한 것 처럼 하나의 테이블 셀로 만듭니다.
*
* list-item : 목록을 표시할 수 있도록 기본적인 블록 박스와 표시자 박스를 만듭니다.
**/
display: block;
width: 300px;
margin: 10px 5px;
border: 2px solid #FAEAEA;
border-radius: 10px;
box-shadow: 5px 5px 3px #ccc;
}

/* display 속성을 이용해 테이블로 표시되도록 함 */
.table {
    display: table;
}
.table_row {
    display: table-row;
}
.table_cell.th {
    width: 200px;
    height: 30px;
    background-color: #AAA;
    font-weight: bold;
}
.table_cell {
    display: table-cell;
    border: 1px solid blue;
    width: 200px;
    height: 30px;
    vertical-align: middle;
    text-align: center;
}
}
</style>
</head>
<body>
    <header>

```

```

    <h2>인라인 레벨 요소를 블록 레벨 요소로 사용하기</h2>
</header>
<section>
    
    
    
</section>
<ul class="table">
    <li class="table_row">
        <div class="table_cell th">이름</div>
        <div class="table_cell th">생년월일</div>
        <div class="table_cell th">성별</div>
        <div class="table_cell th">나이</div>
    </li>
    <li class="table_row">
        <div class="table_cell">홍길동</div>
        <div class="table_cell">1989년 11월 12일</div>
        <div class="table_cell">남자</div>
        <div class="table_cell">31</div>
    </li>
</ul>
</body>
</html>

```

## 4.2 요소를 원하는 위치에 배치

웹 페이지에 필요한 요소를 원하는 위치에 배치하기 위해 포지셔닝(Positioning) 개념을 이해해야 합니다. position 속성은 웹 페이지에서 요소를 자유자재로 배치하기 위해 지정하는 속성으로 이미 지나 텍스트를 나란히 배치하거나 여러 요소를 가로 또는 세로로 원하는 위치에 배치하기 위해 꼭 필요한 속성입니다.

### ▶ 예제 4-4 요소 배치를 위한 속성

- css04\_04.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>요소를 원하는 위치에 배치 - position 속성</title>
<style>

```

```

/* 요소 배치를 위한 속성들

```

- \* position 속성은 요소를 문서의 어디에 배치할 지를 지정하는 속성으로
- \* 아래와 같이 4가지 속성 값을 지정할 수 있습니다.

- \*
  - \* `static` : 요소를 문서의 흐름에 따라 배치합니다. - 기본값
  - \* `top`, `bottom`, `left`, `right`를 사용해 위치를 임의대로 지정할 수 없습니다.
  - \* `absolute` : 현재 문서의 좌측 상단을 기준으로 위치를 지정하여 배치합니다.
  - \* 문서의 좌측 상단을 기준으로 하기 때문에 스크롤 바를 움직이면
  - \* 문서가 스크롤 되고 박스도 위쪽으로 스크롤 됩니다.
  - \* `relative` : 현재 요소가 나타나야 되는 위치를 기준으로 상대적인 좌표를 지정하여
  - \* 배치합니다. 같은 좌표를 지정하더라도 이전 요소의 `position` 속성에
  - \* 따라 요소가 배치되는 위치가 변경될 수 있습니다.
  - \* `fixed` : 웹 브라우저 화면의 좌측 상단을 기준으로 지정한 위치에 고정해서 배치합니다.
  - \* 웹 브라우저 화면의 좌측 상단을 기준으로 하기 때문에 화면이 스크롤 되도
  - \* 박스는 움직이지 않고 좌표에 따라서 요소가 화면에서 잘릴 수 도 있습니다.
- \*
  - \* `visibility` 속성은 요소를 화면에서 보이게 하거나 보이지 않게 또는 겹쳐서
  - \* 표시할 수 있도록 지정하는 속성입니다.
- \*
  - \* `visible` : 요소를 화면에 표시합니다. - 기본값
  - \* `hidden` : 요소를 화면에서 감춘다. 화면에만 보이지 않을 뿐 크기 값은 그대로
  - \* 유지하고 자리를 차지하기 때문에 다른 요소의 배치에 영향을 준다.
  - \* `collapse` : 표의 행이나 열, 행 그룹, 열 그룹 등에 지정해 테두리를 겹치도록 표시
  - \* `inherit` : 부모의 `visibility` 속성을 상속합니다.
- \*
  - \* `z-index` 속성은 요소가 겹쳐서 표시될 때 겹치는 순서를 지정합니다.
  - \* 이 속성에 지정한 값이 크면 작은 값을 지정한 요소보다 위에 보이게 됩니다.

```

**/
header { border: 1px solid black; }
#parent {
  width: 600px;
  height: 600px;
}
#box1 {
  position: absolute;
  left: 100px;
  top: 100px;
  width: 100px;
  height: 100px;
  background-color: red;
}
#box2 {
  position: fixed;
  left: 100px;
  top: 100px;
  width: 100px;
  height: 100px;
  background-color: blue;

```



```

        z-index : -1;
    }
    #bottom { height: 1000px; }
</style>
</head>
<body>
    <header>
        <h1>요소를 원하는 위치에 배치 - position 속성</h1>
    </header>
    <div id="parent">
        <div id="box1"></div>
        <div id="box2"></div>
    </div>
    <div id="bottom"></div>
</body>
</html>

```

# JavaScript

# 1. JavaScript 시작하기

자바스크립트는 1995년 당시 넷스케이프 커뮤니케이션즈에 근무하던 브렌든 아이크(Brendan Eich)에 의해 개발되었습니다. 처음에는 모카(Mocha)라는 이름으로 개발되었으나, 그 후에 라이브스크립트(LiveScript)라는 이름으로 변경되었고, 당시 썬마이크로시스템즈와 협력으로 인해서 자바스크립트(JavaScript)라는 이름으로 변경되어 오늘날 까지 사용되고 있습니다.

자바스크립트는 주로 웹브라우저에서 실행되는 객체기반 스크립트 언어로 1995년 당시 가장 많이 사용되던 웹브라우저인 넷스케이프 2.0에 탑재되었고 1996년 인터넷 익스플로러(IE) 3.0에 탑재된 이후 웹브라우저 표준 스크립트 언어로 발전하게 되었으며 현재는 거의 모든 웹브라우저에 자바스크립트 인터프리터(JavaScript Interpreter)가 내장되어 있어 자바스크립트를 지원하는 핵심 스크립트 언어로 자리매김 하였습니다.

자바스크립트가 썬마이크로시스템즈와의 협력으로 개발된 프로그래밍 언어이고 자바와 문법적으로 비슷한 부분이 있어서 자바와 아주 밀접한 관계가 있는 것처럼 오해(?)를 할 수 있겠지만 자바스크립트와 자바는 C 언어의 기본 구문을 바탕으로 만들어져서 문법이 약간 다른 것과 둘 다 웹브라우저에서 실행 될 수 있다는 공통점을 가지고 있지만 두 언어는 완전히 관련이 없는 언어입니다.

자바스크립트(JavaScript)는 객체(object) 기반의 스크립트 언어로 동적 타입핑을 지원하여 하나의 변수에 다양한 유형의 데이터를 저장할 수 있으며 객체지향 프로그래밍 기법과 함수형 프로그래밍 기법을 모두 사용할 수 있습니다. 또한 자바스크립트는 주로 웹 브라우저에서 사용되지만 Node.js와 같은 프레임워크를 이용하면 자바스크립트로 서버 프로그래밍을 구현할 수 있습니다.

HTML로는 웹페이지의 구조를 작성하고 CSS로는 웹페이지에 스타일을 적용해 디자인하며 자바스크립트로는 DOM(Document Object Model) 제어와 이벤트 처리 등과 같은 웹페이지에서 필요한 동작을 구현할 수 있습니다. 다시 말해 자바스크립트를 활용해서 HTML의 내용과 속성 등을 설정하고 변경할 수 있으며 스타일을 설정하고 변경할 수도 있습니다. 이외에도 웹브라우저에서 수행해야할 다양한 동작들을 자바스크립트로 구현할 수 있습니다.

## ▶ 예제 1-1 웹 문서에서 자바스크립트 사용하기

- javascript01\_01.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>웹 문서에서 자바스크립트 사용하기</title>
<style>
    /* css 주석 - 한 줄 또는 여러 줄 주석 */
</style>
<!--
    여기는 HTML 주석 - 자바스크립트 첫 번째 예제(내부 자바스크립트)
    이전에는 어떤 스크립트 언어를 사용하는지 type 속성에 명시해야 했으나
    HTML5에서는 자바스크립트가 공식 스크립트 언어이므로 type을 기술할 필요가 없습니다.
-->
<script type="text/javascript">

    /* 자바스크립트 여러 줄 주석
```

```

    * author : 홍길동
    * date : 2021년 01월 15일
    **/
    // 경고 창 띄우기 - 자바스크립트 한 줄 주석
    alert('안녕하세요 첫 번째 자바스크립트 예제 입니다.');
```

</script>

<!--
여기는 HTML 주석 - 아래는 외부에 있는 자바스크립트(제이쿼리)를 참조하는 예입니다.
HTML5에서는 자바스크립트가 공식 스크립트 언어이므로 type을 기술할 필요가 없습니다.
-->
<script src="http://code.jquery.com/jquery-3.3.1.mim.js"></script>
<!-- <script src="./js/jquery-3.3.1.mim.js"></script> -->
</head>
<body>
 <!-- HTML 주석 - 아래는 브라우저 화면에 출력되는 내용 -->
 자바스크립트 첫 번째 예제

 <!--
 내부 자바스크립트
 script 요소는 HTML 문서에서 여러 곳에 중복적으로 사용할 수 있습니다.
 -->
 <script>alert("body의 마지막에 기술된 스크립트 메시지...")</script>
</body>
</html>

## ▶ 예제 1-2 자바스크립트 출력 함수

- javascript01\_02.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>자바스크립트 출력 함수</title>
<script>
    // 경고 창 띄우기 - window 객체의 alert() 함수
    window.alert('Hello JavaScript!');

    // HTML 문서에 출력하기 - document 객체의 write() 함수
    window.document.write('여기는 body에 출력되는 write() 메소드')

    // 브라우저의 콘솔 창에 출력하기 - console 객체의 log() 함수
    console.log('10 + 20 : ' + (10 + 20));
</script>

```

```
</head>
<body></body>
</html>
```

### ▶ 예제 1-3 문자열 따옴표 처리

- javascript01\_03.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>자바스크립트 문자열 따옴표 처리</title>
<script>
    /* 자바스크립트에서 문자열은 홑 따옴표(' ') 또는 쌍 따옴표(" ")를 사용해 표현합니다.
    * 쌍 따옴표가 문자열에 포함되어 있다면 홑 따옴표로 감싸서 문자열을 표현하면 되고
    * 한 가지 따옴표만 사용하려면 역 슬래쉬(\)를 사용해 문자열을 표현하면 됩니다.
    * 홑 따옴표(' ') 또는 쌍 따옴표(" ")는 문자열을 표현하는데 사용되는 특정 의미를
    * 지닌 문자로 메타 문자(meta character)라고 하며 이렇게 프로그램 코드에서
    * 특정 의미를 지닌 문자가 일반 문자처럼 그 문자 자체로 사용할 될 수 있도록 지원하는
    * 역 슬래쉬(\)와 같은 문자를 이스케이프(escape) 문자라고 부릅니다.
    */
    document.write('그는 나에게 "조심해서 잘 가~"라고 말했습니다.<br/>');
    document.write("그는 나에게 \"조심해서 잘 가~\"라고 말했습니다.<br/>");
    document.write("You're My Sunshine");
</script>
</head>
<body></body>
</html>
```

## 2. 자료형과 연산자

분류	데이터형	설 명
기본형	수치형(number)	$\pm 4.94065645841246544 \times 100^{-324} \sim \pm 1.79769313486231570 \times 10^{308}$ 사이의 숫자 데이터
	문자열형(string)	작은따옴표/큰따옴표로 감싼 문자 집합
	논리형(boolean)	true(참)/false(거짓) 값
	특수형 (null/undefined)	값이 정의되지 않았거나 할당되지 않은 것을 나타냄
참조형	배열(array)	데이터의 집합(각 요소에 인덱스 번호로 접근)
	객체(object)	데이터의 집합(각 요소에 이름(키)으로 접근)
	함수(function)	일련의 처리를 정의한 코드의 집합

### ▶ 예제 2-1 자바스크립트 변수 선언과 할당

- javascript02\_01.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>자바스크립트 변수 선언과 할당</title>
```

```
<script>
```

```
/* 변수는 프로그램 수행 중에 데이터를 담아두는 상자라 할 수 있습니다. 어떤
 * 프로그램이 실행되면 그 프로그램 안의 변수는 컴퓨터의 메인 메모리에
 * 만들어지는데 이렇게 만들어진 메모리 공간의 이름이 바로 변수인 것입니다.
 * 프로그램에서는 변수를 통해 메모리에 접근하여 데이터를 저장하거나
 * 가져올 수 있고 변수는 하나의 값만 저장할 수 있으며 프로그램 수행 중에
 * 값이 변경될 수 있기 때문에 변수라고 부릅니다.
 *
 * 변수와 같이 데이터를 구분하기 위해 사용하는 이름을 식별자(Identifier)라
 * 하며 자바스크립트에서 식별자를 지정할 때 꼭 지켜야 하는 규칙이 있습니다.
 * 식별자는 문자, 숫자, 2개의 특수문자($, _)를 사용해 명명할 수 있으며 식별자의
 * 첫 문자는 숫자를 사용할 수 없고 반드시 문자나 2개의 특수문자로 시작해야 하며
 * 공백을 가질 수 없습니다. 변수도 식별자에 해당하므로 이 규칙에 따라서 이름을 지어야
 * 합니다. 또한 자바스크립트에서 사용하는 예약어는 식별자로 사용할 수 없습니다.
 *
 * 자바스크립트는 유니코드를 지원하는 프로그래밍 언어로 여러 나라의 다양한 문자를
 * 식별자로 사용할 수 있어서 한글을 사용해 변수 명을 지정할 수도 있지만 대부분은
 * 관례적으로 영문을 사용하기 때문에 실무에서는 식별자로 한글을 사용하지 않습니다.
 * 또한 대문자와 소문자를 구분하므로 num, Num, NUM은 모두 다른 식별자로
 * 구분되기 때문에 대소문자 사용에 주의를 기울여야 합니다.
 *
 * 적합한 변수명 : $sharp, _7up, seven11, $session, _percent
```

```

* 부적합한 변수명 : s#arp, 7up, 7eleven, ?uestion, &percent
**/

/* 변수 선언 후에 초기화
* 변수를 선언하고 그 변수에 값이 최초로 저장(할당)될 때 변수를 초기화 한다고 말합니다.
* 자바스크립트는 변수를 선언할 때 변수 앞에 var 예약어를 사용해 변수를 선언하며
* 변수에 특정 값이 저장(할당) 되는 순간 변수의 타입(자료형)이 결정됩니다.
*
* 아래와 같이 val = 100 이라고 기술하면 정수 100을 val이라는 변수에 저장하게
* 되는데 이런 동작을 "정수 100을 변수 val에 대입(할당) 합니다."라고 말 합니다.
* 그래서 "="를 대입 연산자 또는 할당 연산자라고 부릅니다. 우리 생활에서 "="은
* 같음을 의미하지만 대부분의 프로그래밍 언어에서는 대입 연산자는 같음을 의미하지
* 않고 우측의 데이터를 좌측의 변수에 대입하는 기능을 제공하는 연산자로 사용됩니다.
**/
var val;
val = 100;

// 변수 선언과 동시에 초기화
var value = 200;

// HTML 문서에 출력 - 화면에 출력됨
document.write('val : ' + val + '<br/>');
document.write('value : ' + value + '<br/>');

/* 자바스크립트는 어떤 값이 변수에 저장(할당) 되는 순간 그 변수의 타입(자료형)이
* 결정되는데 이렇게 변수에 값이 저장될 때 마다 변수의 자료형이 결정되는 방식을
* 동적으로 자료의 종류가 결정되기 때문에 동적타입핑이라고 부릅니다.
* 자바와 같은 프로그래밍 언어는 변수를 선언할 때 그 변수가 다룰 데이터의 타입을
* 미리 결정해서 변수를 선언할 때 데이터 타입을 미리 지정해야 하고 이 후에 그 변수에는
* 이미 결정한 데이터 타입만을 대입할 수 있지만 자바스크립트는 동적타입핑을 지원하기
* 때문에 변수 하나에 여러 가지 데이터 타입을 대입할 수 있습니다.
**/
value = "자바스크립트";

// 여러 변수를 먼저 선언한 후에 변수를 초기화
var str, pi;
str = "안녕하세요", pi = 3.141592;

// HTML 문서에 출력 - 화면에 출력됨
document.write('str : ' + str + ', pi : ' + pi + '<br/>');
</script>
</head>
<body>
</body>
</html>

```

## ▶ 예제 2-2 변수와 자료형

- javascript02\_02.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>변수와 자료형</title>
```

```
<script>
```

```
/* 리터럴(Literal)
```

- \* 프로그램에서 리터럴(Literal)이란 변수에 저장되는 값 자체를 의미하거나
  - \* 어떤 자료형에 대한 값을 표현하는 방법을 말합니다. 다시 말해서 숫자 153이라는
  - \* 값을 프로그램 코드에서 표현한다면 아래의 숫자 리터럴 코드와 같이 표현하는데
  - \* 이 코드에서 '153' 자체를 숫자 리터럴이라고 부릅니다. 이외에도 자바스크립트에서
  - \* 사용하는 다양한 데이터를 표현하기 위해서 아래 코드들과 같이 문자열 리터럴,
  - \* 논리 리터럴, 함수 리터럴, 배열 리터럴, 객체 리터럴 등이 있습니다.
- ```
**/
```

```
// 숫자 리터럴 - 정수 리터럴
```

```
var num = 153;
```

```
// 문자열 리터럴
```

```
var str = '문자열 데이터 저장';
```

```
// 논리 리터럴
```

```
var booleanVal = false;
```

```
// 함수 리터럴
```

```
var func = function() {
```

```
    document.write('자료형에 따른 변수에 값 할당하기<br/>');
```

```
};
```

```
/* 배열 리터럴 - 배열은 대괄호([ ... ])로 감싼 형태로 표현합니다.
```

- \* 자바스크립트에서 배열에 아래와 같이 여러 데이터 타입을 저장할 수 있습니다.

```
**/
```

```
var arr = ['배열데이터 1', 123, true];
```

```
/* 객체 리터럴
```

- \* 배열안의 하나의 데이터를 요소라고 하지만 객체는 프로퍼티라고 부릅니다.
- \* 배열은 인덱스를 이용해 요소에 접근하지만 객체는 프로퍼티의 이름을 통해
- \* 접근할 수 있습니다. 이렇게 이름을 키로 접근할 수 있는 배열을 연관배열이라 합니다.
- \* 아래와 같이 객체의 프로퍼티에 함수가 지정되면 이를 메소드(Method)라고 부릅니다.



```

    **/
var obj = {
    x: 1,
    y: 2,
    name: '홍길동',
    func: function() { document.write("나는 obj 메소드 + <br/>"); }
};

document.write("arr : " + arr + '<br/>');
func();
document.write('obj.x : ' + obj.x + ', obj.y : ' + obj.y
    + ', obj["name"] : ' + obj['name'] + '<br/>');

// obj 객체의 메소드 호출
obj.func();

// 변수를 선언하고 초기화 하지 않음 - 변수에 값을 할당 하지 않음
var data;

/* 아래 코드의 실행 결과는 모두 undefined(미정의 값)이 출력됨
 * undefined(미정의 값)는 변수의 값이 정의되어 있지 않음을 의미하며
 * 변수가 초기화 되지 않은 경우와 객체에서 아직 초기화 되지 않은 프로퍼티를
 * 사용할 경우 undefined가 반환됩니다.
 */
document.write("data : " + data + ", obj.age : " + obj.age);
</script>
</head>
<body>
</body>
</html>

```

## ▶ 예제 2-3 산술 연산자

- javascript02\_03.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>산술 연산자</title>
<script>
    //변수 a, b를 선언하고 각각 7과 3으로 초기화
    var a = 7, b = 3;

    /* 사칙 연산자

```

```

    * 더하기 : + , 빼기 : -, 곱하기 : *, 나누기 : / , 나머지 : %, 거듭제곱 : **
    **/
document.write("7 * 3 = " + a * b + ", 7의 3승 = " + a ** 3 + "<br>");
document.write("7 / 3 = " + a / b + ", 7 % 3 = " + a % b + "<br>");

/* 증감 연산자
* 증감 연산자는 연산 대상이 되는 피연산자가 하나인 단항 연산자로 피연산자의 값을
* 1 증가 또는 1 감소하는 연산자입니다. 증감 연산자는 아래와 같이 피연산자의 앞에
* 지정할 수도 있고 뒤에 지정할 수도 있습니다. 앞에 지정하는 경우를 선 증감 연산자라고
* 하며 뒤에 지정하는 경우를 후 증감 연산자라고 합니다. 선 증감 연산자는 다음 연산을
* 수행하기 전에 피연산자의 값을 먼저 1 증감 시키고 다음 연산을 수행하며 후 증감
* 연산자는 다음 연산을 수행한 후에 피연산자의 값을 1 증감 시킨다.
*
* 아래 코드에서 a++은 앞에 있는 문자열 데이터 "a++ = "과 더하기(+) 연산을
* 수행한 후에 a의 값을 1 증가시키고 ++b는 먼저 b의 값을 1 증가 시킨 후에 앞에
* 있는 문자열 데이터 "++b = "과 더하기(+) 연산을 수행합니다.
**/
document.write("a++ = " + a++ + ", a : " + a + "<br>");
document.write("++b = " + ++b + ", b : " + b + "<br>");

/* 연결 연산자
* 다음과 같이 문자열과 문자열의 더하기(+) 연산은 앞뒤의 문자열을 연결해 주는 연산을
* 수행하며 문자열과 숫자의 더하기(+) 연산도 문자열을 우선하여 숫자를 문자열로 변환한
* 후에 앞뒤의 문자열을 연결하는 연산을 수행합니다.
**/
document.write("Hello + JavaScript : " + "Hello " + "JavaScript" + "<br>");
document.write("20 + line : " + 20 + "line" + "<br>");
</script>
</head>
<body>
</body>
</html>

```

## ▶ 예제 2-4 비교 연산자와 논리 연산자

- javascript02\_04.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>비교 연산자와 논리 연산자</title>
<script>
    /* 비교 연산자
    * 비교 연산자는 좌변과 우변의 값을 비교하여 그 결과를 논리 값으로 반환하는 연산자입니다.

```

```

* 비교 연산자는 좌변을 기준으로 좌변의 값이 우변의 값 보다 큰지(>), 크거나 같은지(>=)
* 작은지(<), 작거나 같은지(<=), 같은지(==), 같지 않은지(!=) 등등을 비교할 수 있습니다.
**/
document.write("10 > 30 = " + (10 > 30) + "<br>");
document.write("10 <= 30 = " + (10 <= 30) + "<br>");
document.write("10 != 30 = " + (10 != 30) + "<br><br>");

// 문자열을 비교 연산자로 비교하면 사전적인 위치에 따라서 크고 작음을 비교합니다.
document.write("<안드로이드> > 자바 : " + ("안드로이드" > "자바") + "<br><br>");

/* 논리 연산자
* 논리 연산자는 논리 값을 비교하여 그 결과를 논리 값으로 반환하는 연산자로 연산
* 결과를 논리 값으로 반환하는 비교 연산자와 같이 많이 사용됩니다. 논리 연산자의 종류는
* &&(AND), ||(OR), !(NOT) 3가지 종류가 있으며 && 연산자는 좌변과 우변을 비교해
* 모두 참(true)이면 참(true)을 그렇지 않으면 거짓(false)을 반환하며 || 연산자는
* 좌변과 우변을 비교해 하나만 참(true)이면 참(true)을 반환하는 연산자입니다. 또한
* ! 연산자는 연산 대상이 되는 피연산자가 하나인 단항 연산자로 피연산자의 논리 값이
* 참(true)이면 거짓(false)을 거짓(false)이면 참(true)으로 반전하는 연산자입니다.
**/
// 다음과 같이 비교 연산자와 논리 연산자를 조합해 사용하는 경우가 많다.
document.write("7 > 5 && 3 > 1 : " + (7 > 5 && 3 > 1) + "<br>");
document.write("3 > 1 || 7 < 5 : " + (3 > 1 || 7 < 5) + "<br><br>");

// 부정 연산자는 피연산자의 논리 값을 반전하는 역할을 합니다.
document.write("!true : " + !true + "<br><br>");

/* 문자열과 수치형 데이터의 연산에서 플러스(+) 연산은 문자열을 우선하여
* 문자열 연결 연산을 수행하고 비교 연산이나 나머지 사칙연산은 문자열을
* 수치형으로 변환 후 연산을 수행합니다. 등가연산(==)은 변수의 값이 같은지만
* 비교하고 동치연산(===)은 변수의 자료형과 값이 같은지를 비교합니다.
**/
document.write('15 == "15" : ' + (15 == "15") + "<br>");
document.write('15 === "15" : ' + (15 === "15") + "<br><br>");

// 문자열이 0으로 시작하면 진수 표현으로 인식합니다.
document.write("<0xF> == 15 : ' + (<0xF> == 15) + "<br>");
document.write("<0xF> === 15 : ' + (<0xF> === 15) + "<br><br>");

// 논리 값을 비교 연산이나 플러스(+) 연산을 하면 true=1, false=0으로 변환되어 계산됩니다.
document.write("<true> > <>false> : " + (true > false) + "<br>");
document.write("<>false> + 0 = " + (false + 0) + "<br><br>");
</script>
</head>
<body>
</body>

```

```
</html>
```

## ▶ 예제 2-5 복합 대입연산자

- javascript02\_05.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>복합 대입연산자</title>
```

```
<script>
```

```
    // 다양한 방법으로 변수를 선언하고 초기화
```

```
    var num1 = 10, num2 = 20, num3 = 30, str;
```

```
    /* 아래의 식에서 사용한 "+=", "/=", "%=" 연산자 들은 모두 먼저 더하기(+),
```

```
    * 나누기(/), 나머지(%) 연산을 수행한 후에 좌측의 변수에 대입하는 두 가지
```

```
    * 동작(연산)을 수행하는 연산자로 이런 유형의 대입 연산자를 복합 대입연산자라고
```

```
    * 부릅니다. 아래에서 num1 += 30;의 동작은 먼저 num1 + 30의 연산을 수행한
```

```
    * 후에 그 결과를 다시 num1에 대입하게 됩니다.
```

```
    **/
```

```
    num1 += 30;
```

```
    num2 /= 5;
```

```
    num3 %= 7;
```

```
    // 문자열과 수치 데이터의 + 연산은 문자열 연결 연산이 됩니다.
```

```
    document.write('num1 : ' + num1 + ', num2 : '
```

```
                  + num2 + ', num3 : ' + num3);
```

```
    document.write('<br/><br/>');
```

```
    str = '<ul>';
```

```
    str += ' <li>자바스크립트 완벽가이드</li>';
```

```
    str += ' <li>JavaScript Cookbook</li>';
```

```
    str += ' <li>자바스크립트 마스터북</li>';
```

```
    str += '</ul>';
```

```
    document.write(str);
```

```
</script>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

### 3. 조건문

#### ▶ 예제 3-1 if 조건문

- javascript03\_01.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>if 조건문</title>
<script>
    var x = 10, score = 79;

    document.write("배수 구하기<br>");
    if(9 % 3 == 0) {
        document.write("9 는 3의 배수 입니다.<br><br>");
    }

    document.write("홀/짝수 구하기<br>");
    if(x % 2 == 0) {
        document.write(x + "는 짝수<br><br>");
    } else {
        document.write(x + "는 홀수<br><br>");
    }

    document.write("합격 여부<br>");
    if(score > 80) {
        document.write("축하 합니다. 합격 입니다.<br><br>");
    } else {
        document.write("아쉽네요 다음 기회를...<br><br>");
    }
</script>
</head>
<body>
</body>
</html>
```

#### ▶ 예제 3-2 if else 조건문

- javascript03\_02.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
```

```

<title>if else 조건문</title>
<script>
    function closeWindow() {

        /* Confirm 대화상자는 사용자가 창을 강제로 닫거나 취소 버튼을
         * 클릭하면 false를 반환하고 확인 버튼을 클릭하면 true를 반환합니다.
         * confirm() 함수는 window 객체에 정의되어 있는 함수입니다.
         */
        var input = confirm('프로그램을 종료 하시겠습니까?')

        /* if 문은 조건식이 참일 때 실행되는 조건문 입니다.
         * if(조건식) { 조건식이 참일 때 실행 }
         * if(조건식) { 조건식이 참일 때 실행 } else { 조건식이 거짓일 때 실행 }
         */
        if(input) {
            // 현재 브라우저 창을 닫는다.
            window.close();
            //this.close();

        } else {
            alert('그럴 줄 알았어요..^_^');
        }
    }
</script>
</head>
<body>
    <!--
        버튼에 onclick 속성에 호출할 자바스크립트 함수를 지정하여 버튼이 클릭되었을 때
        필요한 기능을 동작시킨다. 아래와 같이 요소의 이벤트 속성에 이벤트 처리함수를
        등록하여 이벤트를 처리하는 방식을 인라인 이벤트 모델이라고 합니다.
    -->
    <input type="button" value="현재창 닫기" onclick="closeWindow()"/>
</body>
</html>

```

### ▶ 예제 3-3 if else if 조건문

- javascript03\_03.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>if else if 조건문 - 배수 구하기</title>
<script>

```

```

var num = 14;

/* 사용자가 입력한 값이 3의 배수인지 확인
 * 어떤 수를 3으로 나눠서 나머지가 0이면 그 수는 3의 배수입니다.
 */
if(num % 3 == 0) {
    document.write(num + "은 3의 배수 입니다.");

    /* 사용자가 입력한 값이 7의 배수인지 확인
     * 어떤 수를 7로 나눠서 나머지가 0이면 그 수는 7의 배수입니다.
     */
} else if(num % 7 == 0) {
    document.write(num + "은 7의 배수 입니다.");

} else {
    document.write(num + "은 3과 7의 배수 아닙니다.");
}
</script>
</head>
<body>
</body>
</html>

```

### ▶ 예제 3-4 if else if 조건문

- javascript03\_04.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>if else if 조건문 - 숫자를 입력 받아서 배수 구하기</title>
<script>
    function inputMultiple() {

        /* html 문서에서 name 속성의 값이 form1인 폼 안에서 name 속성의
         * 값이 inputNum인 input 요소를 찾아서 이 입력상자에 입력된 값을 읽어온다.
         * 폼 안의 컨트롤들은 value 속성을 이용해 값을 설정하거나 읽어올 수 있습니다.
         */
        var inputNum = document.form1.inputNum.value;

        /* html 문서에서 document 객체의 querySelector() 함수를 이용해서
         * 결과를 출력할 id가 result인 문서 객체를 구합니다.
         */
        var result = document.querySelector("#result");
    }

```

```

/* 사용자 입력한 값이 숫자가 아닌지 확인
 * isNaN() 함수는 인수의 값이 숫자가 아니면 true 숫자면 false를 반환합니다.
 */
if(isNaN(inputNum)) {
    alert('3과 7의 배수를 정수로 입력해주세요')

    /* html 문서에서 name 속성의 값이 form1인 폼 안에서 name 속성의
     * 값이 inputNum인 input 요소를 찾아서 이 입력상자에 입력된 값을 지운다.
     * 폼 안의 컨트롤들은 value 속성을 이용해 값을 설정하거나 읽어올 수 있습니다.
     */
    document.form1.inputNum.value = "";

/* 사용자가 입력한 값이 3의 배수인지 확인
 * 어떤 수를 3으로 나눠서 나머지가 0이면 그 수는 3의 배수입니다.
 */
} else if(inputNum % 3 == 0) {

    /* 결과를 출력할 div 요소의 innerHTML 속성을 이용해 strong 요소를
     * 포함하여 결과를 출력합니다.
     */
    result.innerHTML =
        "<strong>입력 값 : " + inputNum + "은 3의 배수 입니다.</strong>";

    // 다음 입력을 위해서 이전에 입력한 input 요소의 값을 지운다.
    document.form1.inputNum.value = "";

/* 사용자가 입력한 값이 7의 배수인지 확인
 * 어떤 수를 7로 나눠서 나머지가 0이면 그 수는 7의 배수입니다.
 */
} else if(inputNum % 7 == 0) {
    result.innerHTML =
        "<strong>입력 값 : " + inputNum + "은 7의 배수 입니다.</strong>";

    document.form1.inputNum.value = "";

} else {
    result.innerHTML =
        "<strong>입력 값 : " + inputNum + "은 3과 7의 배수가 아닙니다.</strong>";

    document.form1.inputNum.value = "";
}
}
</script>
</head>

```



```

<body>
  <form name="form1">
    3의 배수 또는 7의 배수를 입력해 주세요 : <br/><br/>
    <input type="text" name="inputNum"/>
    <input type="button" value="배수 입력" onclick="inputMultiple()"/>
  </form>
  <div id="result">
  </div>
</body>
</html>

```

### ▶ [연습문제 3-1] 조건문을 이용한 학점 출력

다음 그림과 같이 폼을 만들고 사용자로부터 점수를 입력받아 HTML 문서의 DIV 박스에 점수와 학점을 출력하는 프로그램을 작성해보십시오.

```

<body>
  <form name="inputForm">
    점수를 입력해 주세요
    <input type="text" name="score" id="score"/>
    <input type="button" value="체크하기" onclick="choose()"/>
  </form>
  <div id="result"></div>
</body>

```

점수를 입력해 주세요

- 조건문을 사용하여 입력된 점수에 따라서 각각 아래와 같이 출력되게 작성하십시오.  
 90 ~ 100점 사이 : 입력된 점수(A)  
 80 ~ 89점 사이 : 입력된 점수(B)  
 70 ~ 79점 사이 : 입력된 점수(C)  
 60 ~ 69점 사이 : 입력된 점수(D)  
 이외 점수 : 입력된 점수(F)
- 입력된 데이터가 숫자가 아닌 경우 적절한 경고 창을 띄우고 데이터는 출력되지 않게 하시오.
- 입력된 점수를 학점으로 출력한 기존 데이터에 줄을 바꿔가며 다음과 같이 출력되게 하시오.

점수를 입력해 주세요

91점(A)  
50점(F)  
70점(C)

### ▶ 예제 3-5 switch 조건문

- javascript03\_05.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>switch 조건문</title>
<script>
    function choose() {

        // 문서에서 id가 result인 문서 객체를 선택합니다.
        var result = document.getElementById('result');

        // 문서에서 이름이 inputForm 하위의 score에 입력된 값을 읽어온다.
        var inputNum = document.inputForm.score.value;

        // 입력된 학점을 대문자로 변환해 조건을 체크합니다.
        switch(inputNum.toUpperCase()) {

            /* switch 문의 () 안의 값이 A 이면 아래의 case 'A' 부분이 실행되고 switch
            * 문의 종료됩니다. 이렇게 switch 문의 () 안의 값에 따라서 해당하는 case 문 안의
            * 코드가 실행되고 해당하는 부분이 없을 때는 default 문의 코드가 실행됩니다.
            * default가 필요하지 않으면 생략할 수 있습니다.
            */
            case 'A' :
                // id가 result인 문서 객체의 하위에 html 태그를 포함해 데이터를 추가합니다.
                result.innerHTML = '90점 이상<br/>';
                inputForm.score.value = '';
                break;
            case 'B' :
                result.innerHTML = '80 ~ 89점<br/>';
                inputForm.score.value = '';
                break;
            case 'C' :
                result.innerHTML = '70 ~ 79점<br/>';
                inputForm.score.value = '';
                break;
```

```

        case 'D' :
            result.innerHTML = '60 ~ 69점<br/>';
            inputForm.score.value = '';
            break;
        default :
            result.innerHTML = '60점 미만<br/>';
            inputForm.score.value = '';
    }
}
</script>
</head>
<body>
    <form name="inputForm">
        학점을 입력해 주세요(예 : A, B, C)<br/>
        <input type="text" name="score" id="score"/>
        <input type="button" value="체크하기" onclick="choose()"/>
    </form>
    <div id="result"></div>
</body>
</html>

```

### ▶ 예제 3-6 삼 항 연산자

- javascript03\_06.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>삼 항 연산자</title>
<script>
    /* Prompt 대화상자는 사용자로부터 데이터를 입력 받을 수 있는 대화상자로
     * 사용자가 Prompt 대화상자의 창을 강제로 닫거나 취소 버튼을 클릭하면
     * null을 반환하고 확인 버튼을 클릭하면 대화상자에 입력된 문자열을 반환합니다.
     * 참고로 prompt() 함수는 window 객체에 정의되어 있는 함수입니다.
     */
    var input = prompt("숫자를 입력해주세요", 0);

    /* 삼 항 연산자는 조건 연산자라고도 하면 피연산자가 3개라 삼 항 연산자라고 부릅니다.
     * 조건 ? 조건이 참일 때 반환 값 : 조건이 거짓일 때 반환 값
     */
    var result = input % 2 == 0 ? "짝수" : "홀수";
    console.log(input + "은 " + result);

    // 삼 항 연산자는 다음과 같이 여러 번 중첩하여 사용할 수 있습니다.

```

```

    result = input > 0 ? "양수임" :
        input == 0 ? "0임" :
            input < 0 ? "음수임" : "값이 입력되지 않음"
    console.log(input + "은 " + result);
</script>
</head>
<body></body>
</html>

```

### ▶ [연습문제 3-2] 중첩 if문을 이용한 주사위 게임

다음 그림과 같이 폼을 만들고 사용자로부터 숫자를 입력받아 아래 설명과 같이 동작하는 주사위 게임을 만들어 봅시다.

```

<body>
  <form name="inputForm">
    게임 룰 선택 : <label><input type="radio" name="rule" value="1"/>높은 수 승</label>
      <label><input type="radio" name="rule" value="0"/>낮은 수 승</label>
    <input type="button" value="게임시작" onclick="diceGame()"/>
  </form>
</body>

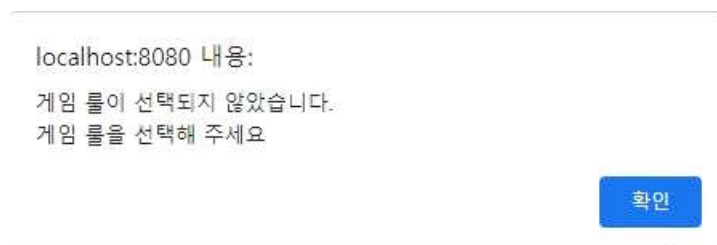
```

게임 룰 선택 : ☐ 높은 수 승 ☐ 낮은 수 승

1. 컴퓨터와 사용자 주사위는 1 ~ 6까지의 난수를 발생하여 생성 하시오.

- Math.random()을 이용

2. 사용자가 게임 룰을 선택하지 않고 “게임시작” 버튼을 클릭하였다면 다음과 같은 경고 창을 띄워서 안내하시오.



3. 사용자로부터 게임 룰이 제대로 선택되고 “게임시작” 버튼이 클릭되면 라디오 버튼에 입력된 게임 룰을 읽어와 현재 게임 룰과, 컴퓨터 숫자, 사용자 숫자 그리고 게임의 결과를 아래와 같이 알림 창으로 출력하시오.

localhost:8080 내용:  
게임 룰은 높은 숫자가 승리!  
컴퓨터 숫자 : 3, 당신의 숫자 : 1 당신이 졌습니다.

확인

localhost:8080 내용:  
게임 룰은 낮은 숫자가 승리!  
컴퓨터 숫자 : 5, 당신의 숫자 : 4 당신이 이겼습니다.

확인

## 4. 반복문

### ▶ 예제 4-1 for 문

- javascript04\_01.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>for 문</title>
<script>
    // for 문을 이용해 1 ~ 100까지 합 구하기
    var sum = 0;

    // for(초기식; 조건식; 증감식) 문을 이용해 1 ~ 100까지 합 구하기
    for(var i = 1; i <= 100; i++) {
        sum += i;
    }
    document.write("<h2>1 ~ 100까지 합 : " + sum + "</h2>");

    document.write("<h2>for 문을 이용해 구구단 7단 출력</h2>");
    var x = 7;
    for(var i = 1; i < 10; i++) {
        document.write(x + " x " + i + " = " + x * i + "<br>");
    }
</script>
</head>
<body></body>
</html>
```

### ▶ 예제 4-2 for 문에서 if 문 사용하기

- javascript04\_02.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>for 문에서 if 문 사용하기</title>
<script>
    document.write("<h2>1 ~ 100까지 짝수와 홀수의 합 구하기</h2>");
    var oddSum = 0;
    var evenSum = 0;

    // 반복문을 돌면서 if 문을 이용해 짝수의 합과 홀수의 합을 구합니다.
```

```

for(var i = 1; i <= 100; i++) {

    /* 반복문 안에서 특정 조건에 해당하는지 판단하기 위해서 if 문 사용
    * 현재 i를 2로 나누어 나머지가 0이면 짝수 그렇지 않으면 홀수로 판단
    **/
    if(i % 2 == 0) {
        evenSum += i;

    } else {
        oddSum += i;
    }
}
document.write("1 ~ 100까지 짝수의 합 : " + evenSum + "<br>");
document.write("1 ~ 100까지 홀수의 합 : " + oddSum + "<br>");

document.write("<h2>1 ~ 100까지 3의 배수와 개수 그리고 합계 구하기</h2>");
var num = 3;
var count = 0;
var sum = 0;
var nums = "";

for(var i = 1; i <= 100; i++) {

    /* 반복문 안에서 if 문을 이용해 3의 배수인지 판단
    * 어떤 수를 3으로 나눠서 그 나머지가 0이면 3의 배수
    **/
    if(i % num == 0) {

        // 3의 배수를 콤마(,)로 구분해 문자열로 저장
        nums += (100 - num >= i) ? i + ", " : i;

        // 3의 배수를 sum에 누적하여 더함
        sum += i;

        // 3의 배수의 count 변수 값을 하나씩 증가
        count++; // count = count + 1;
    }
}
document.write("1 ~ 100까지 3의 배수 :<br>" + nums + "<br>");
document.write("3의 배수의 개수 : " + count + ", 합계 : " + sum + "<br>");
</script>
</head>
<body></body>
</html>

```

### ▶ 예제 4-3 for 문과 for in 문

- javascript04\_03.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>for 문과 for in 문</title>
<script>
    var bookArr = ["자바스크립트 완벽 가이드",
        "자바스크립트 마스터 북",
        "모던 자바스크립트"];

    /* for 문을 이용한 배열 접근
     * for(초기식; 조건식; 증가식) { 조건식이 참인 동안 실행 }
     */
    for(var i = 0; i < bookArr.length; i++) {
        document.write('<li>' + bookArr[i] + '</li>')
    }
    document.write('</ul><br/><br/>');

    document.write('<h2>자바스크립트 참고 도서 - for in 문</h2>');
    document.write('<ul>');

    /* for in 문을 이용한 배열 접근
     * for(가 변수 in 배열 또는 집합 데이터) { 배열의 길이만큼 반복 실행}
     */
    for(var i in bookArr) {
        document.write('<li>' + bookArr[i] + '</li>');

        // bookArr이 배열이므로 가 변수 i에는 배열의 index가 저장됩니다.
        console.log(i);
    }
    document.write('</ul><br/><br/>');
</script>
</head>
<body></body>
</html>
```

### ▶ [연습문제 4-1] 홀수의 개수와 짝수의 개수 구하기

정수 75부터 ~ 355까지 홀수와 짝수의 개수를 구하여 아래와 같이 출력하는 프로그램을 작성하시오.



[실행결과]

75부터 ~ 355까지 홀수의 개수 : 141

75부터 ~ 355까지 짝수의 개수 : 140

▶ [연습문제 4-2] 3의 배수이면서 6의 배수가 아닌 정수와 그 배수의 개수

아래와 같이 57부터 ~ 119까지 3의 배수이면서 6의 배수가 아닌 정수와 그 배수의 개수 그리고 그 배수의 합계를 출력하는 프로그램을 작성하시오.

[실행결과]

57부터 ~ 119까지 3의 배수이면서 6의 배수가 아닌 정수 구하기

3의 배수이면서 6의 배수가 아닌 수 : 57, 63, 69, 75, 81, 87, 93, 99, 105, 111, 117

3의 배수이면서 6의 배수가 아닌 정수의 개수 : 11

3의 배수이면서 6의 배수가 아닌 정수의 합 : 957

▶ 예제 4-4 중첩 for 문을 이용한 구구단 출력

- javascript04\_04.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>중첩 for 문을 이용한 구구단 출력</title>
<script>
    for(var i = 1; i <= 9; i++) {
        for(var j = 2; j <= 9; j++) {
            document.write(j + " x " + i + " = " + i * j + "\t");
        }
        document.write("<br>");
    }
</script>
</head>
<body style="white-space: pre;">
</body>
</html>
```

▶ 예제 4-5 while 문과 do while 문

- javascript04\_05.html

```
<!DOCTYPE html>
```

```

<html>
<head>
<meta charset="UTF-8">
<title>while 반복문과 do while 반복문</title>
<script>
    var i = 0;

    // sum을 초기화 하지 않으면 NaN(Not a Number)
    var sum = 0;

    /* while 문은 조건식이 거짓일 때 단 한 번도 실행되지 않습니다.
     * while(조건식) { 조건이 참인 동안 실행 }
     */
    while(i < 100) {

        /* while 문은 for 문과 같이 증감식 같은 것이 없으므로 while 문 안에서
         * 조건식을 거짓으로 만들 수 있는 코드를 작성하지 않으면 무한 반복문이 됩니다.
         */
        i++;
        if(i % 2 == 0) {
            sum += i;
        }
    }
    document.write('while문의 1 ~ 100까지 짝수 합 : ' + sum + "<br/>");

    i = 0;
    sum = 0;

    /* do~while 문은 최소 한 번은 실행을 합니다.
     * do { 먼저 실행된 후에 조건식이 참인지 확인 } while(조건식);
     */
    do {

        /* do~while 문도 while 문과 마찬가지로 do~while 문의 실행 문장 안에서
         * 조건식을 거짓으로 만들 수 있는 코드를 작성하지 않으면 무한 반복문이 됩니다.
         */
        i++;
        if(i % 2 == 1) {
            sum += i;
        }
    } while(i < 100);
    document.write('do while문의 1 ~ 100까지 홀수의 합 : ' + sum);
</script>
</head>
<body></body>

```

```
</html>
```

## ▶ 예제 4-6 break와 continue

- javascript04\_06.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>break와 continue</title>
<script type="text/javascript">
    var i = 0;
    while(true) {
        if(!confirm('계속 하시겠습니까? - while')) { // 사용자가 취소를 선택하였으면
            alert('while 문 ' + i + '번 실행 후 종료!');

            // 현재 실행 중인 반복문을 빠져 나간다.
            break;

        } else { // 사용자가 확인을 선택하였으면

            /* continue를 만나면 while 문의 조건식으로 올라간다.
             * 그러므로 아래 i++은 실행되지 않기 때문에 i는 증가되지 않습니다.
             */
            continue;
        }
        i++;
    }
</script>
</head>
<body></body>
</html>
```

## ▶ [연습문제 4-3] 1 ~ 60까지 한 줄에 10단위씩 출력하기

반복문을 사용해 아래와 같이 1부터 ~ 60까지의 수를 한 줄에 10단위씩 출력하는 프로그램을 작성 하시오.

[실행결과]

```
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, 32, 33, 34, 35, 36, 37, 38, 39, 40
```

41, 42, 43, 44, 45, 46, 47, 48, 49, 50  
51, 52, 53, 54, 55, 56, 57, 58, 59, 60

## 5. 함수

### ▶ 예제 5-1 함수 정의하고 사용하기

- javascript05\_01.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>함수 정의하고 사용하기</title>
<script>
    /* 이름 없는(익명) 함수 정의 - 함수 리터럴 표현으로 정의
     * 이름이 없기 때문에 변수에 함수를 할당해 다른 곳에서 호출 할 수 있습니다.
     **/
    var rectangle = function(width, height) {
        return width * height;
    }

    /* function 예약어를 사용해 함수에 이름을 부여해 정의
     * 함수를 정의할 때 함수의 괄호는 그 함수가 실행되면서 필요한 데이터를
     * 외부로부터 받을 수 있는 통로입니다. 외부에서 데이터를 받을 필요가 없으면
     * 괄호 안을 비워서 빈 괄호로 표현하면 되고 함수 실행에 필요한 데이터를
     * 외부로부터 받으려면 아래와 같이 괄호 안에 변수를 지정하면 됩니다.
     * 함수 괄호 안의 변수는 함수를 실행하기 위해서 필요한 입력 값이며 함수
     * 외부에서 함수 안쪽으로 값을 넣어서 사용할 수 있도록 연결해 준다는
     * 의미로 매개변수(parameter)라고 부릅니다.
     **/
    function triangle(width, height) {
        return width * height / 2;
    }

    /* 아래와 같이 함수를 사용하는 것을 "함수를 호출합니다."라고 말합니다.
     * 일반적으로 함수를 호출할 때는 함수에 정의된 매개변수의 순서에 맞게
     * 값을 지정해야 합니다. 아래와 같이 함수를 호출할 때 괄호 안에 넣어주는
     * 값을 함수의 매개변수로 전달하는 실제 값이라는 의미하며 전달인수 또는
     * 전달인자라고 하며 간단히 인수(인자, argument)라고 부릅니다.
     * 매개변수와 인수를 정리하자면 매개변수는 함수를 정의할 때 함수의
     * 괄호 안에 선언하는 변수를 의미하며 함수를 호출할 때 지정하는 값은
     * 인수라고 구분하면 됩니다. 참고로 자바스크립트에서는 함수를 호출할 때
     * 인수를 지정하지 않아도 함수를 호출할 수 있지만 함수로부터 정상적으로
     * 원하는 기능을 얻기 위해서는 인수를 정확하게 지정해야 합니다.
     **/
    document.write('사각형의 넓이 : ' + rectangle(10, 20) + '<br/>');
    document.write('삼각형의 넓이 : ' + triangle(20, 5) + '<br/>');
```

```

</script>
</head>
<body></body>
</html>

```

## ▶ 예제 5-2 함수의 매개변수와 반환 값

- javascript05\_02.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>함수의 매개변수와 반환 값</title>
<script>
    /* 익명 함수 방식으로 함수 정의
    * 아래의 폼에서 "나이 입력" 버튼이 클릭되면 실행되는 함수
    *
    * 함수가 실행되면서 외부에서 데이터를 받을 필요가 없으면 아래와 같이
    * 괄호를 비워서 함수를 정의하면 됩니다. 이 함수의 기능은 버튼이 클릭되었을
    * 때 나이를 입력 받아서 폼 안의 나이 입력상자에 입력해 주는 기능을 제공하는
    * 함수이며 필요한 모든 데이터는 함수 내부에서 처리하므로 외부에서 데이터를
    * 받을 필요가 없기 때문에 별도로 매개변수를 선언하지 않았다.
    */
    var inputAge = function() {

        // prompt 창에서 나이를 입력 받는다.
        var age = prompt("나이를 입력해주세요");
        form1.age.value = age;

        /* 함수는 실행한 결과를 return 문을 사용해 반환할 수 있지만 반환할
        * 필요가 없다면 return 문을 생략하면 됩니다. return 문은 현재 실행되는
        * 함수의 실행을 종료하고 함수를 호출한 곳으로 돌아가라는 명령으로 현재
        * 실행되고 있는 함수의 종료를 의미하며 반환 값이 없으면 생략하면 됩니다.
        * 만약 호출한 곳으로 돌아갈 때 실행한 결과 값을 넘겨 주려면 return 문
        * 뒤에 넘겨 주려는 값을 지정하면 됩니다.
        */
        // return;
        // return age;
    }

    /* function 예약어를 사용해 함수에 이름을 부여해 함수 정의
    *
    * 함수가 실행되면서 외부에서 데이터를 받을 필요가 있으면 아래와 같이
    * 괄호에 적절한 변수를 선언하면 됩니다. 이 함수는 선택상자(select 박스)를

```

```

* 매개변수로 받아서 선택상자에서 현재 선택된 텍스트를 추출해 반환하는
* 기능을 제공하는 함수로 선택상자를 매개변수로 받을 수 있도록 선언하였습니다.
**/
function getSelectValue(elem) {

    /* 매개변수로 받은 elem은 선택상자(select 요소)이며 이 요소의
    * selectedIndex 속성으로 현재 선택된 항목의 index 값을 알 수
    * 있으며 이 속성을 이용해 아래와 같이 options 속성에 현재 선택된
    * 항목의 index를 지정하면 value 또는 text를 읽어 올 수 있습니다.
    *
    * 함수는 return 문을 이용해 실행한 결과 값을 반환할 수 있습니다.
    * 만약 실행 결과를 반환할 필요가 없다면 return 문을 생략하면 됩니다.
    */
    return elem.options[elem.selectedIndex].text;
}

/* 관심분야 선택상자(select 박스)에서 선택이 변경될 때마다 실행되는 함수
*
* 선택상자에서 항목 선택이 변경될 때 마다 선택상자에 접근해서 필요한 작업을
* 해야 하므로 함수의 매개변수를 선언해서 선택상자를 받을 수 있도록 정의하였습니다.
**/
var changeSelect = function(elem) {

    /* 매개변수로 받은 elem을 다른 함수를 호출하면서 인수로 지정 했습니다.
    * getSelectValue() 함수의 반환 값이 없으면 undefined가 반환됩니다.
    */
    var val = getSelectValue(elem) + ' 선택됨';
    alert(val);
}

// 폼이 submit 될 때 호출되는 함수
function registSubmit() {
    var name = document.form1.name.value;
    var age = document.form1.age.value;
    var interest = form1.choice.options[form1.choice.selectedIndex].text;

    alert('입력된 정보\n'
        + '이름 : ' + name + "\n"
        + '나이 : ' + age + "\n"
        + '관심분야 : ' + interest);

    return false;
}
</script>

```

```

<!-- jQuery를 이용하면 보다 간편하게 DOM을 제어할 수 있습니다. -->
<script src="js/jquery-3.3.1.min.js"></script>
<script>
    $(function() {
        // select 박스의 선택이 변경될 때 발생하는 이벤트를 처리하는 코드
        $("#choice").change(function() {
            console.log(this + " , " + $(this).val() + "선택됨");
        });
    });
</script>
</head>
<body>
<!--
    form 요소의 onsubmit 속성에 함수를 지정하면 해당 폼이 submit 될 때
    즉 폼에서 submit 이벤트가 발생할 때 지정한 함수가 호출됩니다.
-->
<form name="form1" onsubmit="return registSubmit();">
    이름 : <input type="text" name="name"/><br/>
    나이 : <input type="text" name="age" readonly/>
    <input type="button" value="나이 입력" onclick="inputAge();"/><br/>
    관심분야 : <select name="choice" id="choice" onchange="changeSelect(this);">
        <option>자바</option>
        <option>JSP</option>
        <option>자바스크립트</option>
        <option>안드로이드</option>
        <option>스프링</option>
    </select><br/><br/>
    <input type="submit" value="등록하기" />
</form>
</body>
</html>

```

### ▶ 예제 5-3 가변인자 함수 정의하기

- javascript05\_03.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>가변인자 함수 정의하기</title>
<script>

```

```

function multiplyFunc(num1, num2) {

```



```

/* 매개 변수의 수를 정확히 맞추기 위해 arguments 객체를 사용하였습니다.
 * 매개 변수 정보를 관리하는 arguments 객체는 함수 안에서만 사용할 수 있습니다.
 */
if(arguments.length !== 2) {
    alert('인수의 수가 다릅니다. 인수는 2개만 입력하세요');
    return;
}
alert(num1);
// 인수 값이 undefined 이거나 숫자가 아니면 기본 값을 지정함
if(num1 == 'undefined' || isNaN(num1)) { num1 = 1; }
if(num2 == 'undefined' || isNaN(num2)) { num2 = 1; }

console.log("multiplyFunc(num1, num2) : " + num1 * num2);
}

/* 함수 정의 문에 매개변수가 선언되어 있지 않아도 함수 안에서
 * arguments 객체를 통해서 함수로 전달된 매개변수를 읽을 수 있습니다.
 */
function multiplyParams() {
    var result = 0;
    for(var i in arguments) {
        if(isNaN(arguments[i])) {
            continue;
        }
        result += arguments[i];
    }
    console.log("multiplyParams() : " + result);
}
</script>
</head>
<body>
    <div>
        <!--
            자바스크립트는 함수 정의와 다르게 인수의 개수를 다르게 호출할 수 있지만
            함수의 기능을 제대로 사용하기 위해서는 인수를 정확하게 지정해야 합니다.
        -->
        <input type="button" value="multiplyFunc(1, 2) 호출"
            onclick="multiplyFunc(10, 30)" />
        <input type="button" value="multiplyParams() 호출"
            onclick="multiplyParams(10, 20, 'NaN', 30, 40)" />
    </div>
</body>
</html>

```

▶ [연습문제 5-1] 함수를 사용해 사칙연산을 수행하는 계산기 만들기

다음 그림과 같이 UI를 구성하여 사칙연산을 수행하는 계산기를 만들어 봅시다.



1. 버튼이 클릭되면 텍스트 입력상자에 클릭된 숫자가 표시되고 연산자가 클릭되기 전까지 연속적인 숫자로 표시되게 하시오.
2. 첫 번째 입력된 숫자가 '0'이면 텍스트 입력상자에 입력되지 않게 작성하시오.
3. 텍스트 입력상자에 입력된 숫자 없이 연산자 버튼이 먼저 클릭되면 “먼저 숫자를 입력해 주세요”라는 경고 창을 띄우고 이미 숫자가 입력된 상태라면 기존에 입력된 숫자를 지우고 새롭게 입력한 숫자가 텍스트 입력상자에 표시 되도록 하시오.
4. 연산할 숫자가 입력되지 않은 상태에서 “=” 클릭되면 아무 동작도 하지 않도록 하시오.
5. 모두 정상적으로 입력된 상태에서 “=” 버튼이 클릭되면 입력된 숫자와 연산자를 바탕으로 계산한 결과 값이 텍스트 입력상자에 출력되도록 하시오.