

HTML5 Introduction

- HTML5의 새로운 기능

HTML5에 대한 DOCTYPE 선언은 매우 간단합니다.

```
<!DOCTYPE html >
```

문자 인코딩 (charset) 선언도 매우 간단합니다.

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

HTML5 연습:

```
<!DOCTYPE html >
<html >
<head >
<meta charset ="UTF-8">
<title >Title of the document </title >
</head >

<body >
Content of the document.....
</body >

</html >
```

HTML5의 기본 encoding은 UTF-8. 입니다

- 새로운 HTML5 구성요소(Elements)들

가장 주목되는 새로운 HTML5 구성요소:

- semantic(의미)가 있는 구성요소: <header>, <footer>, <article>, <section>.
- 구성요소의 새로운 속성들: number, date, time, calendar, and range.
- 새로운 그래픽 구성요소들: <svg> and <canvas>.
- 새로운 멀티미디어 구성요소들: <audio> and <video>.

다음 장인 HTML5 지원에서는 "unknown"(새) HTML 요소를 처리하도록 이전 브라우저를 "가르치는" 방법을 배우게 됩니다.

- 새로운 HTML5 API 들 (Application Programming Interfaces)

주목해야할 새로운 HTML5의 API들 in HTML5:

- HTML Geolocation(지리적 위치)
- HTML Drag and Drop
- HTML Local Storage(브라우저 로컬 DB)
- HTML Application Cache(어플리케이션 임시저장소)
- HTML Web Workers

HTML SSE

※ HTML LocalStorage는 쿠키를 대체 할 수있는 강력한 도구입니다.

- HTML5에서 쓰이지 않는 구성요소들

HTML4의 구성요소중 HTML5에서 바뀌는 것들:

삭제된 구성요소	대체된 구성요소
<acronym>	<abbr>
<applet>	<object>
<basefont>	CSS
<big>	CSS
<center>	CSS
<dir>	
	CSS
<frame>	
<frameset>	
<noframes>	
<strike>	CSS, <s>, or
<tt>	CSS

HTML5 마이그레이션 장에서 HTML4에서 HTML5로 손쉽게 마이그레이션하는 방법을 배우게됩니다.

- HTML의 역사

월드 와이드 웹의 초창기부터 많은 HTML 버전이 있었습니다.

Year	Version
1989	Tim Berners-Lee: www 창안
1991	Tim Berners-Lee: HTML 창안
1993	Dave Raggett HTML+ 초안작성
1995	HTML 2.0Working Group 결성
1997	W3C의 HTML 3.2 권장 사항 발표
1999	W3C의 HTML 4.01 권장 사항 발표
2000	W3C의 XHTML 1.0 권장 사항 발표
2008	WHATWG HTML5 초안 작성

2012	WHATWG HTML5 Living Standard
2014	W3C의 HTML5 권장 사항 발표
2016	W3C의 HTML 5.1 후보 추천

1991년부터 1999년까지 HTML은 버전 1에서 버전 4로 발전했습니다.

2000년 W3C (World Wide Web Consortium)는 XHTML 1.0을 권장했습니다. XHTML 구문은 엄격했으며 개발자는 유효하고 "형식이 잘 맞는"코드를 작성해야 했습니다.

2004년 W3C는 XHTML을 위해 HTML 개발을 중단하기로 결정했습니다.

2004년 WHATWG (Web Hypertext Application Technology Working Group)가 구성되었습니다. WHATWG는 웹을 사용하는 방식과 일관되게 HTML을 개발하려고 했지만 이전 버전의 HTML과 호환되지 않는 문제가 발생했습니다.

2004 - 2006년에 WHATWG는 주요 브라우저 공급 업체의 지원을 받았다.

2006년에 W3C는 WHATWG를 지원할 것이라고 발표했습니다.

2008년 첫 번째 HTML5 공개 초안이 공개되었습니다.

2012년 WHATWG와 W3C는 분리를 결정했습니다.

WHATWG는 HTML을 "Living Standard"로 발전시키고 자했습니다. "Living Standard"은 항상 업데이트되고 개선됩니다. 새로운 기능을 추가 할 수는 있지만 이전 기능은 제거 할 수 없습니다.

WHATWG HTML5 Living Standard는 2012년에 발행되었으며 지속적으로 업데이트됩니다.

W3C는 HTML5 및 XHTML 결정판을 발전시켜 표준을 개발하고자했습니다.

W3C HTML5 권장 사항은 2014년 10월 28일에 발표되었습니다.

또한 W3C는 2016년 6월 21일에 HTML 5.1 후보 추천을 게시했습니다.

HTML5 Browser Support

- HTML5 모든 고전적인 Browser를 지원합니다

또한 오래된 브라우저와 새로운 브라우저 모두 인식 할 수없는 요소를 인라인 요소로 자동 처리합니다.

이 때문에 오래된 브라우저가 "알려지지 않은"HTML 요소를 처리하도록 "가르칠"수 있습니다.

IE6 (Windows XP 2001)에서 알려지지 않은 HTML 요소를 처리하는 방법을 가르칠 수도 있습니다.

-
- 시맨틱(Semantic) 요소를 블록 요소로 정의

HTML5는 8 개의 새로운 Semantac 요소를 정의합니다. 이 모든 것은 블록 수준의 요소입니다. 구형 브라우저에서 올바른 동작을 보장하려면 이러한 HTML 요소에 대한 CSS 표시 속성을 다음과 같이 작성하여 실행을 차단하도록 설정할 수 있습니다.

```
header, section, footer, aside, nav, main, article, figure {
    display :block ;
}
```

- 새로운 HTML 구성요소의 추가

브라우저 트릭을 사용하여 HTML 페이지에 새 요소를 추가 할 수도 있습니다.

아래 예제에서는 HTML 페이지에 <myHero>라는 새 요소를 추가하고 스타일을 정의합니다.Example

```
<!DOCTYPE html >
<html >
<head >
    <script >document.createElement ("myHero")</script >
    <style >
        myHero {
            display :block ;
            background-color :#dddddd ;
            padding :50px ;
            font-size :30px ;
        }
    </style >
</head >
<body >

<h1>A Heading </h1>
<myHero >My Hero Element </myHero >

</body>
</html>
```

JavaScript의 document.createElement ("myHero")는 IE 9 이하의 새 요소를 만드는 데 필요합니다.

- Internet Explorer 8의 문제점

대부분의 브라우저에서는 모든 새로운 HTML5 요소에 대해 위에 설명 된 솔루션을 사용할 수 있습니다. 그러나 IE8 (및 이전 버전)에서는 알 수 없는 요소의 스타일을 허용하지 않습니다!

고맙게도 Sjoerd Visscher가 **HTML5Shiv**를 만들었습니다! HTML5Shiv는 9버전 이전의 Internet Explorer에서 HTML5 요소의 스타일을 활성화하는 JavaScript 해결 방법입니다.

Internet Explorer 9 이전의 IE 브라우저에 대한 호환성을 제공하려면 HTML5shiv가 필요합니다.

- HTML5Shiv의 문법

HTML5Shiv는 <head> 태그 내에 있습니다.

HTML5Shiv는 <script> 태그에서 참조되는 javascript 파일입니다.

다음과 같은 새로운 HTML5 요소를 사용할 때 HTML5Shiv를 사용해야 합니다.

<article>, <section>, <aside>, <nav>, <footer>.

github에서 HTML5shiv의 최신 버전을 다운로드하거나에서 CDN 버전을 참조하십시오.

<https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js>

문법구조

```
<head >
  <!--[if lt IE 9]>
    <script src="/js/html5shiv.js"></script>
  <![endif]-->
</head >
```

- HTML5Shiv 예제

사이트에 HTML5Shiv를 다운로드하고 저장하지 않으려면 CDN 사이트에있는 버전을 참조하십시오. HTML5Shiv 스크립트는 스타일 시트 다음에 <head> 요소에 있어야 합니다.

예제)

```
<!DOCTYPE html >
<html >
<head >
<meta charset ="UTF-8">
<!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
<![endif]-->
</head >
<body >

<section >
  <h1 >Famous Cities </h1 >
  <article >
    <h2 >London </h2 >
    <p >런던은 영국의 수도입니다. 영국에서 가장 인구가 많은 도시로 1300 만 명이 넘는
대도시가 있습니다.</p >
  </article >
```

```

<article >
    <h2 >Paris </h2 >
    <p >파리는 프랑스에서 가장 인구가 많은 도시입니다.</p >
</article >

<article >
    <h2 >Tokyo </h2 >
    <p >도쿄는 일본의 중심지이자 도쿄 중심부의 중심지이자 세계에서 가장 인구가 많은 대
도시 지역입니다.</p >
</article >
</section >
</body >
</html >

```

HTML5 새로운 구성요소

- HTML5 새로운 구성요소

다음은 새로운 HTML5 요소 목록과 HTML5 요소가 사용 된 요소에 대한 설명입니다.

새로운 Semantic/Structural 구성요소

HTML5는 더 나은 문서구조(Document Structur)를위한 새로운 요소를 제공합니다.

Tag	설명
<article>	문서의 article 정의
<aside>	페이지 콘텐츠를 제외한 일반 콘텐츠 정의
<bdi>	한 덩어리의 텍스트 방향을 주위와는 별도로 표시하도록 지시
<details>	사용자가 보거나 숨길 수있는 추가 세부 정보를 정의합니다.
<dialog>	dialog box 나 window 정의
<figcaption>	<figure> 구성요소의 caption 정의
<figure>	일러스트레이션, 다이어그램, 사진, 코드 등과 같은 플로우 콘텐츠를 포함
<footer>	문서나 섹션의 꼬리말 정의
<header>	문서나 섹션의 머리말 정의
<main>	문서의 main 콘텐츠 정의
<mark>	강조 텍스트 정의
<menuitem>	컨텍스트 메뉴와 툴바를 정의하는 것으로 command 요소, button 요소등을 이용한 커맨드 목록 정의
<meter>	알려진 범위 내에서의 스칼라 측정 또는 분포 비율을 표현. 디스크 사용 현황, 쿼리 결과의 관련성, 특정 후보에 대한 투표율 등에 사용 이것을 게이지라고도 한다

<nav>	내비게이션 링크 정의
<progress>	작업의 진척도를 표현 정의
<rp>	루비 주석을 지원하지 않는 사용자 에이전트에서 루비 텍스트를 괄호 묶기
<rt>	루비 텍스트의 루비 주석 표현 (아시아권 문자에 사용)
<ruby>	루비 주석을 정의 (아시아권 문자에 사용)
<section>	문서의 섹션 구역 정의
<summary>	details 요소에 포함된 내용의 요약, 캡션, 범례
<time>	date/time 정의
<wbr>	br 요소가 강제로 줄바꿈을 처리하는 것과 달리 부모 요소의 너비에 따라 동적으로 줄바꿈을 처리

다른 HTML5 요소 참조

<http://html5ref.clearboth.org/>

• 새로운 Form 구성요소

새로운 Input Types

새로운 Input Types	새로운 Input 속성
color date datetime datetime-local email month number range search tel time url week	autocomplete autofocus form formation formenctype formmethod formnovalidate formtarget height and width list min and max multiple pattern (regexp) placeholder required step

• HTML5 - 새로운 속성에 대한 문법

HTML5 속성은 4가지 형식을 지원합니다.

아래 예제는 <input> 태그에 사용 된 다양한 구문을 보여줍니다.

Type	예제
Empty	<input type="text" value="John" disabled>
따옴표 없음	<input type="text" value=John>
큰 따옴표	<input type="text" value="John Doe">
작은 따옴표	<input type="text" value='John Doe'>

HTML5에서는 속성에 필요한 내용에 따라 네 가지 구문을 모두 사용할 수 있습니다.

• HTML5 그래픽

Tag	설명
<canvas>	스크립팅(일반적으로 JavaScript)을 통해 그래픽을 그립니다
<svg>	벡터 방식 그래픽을 그립니다

• 새로운 미디어(Media) 구성요소

Tag	설명
<audio>	사운드 콘텐츠 정의
<embed>	HTML이외의 외부 어플리케이션 컨테이너 정의
<source>	미디어 요소 (<video> 및 <audio>)에 대한 여러 미디어 리소스를 정의
<track>	미디어 요소 (<video> 및 <audio>)에 대한 텍스트 트랙을 정의
<video>	비디오나 영화 미디어 정의

HTML5 Semantic Elements

시맨틱스 (semantics) 란 언어로 된 단어와 어구의 의미에 관한 연구입니다.

의미론적 구성요소 = 의미가있는 요소.(Semantic elements = elements with a meaning.)

• Semantic(의미론적) 구성요소란 무엇인가

Semantic 요소는 브라우저와 개발자 모두에게 의미를 명확하게 설명합니다.

비 Semantic 요소의 예 : <div>와 - 내용에 관한 설명이 포함되지 않습니다.

Semantic 요소의 예 : <form>, <table> 및 <article> - 무슨역할을 하는 tag인지 명확합니다

• Browser Support

Chorome	Internet Exporer	FirFox	Safari	Opera
Yes	Yes	Yes	Yes	Yes

HTML5의 Semantic 요소는 대부분의 브라우저에서 지원합니다.

오래된 브라우저에게는 사용법을 “가르쳐” 쓸수있도록 합니다

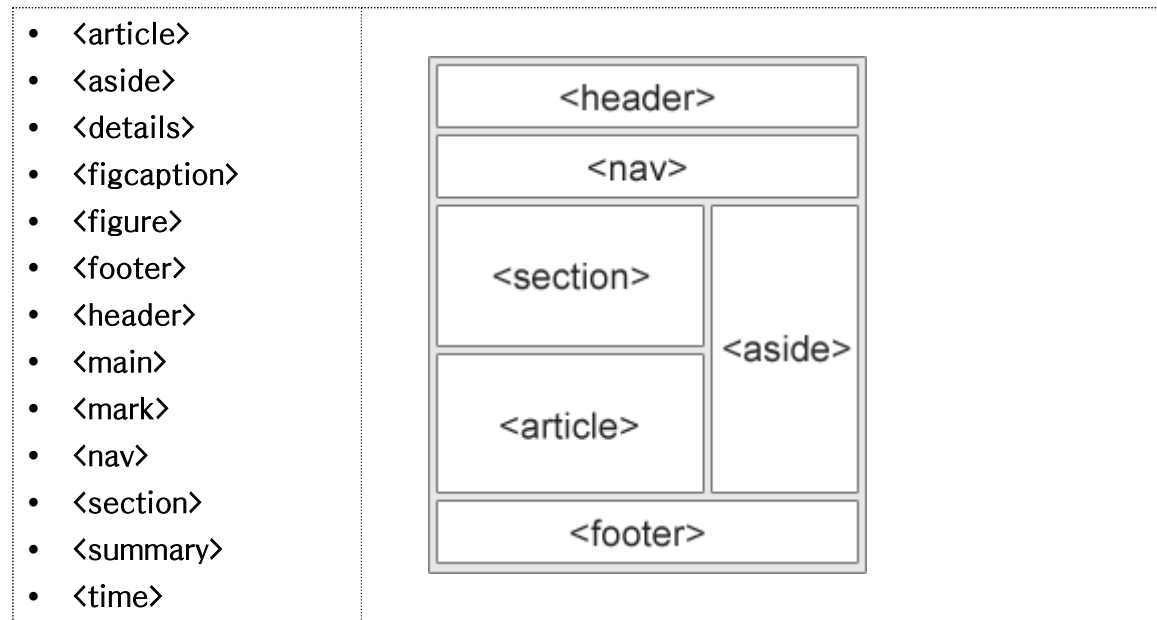
- HTML5 방식의 Semantic 구성요소

많은 웹사이트들이 컨테이너 태그를 다음과 같이 사용합니다.

```
<div id="nav"> <div class="header"> <div id="footer">
```

이것들은 "navigation", "header", "footer"를 나타냅니다.

HTML5에서는 Semantic 구성요소를 다음과 같이 정의 합니다.



- HTML5 <section> 요소

<section> 요소는 문서내에서 section(단락) 항목을 정의 합니다.

W3C의 HTML5 문서에 따르면 "섹션은 일반적으로 제목과 함께 내용의 주제별 그룹입니다." 라고 되어 있습니다.

홈 페이지는 일반적으로 소개, 내용 및 연락처 정보를 위해 섹션으로 나눌 수 있습니다.

예제)

```
<section >
```

```
    <h1 >WWF란</h1 >
```

```
    <p >The World Wide Fund for Nature (WWF) is....</p >
```

```
</section >
```

- HTML5 <article> 요소

문서나 페이지, 또는 사이트의 독립적인 부분을 구성하는 섹션을 나타냅니다. 이것은 포럼에 올라온 글, 잡지나 신문의 기사, 블로그 항목, 사용자가 제시한 의견, 상호작용적인 위젯이나 가젯, 기타 모든

독립 아이템일 수 있습니다.

Example

```
<article >
```

```
    <h1 >WWF는 무엇을 하는가?</h1 >
```

```
    <p>WWF의 사명은 우리 행성의 자연 환경이 파괴되는 것을 막고 인간이 자연과 조화롭게 살아가는 미래를 만드는 것입니다.</p >
```

```
</article >
```

- <article>에 <section> 포함 또는 <section>에 <article>의 포함 ?

<article> 문서나 페이지, 또는 사이트의 독립적인 부분을 구성하는 섹션을 나타냅니다.

<section> 요소는 문서의 섹션을 정의합니다.

인터넷에서 <article> 요소를 포함하는 <section> 요소와 <sections> 요소를 포함하는 <article> 요소가 있는 HTML 페이지를 찾을 수 있습니다.

<section> 요소를 포함하는 <section> 요소와 <article> 요소를 포함하는 <article> 요소가 있는 페이지도 있습니다.

신문 기사의 예 : 스포츠 섹션의 스포츠 기사에는 각 기사의 기술 섹션이 있을 수 있습니다.

- HTML5 <header> 요소

소개 또는 내비게이션 그룹을 나타냅니다.

<header> 요소는 문서 또는 섹션의 머리글을 지정합니다.

<header> 요소는 소개 내용의 컨테이너로 사용해야 합니다.

하나의 문서에 여러 개의 <header> 요소를 포함 할 수 있습니다.

다음 예제에서는 아티클의 머리글을 정의합니다.

Example

```
<article >
```

```
    <header >
```

```
        <h1 >WWF 무엇을 하는가?</h1 >
```

```
        <p >WWF의 임무:</p >
```

```
    </header >
```

```
    <p >WWF의 사명은 우리 행성의 자연 환경이 파괴되는 것을 막고 인간이 자연과 조화롭게 살아가는 미래를 만드는 것입니다.</p >
```

```
</article >
```

- HTML5 <footer> 요소

<footer> 요소는 문서 또는 섹션의 바닥 글을 지정합니다.

<footer> 요소는 포함 요소에 대한 정보를 포함해야 합니다.

바닥 글은 일반적으로 문서의 저자, 저작권 정보, 사용 약관 링크, 연락처 정보 등을 포함합니다. 하나의 문서에 여러 개의 <footer> 요소가 있을 수 있습니다.

예제)

```
<footer >
  <p>Posted by: Hege Refsnes </p>
  <p>Contact information: <a href ="mailto:someone@example.com">
    someone@example.com </a >.</p >
</footer >
```

- HTML5 <nav> 요소

<nav> 요소는 일련의 탐색 링크를 정의합니다.(일종의 메뉴)

문서의 모든 링크가 <nav> 요소 내에 있어야 하는 것은 아닙니다. <nav> 요소는 탐색 링크의 주요 블록에만 사용됩니다.

예제)

```
<nav>
  <a href ="/html/">HTML </a>|
  <a href ="/css/">CSS </a>|
  <a href ="/js/">JavaScript </a>|
  <a href ="/jquery/">jQuery </a>
</nav >
```

- HTML5 <aside> 요소

주위 요소의 내용과 접점을 이루는 섹션으로, 보통 사이드바 형태로 표현됩니다.

별도의 콘텐츠는 주변 콘텐츠와 관련되어야 합니다.Example

```
<p >우리 가족과 나는 이번 여름에Epcot 센터를 방문했다.</p >
```

```
<aside >
  <h4>Epcot 센터</h4 >
  <p>Epcot 센터는 플로리다 디즈니 월드에 있는 테마파크이다.</p >
</aside >
```

- HTML5 <figure> 와 <figcaption> 요소

<figcaption>의 목적은 이미지에 시각적 인 설명을 추가하는 것입니다.

HTML5에서 이미지와 캡션을 함께 <figure> 요소로 그룹화 할 수 있습니다.

예제)

<figure >

<figcaption >그림 1. - 폴핏 락(노르웨이)</figcaption >

</figure >

 요소는 이미지를 정의하고 <figcaption> 요소는 캡션을 정의합니다.



그림 1 폴핏락(노르웨이)

- 왜 Semantic 구성요소를 사용해야 하는가

HTML4를 사용하여 개발자는 머리글, 위쪽, 아래쪽, 바닥 글, 메뉴, 탐색, 기본, 컨테이너, 내용, 아티클, 사이드 바, topnav 등의 요소 스타일에 고유 한 ID / 클래스 이름을 사용했습니다.

이로 인해 검색 엔진이 정확한 웹 페이지 콘텐츠를 식별 할 수 없었습니다.

새로운 HTML5 요소 (<header> <footer> <nav> <section> <article>)를 사용하면 작업이 더욱 쉬워집니다.

W3C에 따르면 시맨틱 웹 (Semantic Web) : "응용 프로그램, 기업 및 커뮤니티에서 데이터를 공유하고 재사용 할 수 있습니다."

- HTML5에 포함된 Semantic 구성요소

다음은 HTML5의 새로운 의미 요소 알파벳순 목록입니다.

링크는 Google의 전체 HTML5 참조로 이동합니다.

Tag	설명
<article>	문서의 article 정의
<aside>	페이지 콘텐츠를 제외한 일반 콘텐츠 정의
<details>	한 덩어리의 텍스트 방향을 주위와는 별도로 표시하도록 지시
<figcaption>	사용자가 보거나 숨길 수있는 추가 세부 정보를 정의합니다.
<figure>	dialog box 나 window 정의

<u><footer></u>	<figure> 구성요소의 caption 정의
<u><header></u>	일러스트레이션, 다이어그램, 사진, 코드 등과 같은 플로우 콘텐츠를 포함
<u><main></u>	문서나 섹션의 꼬리말 정의
<u><mark></u>	문서나 섹션의 머리말 정의
<u><nav></u>	문서의 main 콘텐츠 정의
<u><section></u>	강조 텍스트 정의
<u><summary></u>	컨텍스트 메뉴와 툴바를 정의하는 것으로 command 요소, button 요소등을 이용한 커맨드 목록 정의
<u><time></u>	알려진 범위 내에서의 스칼라 측정 또는 분포 비율을 표현. 디스크 사용 현황, 쿼리 결과의 관련성, 특정 후보에 대한 투표율 등에 사용 이것을 게이지라고도 한다

HTML5 Migration

• HTML4 에서 HTML5 으로 Migration

이 장에서는 HTML4에서 HTML5로 이전하는 방법에 대해 다룹니다.

이 장에서는 원래 내용이나 구조를 손상시키지 않으면서 HTML4 페이지를 HTML5 페이지로 변환하는 방법을 보여줍니다.

동일한 레시픽을 사용하여 XHTML에서 HTML5로 마이그레이션 할 수 있습니다.

HTML4	HTML5
<div id="header">	<header>
<div id="menu">	<nav>
<div id="content">	<section>
<div class="article">	<article>
<div id="footer">	<footer>

• 일반적인 HTML4 Page

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

<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>HTML4</title>
<style>
body {
    font-family: Verdana, sans-serif ;
```

```
    font-size :0.9em ;  
}
```

```
div#header, div#footer {  
    padding :10px ;  
    color :white ;  
    background-color :black ;  
}
```

```
div#content {  
    margin :5px ;  
    padding :10px ;  
    background-color :lightgrey ;  
}
```

```
div.article {  
    margin :5px ;  
    padding :10px ;  
    background-color :white ;  
}
```

```
div#menu ul {  
    padding :0 ;  
}
```

```
div#menu ul li {  
    display :inline ;  
    margin :5px ;  
}
```

```
</style >
```

```
</head >
```

```
<body >
```

```
<div id ="header">  
    <h1 >Monday Times </h1 >  
</div >
```

```
<div id ="menu">  
    <ul >  
        <li >News </li >  
        <li >Sports </li >  
        <li >Weather </li >  
    </ul >
```

```
</div >
```

```
<div id ="content">
```

```
  <h2 >News Section </h2 >
```

```
  <div class ="article">
```

```
    <h2 >News Article </h2 >
```

```
    <p >Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
Morbi condimentum est nibh, et consectetur tortor feugiat at.</p >
```

```
  </div >
```

```
  <div class ="article">
```

```
    <h2 >News Article </h2 >
```

```
    <p >Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
Morbi condimentum est nibh, et consectetur tortor feugiat at.</p >
```

```
  </div >
```

```
</div >
```

```
<div id ="footer">
```

```
  <p >&copy; 2016 Monday Times. All rights reserved.</p >
```

```
</div >
```

```
</body >
```

```
</html >
```

- HTML5의 Doctype

doctype 변경:

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

HTML5 doctype:

```
<!DOCTYPE html >
```

HTML4 Encoding

```
<meta http-equiv ="Content-Type" content ="text/html;charset=utf-8">
```

HTML5 encoding:

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

- HTML5 Semantic 구성요소로 변경

HTML4 이전의 CSS

```
body {  
    font-family : Verdana,sans-serif ;  
    font-size : 0.9em ;  
}
```

```
div#header, div#footer {  
    padding : 10px ;  
    color : white ;  
    background-color : black ;  
}
```

```
div#content {  
    margin : 5px ;  
    padding : 10px ;  
    background-color : lightgrey ;  
}
```

```
div.article {  
    margin : 5px ;  
    padding : 10px ;  
    background-color : white ;  
}
```

```
div#menu ul {  
    padding : 0 ;  
}
```

```
div#menu ul li {  
    display : inline ;  
    margin : 5px ;  
}
```

HTML5의 CSS

```
body {  
    font-family : Verdana,sans-serif ;  
    font-size : 0.9em ;  
}
```

```
header, footer {  
    padding : 10px ;  
}
```



```

        color : white ;
        background-color : black ;
    }

    section {
        margin : 5px ;
        padding : 10px ;
        background-color : lightgrey ;
    }

    article {
        margin : 5px ;
        padding : 10px ;
        background-color : white ;
    }

    nav ul {
        padding : 0 ;
    }

    nav ul li {
        display : inline ;
        margin : 5px ;
    }

```

HTML5의 Tag 구성요소

```

<body >
<header >
    <h1 >Monday Times </h1 >
</header >

<nav >
    <ul >
        <li >News </li >
        <li >Sports </li >
        <li >Weather </li >
    </ul >
</nav >

<section >
    <h2 >News Section </h2 >
    <article >

```

```

        <h2 >News Article </h2 >
        <p >Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta
        lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p >
    </article >
    <article >
        <h2 >News Article </h2 >
        <p >Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
        Morbi condimentum est nibh, et consectetur tortor feugiat at.</p >
    </article >
</section >

<footer >
<p >&copy; 2014 Monday Times. All rights reserved.</p >
</footer >

</body >

```

- <article> <section> 와 <div>의 차이

<article> <section>과 <div> 사이에는 HTML5 표준에 혼란(차이가 없음)이 있습니다.
 HTML5 표준에서 <section> 요소는 관련된 요소의 블록으로 정의됩니다.
 <article> 요소는 관련 요소로 구성된 완전한 자체 블록으로 정의됩니다.
 <div> 요소는 자식 요소의 블록으로 정의됩니다.

어떻게 해석할까요?

위의 예제에서는 <section>을 관련 <article>의 컨테이너로 사용했습니다.
 그러나 <article>을 기사의 컨테이너로 사용할 수 도 있었습니다.
 몇 가지 다른 예가 있습니다.

<article> 예 <article> 포함:

```

<article >
<h2 >Famous Cities </h2 >
    <article >
        <h2 >London </h2 >
        <p >London is the capital city of England. It is the most populous city in the United
        Kingdom, with a metropolitan area of over 13 million inhabitants.</p >
    </article >

    <article >
        <h2 >Paris </h2 >

```

```

        <p >Paris is the capital and most populous city of France.</p >
    </article >
    <article >
        <h2 >Tokyo </h2 >
        <p >Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.</p >
    </article >
</article >

<div> 예 <article> 포함:
<article >
    <h2 >Famous Cities </h2 >
    <div class ="city">
        <h2 >London </h2 >
        <p >London is the capital city of England. It is the most populous city in the United
Kingdom,
with a metropolitan area of over 13 million inhabitants.</p >
    </div >
    <div class ="city">
        <h2 >Paris </h2 >
        <p >Paris is the capital and most populous city of France.</p >
    </div >
    <div class ="city">
        <h2 >Tokyo </h2 >
        <p >Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.</p >
    </div >
</article >

```

```

<div>, <section>, <article>의 중첩:
<article >
    <section >
        <h2 >Famous Cities </h2 >
        <div class ="city">
            <h2 >London </h2 >
            <p >London is the capital city of England. It is the most populous city in
the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p >
        </div >
        <div class ="city">
            <h2 >Paris </h2 >
            <p >Paris is the capital and most populous city of France.</p >
        </div >
        <div class ="city">

```

```

        <h2 >Tokyo </h2 >
        <p >Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.</p >
    </div >
</section >

<section >
    <h2 >Famous Countries </h2 >
    <div class ="country">
        <h2 >England </h2 >
        <p >London is the capital city of England. It is the most populous city in
the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p >
    </div >

    <div class ="country">
        <h2 >France </h2 >
        <p >Paris is the capital and most populous city of France.</p >
    </div >

    <div class ="country">
        <h2 >Japan </h2 >
        <p >Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.</p >
    </div >
</section >
</article >

```

HTML5 Style 가이드와 코딩 규약

- HTML Coding 유통성

웹 개발자는 HTML에서 사용할 코딩 스타일과 구문에 대해 종종 확신하지 못합니다. 2000 년에서 2010 년 사이에 많은 웹 개발자가 HTML을 XHTML로 변환했습니다. XHTML을 사용하여 개발자는 유효하고 "올바른 형식의"코드를 작성해야했습니다. HTML5는 코드 유효성 검사와 관련하여 좀 더 유통성이 있습니다.

- 스마트한 미래를 보장하십시오.

스타일을 일관되게 사용하면 다른 사람들이 HTML을 더 쉽게 이해할 수 있습니다. 앞으로는 XML 독자와 같은 프로그램이 HTML을 읽을 수도 있습니다.

잘 구성된 "XHTML에 가깝다"구문을 사용하면 현명 할 수 있습니다.
코드를 깔끔하게 정리하고 잘 정리 된 상태로 유지하십시오.

- 올바른 문서 유형 선언

항상 문서 형식을 문서의 첫 번째 줄로 선언하십시오.

```
<!DOCTYPE html >
```

소문자 태그와의 일관성을 유지하려면 다음처럼 사용할 수 있습니다.(그러나, 가급적 규약을 지키십시오)

```
<!doctype html >
```

- 소문자 요소 이름 사용

HTML5에서는 요소 이름에 대문자와 소문자를 섞을 수 있습니다.

소문자로 된 요소 이름을 사용하는 것이 좋습니다.

대문자와 소문자를 혼용하면 좋지 않습니다.

- * 개발자는 일반적으로 소문자 이름을 사용 합니다 (XHTML 규약중).
- * 소문자 모양 cleaner
- * 소문자는 쓰기 쉽습니다

좋지 않은 코딩:

```
<SECTION>
```

```
<p >This is a paragraph.</p >
```

```
</SECTION >
```

매우 나쁜 코딩:

```
<Section>
```

```
<p>This is a paragraph.</p >
```

```
</SECTION >
```

좋은 코딩:

```
<section >
```

```
<p >This is a paragraph.</p >
```

```
</section >
```

- 모든 HTML 구성요소는 닫혀야 합니다.

HTML5에서는 닫히지 않는 요소를 허용합니다.(예 : <p> 요소)

하지만 모든 HTML 요소를 닫는 것이 좋습니다.

나쁜코딩:

```
<section >
  <p>This is a paragraph.
  <p>This is a paragraph.
</section >
```

좋은 코딩:

```
<section >
  <p >This is a paragraph.</p >
  <p >This is a paragraph.</p >
</section >
```

- 닫지 않아도 되는 HTML 요소

HTML5는 예외적으로 닫지 않아도 되는 요소가 있습니다.

허용:

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

좋은 코딩:

```
<meta charset ="utf-8" />
```

※ XHTML 및 XML에서는 닫는 슬래시 (/)가 필요합니다.

XML 소프트웨어가 귀하의 페이지에 액세스 할 것으로 예상되는 경우 닫기 슬래시를 유지하는 것이 좋습니다.

- 소문자 Attribute 이름 사용

HTML5에서는 속성 이름에도 대문자와 소문자를 섞을 수 있습니다.

소문자 속성 이름은 다음과 같은 이유로 사용하는 것이 좋습니다.

- * 대문자와 소문자를 혼용하면 좋지 않습니다.
- * 개발자는 일반적으로 소문자 이름을 사용합니다 (예 : XHTML).
- * 소문자 모양 cleaner
- * 소문자는 쓰기 쉽다.

나쁜코드:

```
<div CLASS ="menu">
```

좋은코드:

```
<div class ="menu">
```

- Attribute Values를 따옴표로 묶기

HTML5는 따옴표없이 속성 값을 허용합니다.

다음과 같은 이유로 인해 속성 값을 인용하는 것이 좋습니다.

- * 대문자와 소문자를 혼용하면 좋지 않습니다.
- * 따옴표로 묶인 값을 읽기 쉽습니다.
- * 값에 공백이 포함되어 있으면 반드시 따옴표를 사용해야 합니다. Very bad:

다음 코드는 값에 공백이 포함되어 있기 때문에 정상 작동하지 않습니다.

```
<table class = table striped >
```

나쁜코드:

```
<table class = striped >
```

좋은코드:

```
<table class ="striped">
```

- Image 속성

항상 이미지에 "alt"속성을 추가하십시오. 이 속성은 이미지가 표시되지 않을 때 중요합니다.

또한 항상 이미지 너비와 높이를 정의하십시오. 로드하기 전에 브라우저가 이미지 공간을 예약 할 수 있기 때문에 깜박임 현상이 줄어듭니다.

나쁜코드:

```
<img src ="html5.gif">
```

Good:

```
<img src ="html5.gif" alt ="HTML5" style ="width:128px;height:128px">
```

- Spaces and Equal Signs

HTML5는 등호 주위에 공백을 허용합니다. 그러나 공간이 적은 것은 읽기가 쉬우며 엔티티를 더 잘 그룹화합니다.

나쁜코드:

```
<link rel = "stylesheet" href = "styles.css">
```

좋은 코드:

```
<link rel="stylesheet" href="styles.css">
```

- 긴코드 피하기

HTML 편집기를 사용할 때 HTML 코드를 읽으려면 좌우로 스크롤하는 것이 불편합니다.
80 자 이상의 코드 행을 피하십시오.

- 빈줄과 들여쓰기

이유없이 빈 줄을 추가하지 마십시오.
가독성을 위해 빈 줄을 추가하여 큰 코드 블록이나 논리 코드 블록을 분리하십시오.
가독성을 위해 두 개의 들여 쓰기 공백을 추가하십시오. 탭 키를 사용하지 마십시오.
불필요한 빈 줄과 들여 쓰기를 사용하지 마십시오. 매번 들여 쓰기 할 필요는 없습니다.
element:

불필요한 빈줄 추가한 예:

```
<body >
```

```
    <h1 >Famous Cities </h1 >
```

```
    <h2 >Tokyo </h2 >
```

```
    <p >
```

```
        Tokyo is the capital of Japan, the center of the Greater Tokyo Area,  
        and the most populous metropolitan area in the world.
```

```
        It is the seat of the Japanese government and the Imperial Palace,  
        and the home of the Japanese Imperial Family.
```

```
    </p >
```

```
</body >
```

좋은 코드:

```
<body >
```

```
<h1 >Famous Cities </h1 >
```

```
<h2 >Tokyo </h2 >
```

```
<p >Tokyo is the capital of Japan, the center of the Greater Tokyo Area,  
and the most populous metropolitan area in the world.
```

```
It is the seat of the Japanese government and the Imperial Palace,  
and the home of the Japanese Imperial Family.</p >
```

```
</body >
```


Table Example:

```
<table >
  <tr >
    <th >Name </th >
    <th >Description </th >
  </tr >
  <tr >
    <td >A </td >
    <td >Description of A </td >
  </tr >
  <tr >
    <td >B </td >
    <td >Description of B </td >
  </tr >
</table >
```

List Example:

```
<ol >
  <li >London </li >
  <li >Paris </li >
  <li >Tokyo </li >
</ol >
```

- **<html> 과 <body>의 생략**

HTML5 표준에서는 <html> 태그와 <body> 태그를 생략 할 수 있습니다.
다음 코드는 HTML5로 확인합니다.

```
<!DOCTYPE html >
<head >
  <title >Page Title </title >
</head >
```

```
<h1>This is a heading </h1 >
<p>This is a paragraph.</p >
```

하지만, <html> 및 <body> 태그는 생략하지 않는 것이 좋습니다.

<html> 요소는 문서 루트입니다. 페이지를 지정하는 데 권장되는 장소입니다.language:

언어설정:

```
<!DOCTYPE html >
<html lang ="en-US">
```

- * 접근성 응용 프로그램 (화면 판독기) 및 검색 엔진에 언어를 선언하는 것이 중요합니다.Omitting
- * <html> 이나 <body>는 DOM 및 XML 소프트웨어를 손상시킬 수 있습니다.
- * <body>를 생략하면 구형 브라우저 (IE9)에서 오류가 발생할 수 있습니다.

- <head>의 생략

HTML5 표준에서는 <head> 태그를 생략 할 수도 있습니다.
기본적으로 브라우저는 <body> 앞에있는 모든 요소를 기본 <head> 요소에 추가합니다.
<head> 태그를 생략하면 HTML의 복잡성을 줄일 수 있습니다.

```
<!DOCTYPE html >
<html >
<title >Page Title </title >

<body >
  <h1>This is a heading </h1 >
  <p>This is a paragraph.</p >
</body >

</html >
```

<head> 태그는 생략하지 않는 것이 좋습니다.
태그를 생략하면 웹 개발자에게는 익숙하지 않습니다. 가이드 라인을 수립하는데 시간이 필요합니다.

- Meta Data

HTML5에는 <title> 요소가 필요합니다. 제목을 가능한 한 의미있게 만드십시오.
<title> HTML5 구문 및 코딩 스타일 </title>
적절한 해석을 하고 검색 엔진 색인을 수정하려면 언어와 문자 인코딩을 문서에서 가능한 한 빨리 정의해야 합니다.

```
<!DOCTYPE html >
<html lang ="en-US">
<head >
  <meta charset ="UTF-8">
  <title >HTML5 Syntax and Coding Style </title >
</head >
```

- Viewport 설정

HTML5는 웹 디자이너가 <meta> 태그를 통해 뷰포트를 제어 할 수 있게 하는 메소드를 도입했습니다. 뷰포트는 웹 페이지의 사용자가 볼 수있는 영역입니다. 그것은 장치에 따라 다르며 휴대 전화에서는 컴퓨터 화면보다 작습니다.

모든 웹 페이지에 다음과 같은 <meta> 뷰포트 요소를 포함해야 합니다.

```
<meta name = "viewport" content = "width=device-width, initial-scale=1.0">
```

<meta> viewport 요소는 페이지의 크기와 배율을 제어하는 방법에 대한 브라우저 지침을 제공합니다.

width = device-width 부분은 페이지의 너비를 장치의 화면 너비에 따라 설정합니다 (장치에 따라 다름).

initial-scale = 1.0 부분은 브라우저가 페이지를 처음 로드 할 때 초기 줌 레벨을 설정합니다.

다음은 뷰포트 메타 태그가 없는 웹 페이지와 뷰포트 메타 태그가 있는 동일한 웹 페이지의 예입니다.

도움말 : 휴대 전화 또는 태블릿으로 이 페이지를 탐색하는 경우 아래의 두 링크를 클릭하여 차이를 확인할 수 있습니다.



View Port meta tag를 사용하지 않은 경우



Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed diam nonummy nibh
euismod tincidunt ut laoreet dolore magna
aliquam erat volutpat. Ut wisi enim ad minim
veniam, quis nostrud exerci tation ullamcorper
suscipit lobortis nisl ut aliquip ex ea commodo
consequat. Duis autem vel eum iriure dolor in
hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla
facilisis at vero eros et accumsan et iusto odio
dignissim qui blandit praesent luptatum zzril
delenit augue duis dolore te feugait nulla
facilisi. Nam liber tempor cum soluta nobis
eleifend ntion conone nihil inperdiet domine...

viewport meta tag를 사용한 경우

- HTML 주석

한줄 짜리 은 주석:

```
<!-- This is a comment -->
```

여러줄 주석:

```
<!--
```

```
This is a long comment example. This is a long comment example.
```

```
This is a long comment example. This is a long comment example.
```

```
-->
```

긴줄 주석은 2칸의 들여쓰기를 하여야 한다.

- Style Sheets

스타일 시트에 연결하기 위해 간단한 구문을 사용하십시오 (type 속성은 필요하지 않습니다).

```
<link rel="stylesheet" href="styles.css">
```

짧은 규칙은 다음과 같이 한 줄에 압축하여 작성할 수 있습니다.

```
p.intro {font-family : Verdana ; font-size :16em ;}
```

긴 규칙은 여러 행에 작성해야 합니다.

```
body {
```

```
background-color :lightgrey ;
```

```
font-family : "Arial Black", Helvetica, sans-serif ;
font-size : 16em ;
color : black ;
}
```

Selector 와 동일한 줄에 여는 괄호를 배치하십시오.

여는 괄호 앞에 하나의 공간을 사용하십시오.

두 개의 들어 쓰기 공간 사용하십시오

마지막 속성을 포함하여 각 속성 - 값 쌍 다음에 **세미콜론**을 사용하십시오.

값에 공백이 포함 된 경우에만 값을 따옴표로 묶으십시오.

공백이없는 새 행에 닫는 괄호를 배치하십시오.

80 자 이상의 행 피하십시오

- **HTML에서 JavaScript 읽기**

외부 스크립트를로드하는 데 간단한 구문을 사용합니다 (type 속성은 필요하지 않음).

```
<script src ="myscript.js">
```

- **JavaScript에서 HTML 요소에 접근**

"깔끔하지 않은"HTML 스타일을 사용하면 JavaScript 오류가 발생할 수 있습니다.

이 두 가지 JavaScript 문은 서로 다른 결과를 생성합니다.

```
var obj = getElementByld("Demo")
```

```
var obj = getElementByld("demo")
```

- **소문자 파일이름 사용**

일부 웹 서버 (Apache, Unix)는 파일 이름에 대소문자를 구분합니다. "london.jpg"는 "London.jpg"로 액세스 할 수 없습니다.

다른 웹 서버 (Microsoft, IIS)는 대소문자를 구분하지 않습니다. "london.jpg"는 "London.jpg"또는 "london.jpg"로 액세스 할 수 있습니다.

대문자와 소문자를 혼용하면 일관성이 있어야합니다.

대소문자를 구분하지 않는 서버에서 대소문자를 구분하는 서버로 이동하면 작은 오류라도 웹이 손상됩니다!

이러한 문제를 피하려면 항상 소문자 파일 이름을 사용하십시오.

- **File 확장자**

HTML 파일은 .html 또는 .htm 확장자를 가져야합니다.

CSS 파일의 확장자는 .css 여야합니다.

JavaScript 파일의 확장자는 .js 여야합니다.

- **.htm과 .html의 차이점**

.htm과 .html 확장자에는 차이가 없습니다. 둘 다 모든 웹 브라우저 또는 웹 서버에서 HTML로 처리됩니다.

문화적 차이는 다음과 같습니다.

.htm은 시스템이 확장자를 3 자로 제한 한 초기 DOS 시스템의 “추억”입니다.

.html은 제한이 없는 유닉스 운영체제의 "추억"입니다.

- **기교적 차이**

URL이 파일 이름 (예 : <https://www.w3schools.com/css/>)을 지정하지 않으면 서버는 기본 파일 이름을 반환합니다. 일반적인 기본 파일 이름은 index.html, index.htm, default.html 및 default.htm입니다.

서버가 기본 파일 이름으로 "index.html"로만 구성되어 있으면 파일의 이름은 "index.htm"이 아니라 "index.html"이어야합니다.

그러나 서버는 둘 이상의 기본 파일 이름으로 구성 될 수 있으며 일반적으로 필요한만큼 많은 기본 파일 이름을 설정할 수 있습니다.

어쨌든 HTML 파일의 전체 확장자는 .html입니다.