HTML Tutorial

Part 4.

41. HTML Layout Elements and Techniques

- 웹사이트는 종종 잡지나 신문 사이트와 같은 여러 열에 콘텐츠를 배열하여 표시합니다.

◆ HTML 레이아웃 요소

- HTML에는 웹 페이지의 각각의 부분을 정의하는 몇 가지 의미론적 요소(semantic elements)가 있습니다.

  - HTML5 시맨틱 요소

<header>	- 문서 또는 섹션의 헤더를 정의합니다.
<nav>	- 탐색 링크 세트를 정의합니다.
<section>	- 문서의 섹션을 정의합니다.
<article>	- 독립적인 콘텐츠를 정의합니다.
<aside>	- 콘텐츠 이외의 콘텐츠 정의(사이드바 등)
<footer>	- 문서 또는 섹션의 바닥글을 정의합니다.
<details>	- 사용자가 필요에 따라 열고 닫을 수 있는 추가 세부 정보 정의
<summary>	- <details> 요소의 제목을 정의합니다.

◆ HTML 레이아웃 기법

- 여러 열 레이아웃을 만드는 네 가지 기술이 있습니다. 각 기술에는 장단점이 있습니다.

  - CSS 프레임워크(Framework)

- CSS 부동 속성(Float Property)
- CSS 플렉스박스(Flex Box)
- CSS 그리드(Grid)

◆ CSS 프레임워크

- 레이아웃을 빠르게 만들고 싶다면 CSS 또는 Bootstrap 과 같은 CSS 프레임워크를 사용할 수 있습니다 .

◆ CSS Float 레이아웃

- CSS float속성을 사용하여 전체 웹 레이아웃을 수행하는 것이 일반적입니다.
- Float은 배우기 쉽습니다 . float및 clear 속성이 작동하는 방식만 기억하면 됩니다.
- 단점 : Floating 요소는 문서 흐름에 연결되어 유연성에 해를 끼칠 수 있습니다.
- 아래의 예는 header와 그 아래 두 개의 column, 그리고 맨 아래 footer를 생성합니다.
- 작은 화면이나 화면을 줄이면 각 column이 stack 형식으로 쌓이는 모양으로 표시됩니다.

예시

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Template</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
```

```
  box-sizing: border-box;
}

body {
  font-family: Arial, Helvetica, sans-serif;
}

/* Style the header */
header {
  background-color: #666;
  padding: 30px;
  text-align: center;
  font-size: 35px;
  color: white;
}

/* Create two columns/boxes that floats next to each other */
nav {
  float: left;
  width: 30%;
  height: 300px; /* only for demonstration, should be removed */
  background: #ccc;
  padding: 20px;
}

/* Style the list inside the menu */
nav ul {
  list-style-type: none;
```

```
  padding: 0;
}

article {
  float: left;
  padding: 20px;
  width: 70%;
  background-color: #f1f1f1;
  height: 300px; /* only for demonstration, should be removed */
}

/* Clear floats after the columns */
section::after {
  content: "";
  display: table;
  clear: both;
}

/* Style the footer */
footer {
  background-color: #777;
  padding: 10px;
  text-align: center;
  color: white;
}

/* Responsive layout - makes the two columns/boxes stack on top of each other instead of next to each other,
on small screens */
```

```
@media (max-width: 600px) {
  nav, article {
    width: 100%;
    height: auto;
  }
}
</style>
</head>
<body>

<header>
  <h2>Cities</h2>
</header>

<section>
  <nav>
    <ul>
      <li><a href="#">London</a></li>
      <li><a href="#">Paris</a></li>
      <li><a href="#">Tokyo</a></li>
    </ul>
  </nav>

  <article>
    <h1>London</h1>
    <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>
    <p>Standing on the River Thames, London has been a major settlement for two millennia, its history going
```

```
back to its founding by the Romans, who named it Londinium.</p>
  </article>
</section>

<footer>
  <p>Footer</p>
</footer>

</body>
</html>
```

◆ CSS Flexbox Layout

- flexbox를 사용하면 페이지 레이아웃이 다양한 화면 크기와 다양한 디스플레이 장치를 수용해야 할 때 요소가 예측 가능하게 작동합니다.
- CSS 강좌에서 flexbox에 대해 자세히 배우게 됩니다.

예시

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Template</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
```

```
  box-sizing: border-box;
}

body {
  font-family: Arial, Helvetica, sans-serif;
}

/* Style the header */
header {
  background-color: #666;
  padding: 30px;
  text-align: center;
  font-size: 35px;
  color: white;
}

/* Container for flexboxes */
section {
  display: -webkit-flex;
  display: flex;
}

/* Style the navigation menu */
nav {
  -webkit-flex: 1;
  -ms-flex: 1;
  flex: 1;
  background: #ccc;
```

```
  padding: 20px;
}

/* Style the list inside the menu */
nav ul {
  list-style-type: none;
  padding: 0;
}

/* Style the content */
article {
  -webkit-flex: 3;
  -ms-flex: 3;
  flex: 3;
  background-color: #f1f1f1;
  padding: 10px;
}

/* Style the footer */
footer {
  background-color: #777;
  padding: 10px;
  text-align: center;
  color: white;
}

/* Responsive layout - makes the menu and the content (inside the section) sit on top of each other instead of
next to each other */
```

```
@media (max-width: 600px) {
  section {
    -webkit-flex-direction: column;
    flex-direction: column;
  }
}
</style>
</head>
<body>

<h2>CSS Layout Flexbox</h2>
<p>In this example, we have created a header, two columns/boxes and a footer. On smaller screens, the columns will stack on top of each other.</p>
<p>Resize the browser window to see the responsive effect.</p>
<p><strong>Note:</strong> Flexbox is not supported in Internet Explorer 10 and earlier versions.</p>

<header>
  <h2>Cities</h2>
</header>

<section>
  <nav>
    <ul>
      <li><a href="#">London</a></li>
      <li><a href="#">Paris</a></li>
      <li><a href="#">Tokyo</a></li>
    </ul>
  </nav>
```

```
  <article>
    <h1>London</h1>
    <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>
    <p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p>
  </article>
</section>

<footer>
  <p>Footer</p>
</footer>

</body>
</html>
```

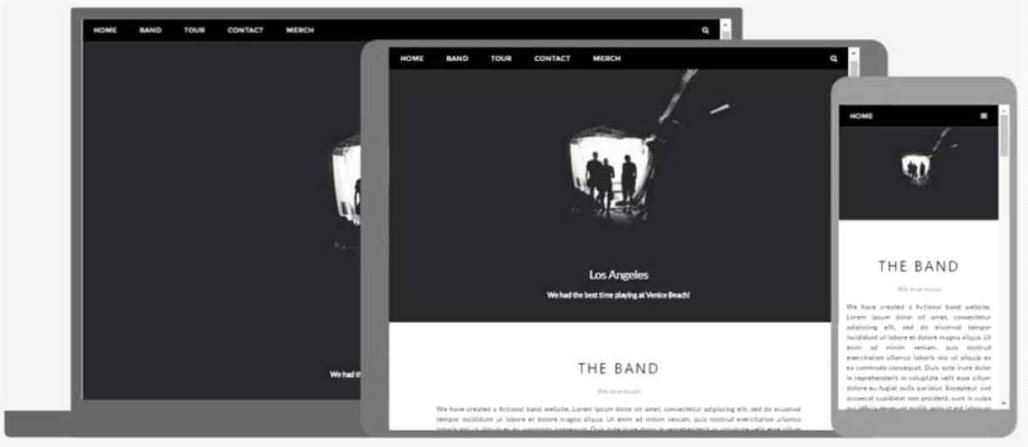
◆ CSS 그리드 레이아웃

- CSS Grid Layout Module은 행과 열이 있는 그리드 기반 레이아웃 시스템을 제공하므로 float과 positioning을 사용하지 않고도 웹 페이지를 더 쉽게 디자인할 수 있습니다.

* 참고 : CSS 와 Bootstrap 강좌에서 자세히 배우게 됩니다.

42. HTML Responsive Web Design

- 반응형 웹 디자인은 모든 장치에서 보기 좋은 웹 페이지를 만드는 것입니다!
- 반응형 웹 디자인은 다양한 화면 크기와 뷰포트에 맞게 자동으로 조정됩니다.



◆ 반응형 웹 디자인이란?

- 반응형 웹 디자인은 HTML 및 CSS를 사용하여 웹 사이트의 크기를 자동으로 조정, 숨기기, 축소 또는 확대하여 모든 장치(데스크톱, 태블릿 및 휴대폰)에서 보기 좋게 만드는 것입니다.

◆ Viewport 설정

- 반응형 웹사이트를 만들려면 모든 웹페이지에 다음의 <meta> 태그를 추가 하세요.

예시

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

- 이렇게 하면 페이지의 표시 영역이 설정되어 페이지의 크기와 배율을 제어하는 방법에 대한 브라우저 지침이 제공됩니다.
- 다음은 뷰포트 메타 태그가 없는 웹 페이지 와 뷰포트 메타 태그가 있는 동일한 웹 페이지의 예 입니다 .

Without the viewport meta tag:

With the viewport meta tag:

◆ 반응형 이미지

- 반응형 이미지는 모든 브라우저 크기에 맞게 잘 확장되는 이미지입니다.
- CSS width 속성을 사용하여 width가 100%로 설정되면 이미지가 반응하고 확장 및 축소됩니다.

예시

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

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


<h2>Responsive Image</h2>
<p>When the CSS width property is set in a percentage value, the image will scale up and down when resizing the browser window. Resize the browser window to see the effect.</p>


<img src="images/img_girl.jpg" style="width:100%;">


</body>
</html>
```

- 위의 예에서 이미지는 원래 크기보다 크게 확대될 수 있습니다. 많은 경우에 더 나은 솔루션은 max-width속성을 대신 사용하는 것입니다.

◆ 최대 너비 속성 사용

- max-width속성이 100%로 설정된 경우 이미지는 필요한 경우 축소되지만 원래 크기보다 크게 확대되지는 않습니다.

예시

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
```

```
<h2>Responsive Image</h2>
<p>"max-width:100%" prevents the image from getting bigger than its original size. However, if you make the browser window smaller, the image will still scale down.</p>
<p>Resize the browser window to see the effect.</p>

<img src="images/img_girl.jpg" style="max-width:100%;height:auto;">

</body>
</html>
```

◆ 브라우저 너비에 따라 다른 이미지 표시

- HTML <picture>요소를 사용하면 브라우저 창 크기에 따라 다른 이미지를 정의할 수 있습니다.
- 브라우저 창의 크기를 조정하여 너비에 따라 아래 이미지가 어떻게 변하는지 확인합니다.

예시

```
<picture>
  <source srcset="images/img_smallflower.jpg" media="(max-width: 600px)">
  <source srcset="images/img_flowers.jpg" media="(max-width: 900px)">
  <source srcset="images/flowers.jpg">
  <img src="images/img_smallflower.jpg" alt="Flowers">
</picture>
```

◆ 반응형 텍스트 크기

- 텍스트 크기는 "뷰포트 너비"를 의미하는 "vw" 단위로 설정할 수 있습니다.
- 그렇게 하면 텍스트 크기가 브라우저 창의 크기를 따릅니다.

예시

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>

<h1 style="font-size : 10vw;">Responsive Text</h1>

<p style="font-size : 5vw;">Resize the browser window to see how the text size scales.</p>

<p style="font-size : 5vw;">Use the "vw" unit when sizing the text. 10vw will set the size to 10% of the
viewport width.</p>

<p>Viewport is the browser window size. 1vw = 1% of viewport width. If the viewport is 50cm wide, 1vw is
0.5cm.</p>

</body>
</html>
```

- 뷰포트는 브라우저 창 크기입니다. 1vw = 뷰포트 너비의 1%. 뷰포트의 너비가 50cm인 경우 1vw는 0.5cm입니다.

◆ 미디어 쿼리

- 텍스트와 이미지의 크기를 조정하는 것 외에도 반응형 웹 페이지에서 미디어 쿼리를 사용하는 것도 일반적입니다.
- 미디어 쿼리를 사용하면 다양한 브라우저 크기에 대해 완전히 다른 스타일을 정의할 수 있습니다.
- 예: 브라우저 창의 크기를 조정하여 아래의 세 div 요소가 큰 화면에서는 가로로 표시되고 작은 화면에서는 세로로 쌓이도록 합니다.

예시

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
* {
  box-sizing: border-box;
}

.left {
  background-color: #2196F3;
  padding: 20px;
  float: left;
  width: 20%; /* The width is 20%, by default */
}

.main {
  background-color: #f1f1f1;
  padding: 20px;
```

```
  float: left;
  width: 60%; /* The width is 60%, by default */
}

.right {
  background-color: #04AA6D;
  padding: 20px;
  float: left;
  width: 20%; /* The width is 20%, by default */
}

/* Use a media query to add a break point at 800px: */
@media screen and (max-width: 800px) {
  .left, .main, .right {
    width: 100%; /* The width is 100%, when the viewport is 800px or smaller */
  }
}
</style>
</head>
<body>

<h2>Media Queries</h2>
<p>Resize the browser window.</p>

<p>Make sure you reach the breakpoint at 800px when resizing this frame.</p>

<div class="left">
  <p>Left Menu</p>
```

```
</div>

<div class="main">
  <p>Main Content</p>
</div>

<div class="right">
  <p>Right Content</p>
</div>

</body>
</html>
```

◆ 반응형 웹 페이지 - 전체 예

- 반응형 웹 페이지는 큰 데스크톱 화면과 작은 휴대전화에서 보기 좋게 표시되어야 합니다.

예

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
* {
  box-sizing: border-box;
}
```

```
.menu {
  float: left;
  width: 20%;
  text-align: center;
}

.menu a {
  background-color: #e5e5e5;
  padding: 8px;
  margin-top: 7px;
  display: block;
  width: 100%;
  color: black;
}

.main {
  float: left;
  width: 60%;
  padding: 0 20px;
}

.right {
  background-color: #e5e5e5;
  float: left;
  width: 20%;
  padding: 15px;
  margin-top: 7px;
```

```
  text-align: center;
}

@media only screen and (max-width: 620px) {
  /* For mobile phones: */
  .menu, .main, .right {
    width: 100%;
  }
}
</style>
</head>
<body style="font-family:Verdana;color:#aaaaaa;">

<div style="background-color:#e5e5e5;padding:15px;text-align:center;">
  <h1>Hello World</h1>
</div>

<div style="overflow:auto">
  <div class="menu">
    <a href="#">Link 1</a>
    <a href="#">Link 2</a>
    <a href="#">Link 3</a>
    <a href="#">Link 4</a>
  </div>

  <div class="main">
    <h2>Lorum Ipsum</h2>
    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut
```

```
laoreet dolore magna aliquam erat volutpat.</p>
  </div>

  <div class="right">
    <h2>About</h2>
    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit.</p>
  </div>
</div>

<div    style="background-color:#e5e5e5;text-align:center;padding:10px;margin-top:7px;">©    copyright    IT
Acacemy</div>

</body>
</html>
```

◆ 반응형 웹 디자인 - 프레임워크

- 모든 인기 있는 CSS 프레임워크는 반응형 디자인을 제공합니다.
- 대표적으로 Bootstrap이 있습니다. Bootstrap 강좌에서 상세히 학습할 수 있습니다.

43. HTML Computer Code Elements

- HTML에는 사용자 입력 및 컴퓨터 코드를 정의하기 위한 여러 요소가 포함되어 있습니다.

◆ 키보드 입력용 HTML <kbd>

- HTML <kbd>요소는 키보드 입력을 정의하는 데 사용됩니다. 내부 콘텐츠는 브라우저의 기본 고정 폭 글꼴로 표시됩니다.

예시
문서에서 일부 텍스트를 키보드 입력으로 정의:

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

<h2>The kbd Element</h2>

<p>The kbd element is used to define keyboard input:</p>

<p>Save the document by pressing <kbd>Ctrl + S</kbd></p>

</body>
</html>
```

◆ 프로그램 출력용 HTML <samp>

- HTML <samp>요소는 컴퓨터 프로그램의 샘플 출력을 정의하는 데 사용됩니다. 내부 콘텐츠는 브라우저의 기본 고정 폭 글꼴로 표시됩니다.

예시

문서에 있는 컴퓨터 프로그램의 샘플 출력으로 일부 텍스트를 정의합니다.

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

<h2>The samp Element</h2>

<p>The samp element is used to define sample output from a computer program.</p>

<p>Message from my computer:</p>
<p><samp>File not found.<br>Press F1 to continue</samp></p>

</body>
</html>
```

◆ 컴퓨터 코드용 HTML <code>

- HTML <code>요소는 컴퓨터 코드를 정의하는 데 사용됩니다. 내부 콘텐츠는 브라우저의 기본 고정 폭 글꼴로 표시됩니다.

예시

문서에서 일부 텍스트를 컴퓨터 코드로 정의:

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

<h2>The code Element</h2>
<p>Programming code example:</p>

<code>
x = 5;
y = 6;
z = x + y;
</code>

</body>
</html>
```

- 참고 : <code>요소가 추가 공백과 줄 바꿈을 보존하지 않습니다. 이 문제를 해결하려면 <code>요소 내부에 <pre>요소를 넣을 수 있습니다 .

예시

```
<!DOCTYPE html>
```

```
<html>
<body>

<p>The code element does not preserve whitespace and line-breaks.</p>
<p>To fix this, you can put the code element inside a pre element:</p>

<pre>
<code>
x = 5;
y = 6;
z = x + y;
</code>
</pre>

</body>
</html>
```

◆ 변수용 HTML <var>

- HTML <var>요소는 프로그래밍이나 수학적 표현에서 변수를 정의하는 데 사용됩니다. 내부 콘텐츠는 일반적으로 기울임꼴로 표시됩니다.

예시

문서에서 일부 텍스트를 변수로 정의합니다.

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

```
<body>


<h2>The var Element</h2>


<p>The area of a triangle is: 1/2 x <var>b</var> x <var>h</var>, where <var>b</var> is the base, and
<var>h</var> is the vertical height.</p>


</body>
</html>
```

44. HTML Semantic Elements

◆ 시맨틱 요소란?

- 의미론적 요소는 브라우저와 개발자 모두에게 의미를 명확하게 설명합니다.
- 의미 없는 요소의 예 : 및 - 내용에 대해 아무 것도 알려주지 않습니다.
- 의미 요소의 예 : <form>, <table>및 <article>- 내용을 명확하게 정의합니다.

◆ HTML의 시맨틱 요소

- 많은 웹 사이트에는 navigation, header 및 footer을 나타내는   와 같은 HTML 코드가 포함되어 있습니다.
- HTML에는 웹 페이지의 다른 부분을 정의하는 데 사용할 수 있는 몇 가지 의미론적 요소가 있습니다.

- <article>, <aside>
- <details>
- <figcaption>, <figure>, <footer>
- <header>
- <main>, <mark>
- <nav>
- <section>, <summary>
- <time>

◆ HTML <section> 요소

- <section>요소는 문서의 섹션을 정의합니다.
- W3C의 HTML 문서에 따르면 "섹션은 일반적으로 제목이 있는 주제별 콘텐츠 그룹입니다."
- <section>요소를 사용할 수 있는 위치의 예 :

  - ✓ Chapters
  - ✓ Introduction
  - ✓ News items
  - ✓ Contact information

- 웹 페이지는 일반적으로 소개, 콘텐츠 및 연락처 정보 섹션으로 분할될 수 있습니다.

예시

문서의 두 섹션:

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

<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.</p>
</section>

<section>
```

```
  <h1>WWF's Panda symbol</h1>
  <p>The Panda has become the symbol of WWF. The well-known panda logo of WWF originated from a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo in the same year of the establishment of WWF.</p>
</section>


</body>
</html>
```

◆ HTML <article> 요소

- <article>요소는 독립적인, 자체적으로 어떤 내용을 포함하는 부분을 정의하는데 사용됩니다.
- 예를 들어, 신문 기사는 그 자체로 의미가 있어야 하며 웹 사이트의 나머지 부분과 독립적으로 배포할 수 있어야 합니다.
- <article>요소를 사용할 수 있는 위치의 예 :

  - ✓ Forum posts
  - ✓ Blog posts
  - ✓ User comments
  - ✓ Product cards
  - ✓ Newspaper articles

예시

독립적이고 독립적인 콘텐츠가 포함된 세 개의 기사:

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

```
<h1>The article element</h1>

<article>
  <h2>Google Chrome</h2>
  <p>Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today!</p>
</article>

<article>
  <h2>Mozilla Firefox</h2>
  <p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018.</p>
</article>

<article>
  <h2>Microsoft Edge</h2>
  <p>Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft Edge replaced Internet Explorer.</p>
</article>

</body>
</html>
```

실시 예 2

CSS를 사용하여 <article> 요소 스타일 지정:

```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
.all-browsers {
  margin: 0;
  padding: 5px;
  background-color: lightgray;
}

.all-browsers > h1, .browser {
  margin: 10px;
  padding: 5px;
}

.browser {
  background: white;
}

.browser > h2, p {
  margin: 4px;
  font-size: 90%;
}
</style>
</head>
<body>

<article class="all-browsers">
  <h1>Most Popular Browsers</h1>
```

```
  <article class="browser">
    <h2>Google Chrome</h2>
    <p>Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today!</p>
  </article>
  <article class="browser">
    <h2>Mozilla Firefox</h2>
    <p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018.</p>
  </article>
  <article class="browser">
    <h2>Microsoft Edge</h2>
    <p>Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft Edge replaced Internet Explorer.</p>
  </article>
</article>


</body>
</html>
```

◆ <section>에 <article>을 중첩하거나 혹은 그 반대로?

- <article> 요소는 독립적인 자체에 포함 된 내용을 지정합니다.
- <section>요소는 문서의 섹션을 정의합니다.
- 따라서 HTML 페이지에는 <article> 요소를 포함하는 <section> 요소와 <section> 요소를 포함하는 <article> 요소를 사용할 수 있습니다.

◆ HTML <header> 요소

- <header>요소는 소개 내용이나 탐색 링크의 집합에 대한 컨테이너를 나타냅니다.
- <header>요소는 일반적으로 다음을 포함 :

  - ✓ 하나 이상의 headings 요소(<h1> - <h6>)
  - ✓ 로고 또는 아이콘
  - ✓ 저작권 정보 등

- 참고: 하나의 HTML 문서에 여러 <header> 요소가 있을 수 있습니다 . 그러나 <header> 내에 배치 할 수없는 요소는 <footer>, <address>또는 다른 <header>요소입니다.

예시

<article>의 헤더:

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

<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural environment, and build a future in which humans live in harmony with nature.</p>
</article>
```

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

◆ HTML <footer> 요소

- <footer>요소는 문서나 구역의 바닥 글을 정의합니다.
- <footer>요소는 일반적으로 다음을 포함 :

  - ✓ 저작권 정보
  - ✓ 저작권 정보
  - ✓ 연락처 정보
  - ✓ 사이트맵
  - ✓ 맨 위로 링크로 돌아가기
  - ✓ 관련된 문서

- 일반적으로 한 document에 여러개의 <footer>요소가 있을 수 있습니다.

예시
문서의 바닥글 섹션:

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

<footer>
  <p>Author: Hege Refsnes</p>
```

```
  <p><a href="mailto:hege@example.com">hege@example.com</a></p>
</footer>

</body>
</html>
```

◆ HTML <nav> 요소

- <nav>요소는 탐색 링크의 집합을 정의합니다.
- 문서의 모든 링크가 <nav>요소 내부에 있어야 하는 것은 아닙니다 . <nav> 요소는 탐색 링크의 주요 블록을위한 것입니다.

예시

탐색 링크 세트:

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

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

</body>
</html>
```

◆ HTML <aside> 요소

- <aside> 요소는 sidebar와 같이 배치된 내용 외에 일부 내용을 정의합니다.
- <aside> 콘텐츠는 주변 콘텐츠와 간접적으로 관련되어야 한다.

예시

배치된 콘텐츠 외에 일부 콘텐츠를 표시합니다.

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

<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>

<aside>
  <h4>Epcot Center</h4>
  <p>Epcot is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.</p>
</aside>

</body>
</html>
```

실시예 2

CSS를 사용하여 <aside> 요소 스타일 지정:

```
<!DOCTYPE html>
<html>
<head>
<style>
aside {
  width: 30%;
  padding-left: 15px;
  margin-left: 15px;
  float: right;
  font-style: italic;
  background-color: lightgray;
}
</style>
</head>
<body>


<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>


<aside>
<p>The Epcot center is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.</p>
</aside>


<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>
<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>
```

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

◆ HTML <Figure> 및 <figcaption> 요소

- <figure> 태그는 그림, 다이어그램, 사진, 코드 목록 등과 같은 자체 포함 콘텐츠를 지정합니다.
- <figcaption> 태그는 <figure> 요소에 대한 캡션을 정의한다. <figcaption> 요소는 <figure> 요소의 첫 번째 또는 마지막 자식 요소로 배치할 수 있다.
- <img> 요소는 실제 영상/일러스트레이션을 정의합니다.

예시

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


<h2>Places to Visit</h2>


<p>Puglia's most famous sight is the unique conical houses (Trulli) found in the area around Alberobello, a
declared UNESCO World Heritage Site.</p>


<figure>
  <img src="images/pic_trulli.jpg" alt="Trulli" style="width:100%">
  <figcaption>Fig.1 - Trulli, Puglia, Italy.</figcaption>
</figure>


</body>
```

```
</html>
```

◆ 왜 시맨틱 요소인가?

- W3C에 따르면 "시맨틱 웹을 사용하면 애플리케이션, 기업 및 커뮤니티에서 데이터를 공유하고 재사용할 수 있습니다."

◆ HTML의 시맨틱 요소

- 다음은 HTML에서 의미론적 요소의 일부 목록입니다.

Tag	Description
<article>	Defines independent, self-contained content
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for a document or section
<header>	Specifies a header for a document or section
<main>	Specifies the main content of a document
<mark>	Defines marked/highlighted text
<nav>	Defines navigation links
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time

- 사용 가능한 모든 HTML 태그의 전체 목록을 보려면 HTML 태그를 참조하세요.

45. HTML Style Guide and Coding Conventions

- 일관되고 깨끗하며 깔끔한 HTML 코드는 다른 사람들이 코드를 더 쉽게 읽고 이해할 수 있도록 합니다.
- 다음은 좋은 HTML 코드를 만들기 위한 몇 가지 지침과 팁입니다.

◆ 항상 문서 유형 선언

- 항상 문서 유형을 문서의 첫 번째 줄로 선언하십시오.
- HTML에 대한 올바른 문서 유형은 다음과 같습니다.

```
<!DOCTYPE html>
```

◆ 소문자 요소 이름 사용

- HTML은 요소 이름에 대문자와 소문자를 혼합할 수 있습니다.
- 그러나 다음과 같은 이유로 소문자 요소 이름을 사용하는 것이 좋습니다.

  ✓ 대문자와 소문자 이름을 혼합하여 보기에 좋지 않습니다.
  ✓ 개발자는 일반적으로 소문자 이름을 사용합니다.
  ✓ 소문자가 더 깔끔해 보입니다.
  ✓ 소문자가 쓰기 쉽습니다.

◆ 모든 HTML 요소 닫기

- HTML에서는 모든 요소(예: <p>요소)를 닫을 필요가 없습니다 .
- 그러나 대부분의 HTML 요소는 반드시 닫는 것이 좋습니다.

◆ 소문자 속성 이름 사용

- HTML은 속성 이름에 대문자와 소문자를 혼합할 수 있습니다.
- 그러나 다음과 같은 이유로 소문자 속성 이름을 사용하는 것이 좋습니다.

  ✓ 대문자와 소문자 이름을 혼합하여 보기에 좋지 않습니다.
  ✓ 개발자는 일반적으로 소문자 이름을 사용합니다.
  ✓ 소문자 모양이 더 깔끔해집니다.
  ✓ 소문자가 쓰기 쉽습니다.

◆ 항상 속성 값에는 따옴표를 사용

- HTML은 따옴표 없이 속성 값을 허용합니다.
- 그러나 다음과 같은 이유로 속성 값을 인용하는 것이 좋습니다.

  ✓ 개발자는 일반적으로 속성 값을 인용합니다.
  ✓ 인용된 값은 더 읽기 쉽습니다.
  ✓ 값에 공백이 포함된 경우 따옴표를 사용해야 합니다.

◆ 항상 이미지의 Alt, 너비 및 높이 지정

- 항상 이미지의 alt 속성을 지정하십시오 . 이 속성은 어떤 이유로 이미지를 표시할 수 없는 경우에 중요합니다.
- 또한 항상 width및 height를 정의하십시오 . 이렇게 하면 브라우저가 로드하기 전에 이미지를 위한 공간을 예약할 수 있으므로 깜박임이 줄어듭니다.

◆ 공백 및 등호

- HTML은 등호 주위에 공백을 허용합니다. 그러나 space-less는 읽기 쉽고 엔터티를 더 잘 그룹화합니다.

◆ 긴 코드 줄 피하기

- HTML 편집기를 사용할 때 HTML 코드를 읽기 위해 좌우로 스크롤하는 것은 편리하지 않습니다.
- 너무 긴 코드 라인을 피하십시오.

◆ 빈 줄 및 들여쓰기

- 이유 없이 빈 줄, 공백 또는 들여쓰기를 추가하지 마십시오.
- 가독성을 위해 빈 줄을 추가하여 크거나 논리적인 코드 블록을 구분합니다.
- 가독성을 위해 들여쓰기를 두 칸 추가합니다. 불필요한 탭 키를 사용하지 마십시오.

◆ <title> 요소를 건너뛰지 마십시오.

- <title>요소는 HTML에서 필요합니다.
- 페이지 제목의 내용은 검색엔진최적화(SEO)에 매우 중요합니다! 페이지 제목은 검색 엔진 알고리즘에서 검색 결과에 페이지를 나

열할 때 순서를 결정하는 데 사용됩니다.

- <title>요소는 :

  - ✓ 브라우저 도구 모음에서 제목을 정의합니다.
  - ✓ 즐겨찾기에 추가될 때 페이지의 제목을 제공합니다.
  - ✓ 검색 엔진 결과에 페이지 제목을 표시합니다.

- 따라서 제목을 가능한 한 정확하고 의미 있게 만드십시오.

```
<title>HTML Style Guide and Coding Conventions</title>
```

◆ <html> 및 <body> 생략?

- HTML 페이지는 <html>및 <body>태그 없이도 가능합니다.
- 그러나 반드시 <html>및 <body>태그를 추가하는 것이 좋습니다 !

◆ <head> 생략?

- HTML <head> 태그도 생략할 수 있습니다.
- 브라우저는 <body>, 이전의 모든 요소를 기본 <head> 요소에 추가합니다 .
- 그러나 반드시 <head>태그를 사용하는 것이 좋습니다 .

◆ 빈(empty) HTML 요소를 닫으시겠습니까?

- HTML에서 빈 요소를 닫는 것은 선택 사항입니다.

허용 :

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

또한 허용됨:

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

◆ 언어 속성 추가

- 웹 페이지의 언어를 선언하려면 항상 태그 <html> 내부 에 lang속성을 포함해야 합니다. 이것은 검색 엔진과 브라우저를 지원하기 위한 것입니다.

예시

```
<!DOCTYPE html>
<html lang="en-us">
<head>
  <title>Page Title</title>
</head>
<body>

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

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

◆ 메타 데이터

- 적절한 해석과 올바른 검색 엔진 인덱싱을 보장하려면 언어와 문자 인코딩을 HTML 문서에서 가능한 한 빨리 정의해야 합니다.

```
<!DOCTYPE html>
<html lang="en-us">
<head>
  <meta charset="UTF-8">
  <title>Page Title</title>
</head>
```

◆ 뷰포트 설정

- 뷰포트는 웹 페이지에서 사용자가 볼 수 있는 영역입니다. 기기에 따라 다릅니다. 컴퓨터 화면보다 휴대전화가 더 작습니다.

모든 웹 페이지에 다음 <meta> 요소를 포함해야 합니다.

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

- 이것은 페이지의 크기와 배율을 제어하는 방법에 대한 브라우저 지침을 제공합니다.
- width=device-width부분 (기기에 따라 다름) 장치의 화면 폭을 따라 페이지의 폭을 설정한다.

- 이 initial-scale=1.0부분은 브라우저에서 페이지를 처음 로드할 때 초기 확대/축소 수준을 설정합니다.

◆ HTML 주석

- 짧은 주석은 다음과 같이 한 줄로 작성해야 합니다.

  <!-- This is a comment -->
  한 줄 이상의 주석은 다음과 같이 작성해야 합니다.

  <!--
    This is a long comment example. This is a long comment example.
    This is a long comment example. This is a long comment example.
  -->

- 긴 주석은 두 개의 공백으로 들여쓰기되어 있으면 더 쉽게 관찰할 수 있습니다.

◆ 스타일 시트 사용

- 스타일 시트에 연결하기 위해 간단한 구문을 사용하십시오(type속성은 필요하지 않음):

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

- 짧은 CSS 규칙은 다음과 같이 압축하여 작성할 수 있습니다.

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

- 긴 CSS 규칙은 여러 줄에 걸쳐 작성해야 합니다.

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

- Selector와 같은 줄에 여는 대괄호를 놓습니다.
- 여는 대괄호 앞에 한 칸 띄우기
- 들여쓰기 두 칸 사용
- 마지막을 포함하여 각 속성-값 쌍 뒤에 세미콜론을 사용하십시오.
- 값에 공백이 포함된 경우에만 값 주위에 따옴표를 사용하십시오.
- 선행 공백 없이 새 줄에 닫는 대괄호 배치

◆ HTML에서 JavaScript 로드

- 외부 스크립트를 로드하는 데 간단한 구문을 사용합니다(type속성은 필요하지 않음).

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

◆ JavaScript로 HTML 요소에 접근하기

- "정확하지 않은" HTML 코드를 사용하면 JavaScript 오류가 발생할 수 있습니다.

- 아래 두 JavaScript 문은 다른 결과를 생성합니다.

예시

```
getElementById("Demo").innerHTML = "Hello";

getElementById("demo").innerHTML = "Hello";
```

- JavaScript 과정에서 더 많은 것을 학습할 것입니다.

◆ 소문자 파일 이름 사용

- 일부 웹 서버(Apache, Unix)는 파일 이름에 대해 대소문자를 구분합니다. "london.jpg"는 "London.jpg"로 액세스할 수 없습니다.
- 다른 웹 서버(Microsoft, IIS)는 대소문자를 구분하지 않습니다. "london.jpg"는 "London.jpg"로 액세스할 수 있습니다.
- 대문자와 소문자를 혼용하여 사용하는 경우에는 이 점에 유의해야 합니다.
- 대소문자를 구분하지 않는 서버에서 대소문자를 구분하는 서버로 이동하면 작은 오류로도 웹이 손상됩니다!
- 이러한 문제를 방지하려면 항상 소문자 파일 이름을 사용하십시오!

◆ 파일 확장자

- HTML 파일에는 .html 확장자 가 있어야 합니다 (.htm 허용).
- CSS 파일의 확장자 는 .css 여야 합니다.
- JavaScript 파일에는 .js 확장자 가 있어야 합니다 .

◆ .htm과 .html의 차이점은 무엇입니까?

- .htm과 .html 파일 확장자 사이에는 차이가 없습니다!
- 둘 다 모든 웹 브라우저와 웹 서버에서 HTML로 처리됩니다.

◆ 기본 파일 이름

- URL이 끝에 파일 이름을 지정하지 않으면(예: "https://www.academy.com/"), 서버는 "index.html", "index.htm", "와 같은 기본 파일 이름을 추가합니다 (혹은 default.html" 또는 "default.htm").

- 서버가 기본 파일 이름으로 "index.html"만 사용하여 구성된 경우 파일 이름은 "default.html"이 아니라 "index.html"이어야 합니다.
- 그러나 서버는 둘 이상의 기본 파일 이름으로 구성할 수 있습니다. 일반적으로 원하는 만큼 기본 파일 이름을 설정할 수 있습니다.

45. HTML Entities

- HTML의 예약 문자는 문자 엔터티로 대체되어야 합니다.

◆ HTML 엔티티

- 일부 문자는 HTML에 예약되어 있습니다.
- 텍스트에 보다 작음(<) 또는 보다 큼(>) 기호를 사용하는 경우 브라우저에서 태그와 혼합할 수 있습니다.
- 문자 엔터티는 HTML에서 예약된 문자를 표시하는 데 사용됩니다.
- 캐릭터 엔티티는 다음과 같습니다.

```
&entity_name;
OR
&#entity_number;
```

- 보다 작음 기호(<)를 표시하려면 다음 과 같이 작성해야 합니다. **<** or **<**

- 엔티티 이름 사용의 장점: 엔티티 이름은 기억하기 쉽습니다.
- 엔티티 이름 사용의 단점: 브라우저는 모든 엔티티 이름을 지원하지 않을 수 있지만 엔티티 번호에 대한 지원은 대부분 합니다.

◆ Non-breaking Space

- HTML에서 일반적으로 사용되는 엔터티는 줄 바꿈하지 않는 공백입니다.
- 줄 바꿈하지 않는 공백은 새 줄로 나누지 않는 공백입니다.
- 줄 바꿈하지 않는 공백으로 구분된 두 단어는 함께 붙습니다(새 줄로 나누지 않음). 이것은 단어를 깨는 것이 방해가 될 수 있을

때 편리합니다.

예:

§ 10
10 km/h
10 PM

- 줄 바꿈하지 않는 공백의 또 다른 일반적인 용도는 브라우저가 HTML 페이지에서 공백을 자르는 것을 방지하는 것입니다.
- 텍스트에 10개의 공백을 쓰면 브라우저는 그 중 9개를 제거합니다. 텍스트에 실제 공백을 추가하려면 ** ** 캐릭터 엔티티를 추가하세요.

- 팁: 줄 바꿈하지 않는 하이픈(‑)은 줄 바꿈하지 않는 하이픈 문자(‑)를 정의하는 데 사용됩니다.

◆ 몇 가지 유용한 HTML 문자 엔터티

Result	Description	Entity Name	Entity Number
	non-breaking space		
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
"	double quotation mark	"	"
'	single quotation mark (apostrophe)	'	'
¢	cent	¢	¢
£	pound £	£	#163;
¥	yen	¥	¥

€	euro	&euro;	&#8364;
©	copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;

- 참고: 엔티티 이름은 대소문자를 구분합니다.

◆ 분음 부호 결합(Combining Diacritical Marks)

- 발음 구별 기호는 문자에 추가된 "상형 문자"입니다.
- 억음( ) 및 예음( )과 같은 일부 발음 구별 기호를 액센트라고 합니다.
- 분음 부호는 문자 위와 아래, 문자 내부 및 두 문자 사이에 나타날 수 있습니다.
- 분음 부호는 영숫자 문자와 함께 사용하여 페이지에 사용된 문자 집합(인코딩)에 없는 문자를 생성할 수 있습니다.

- 여기 몇 가지 예가 있습니다.

Mark	Character	Construct	Result
	a	à	a
	a	á	a
	a	â	a
	a	ã	a
	O	Ò	ò
	O	Ó	ó
	O	Ô	ô
	O	Õ	õ

46. HTML Symbols

- 키보드에 없는 기호는 엔터티를 사용하여 추가할 수도 있습니다.

◆ HTML Symbol Entities

- HTML 엔터티는 이전 장에서 설명했습니다.
- 많은 수학, 기술 및 통화 기호는 일반 키보드에 없습니다.
- HTML 페이지에 이러한 기호를 추가하려면 기호에 대한 엔티티 이름 또는 엔티티 번호(10진수 또는 16진수 참조)를 사용할 수 있습니다.

예시

유로 기호(€)를 엔티티 이름, 10진수 및 16진수 값으로 표시합니다.

```
<p>I will display &euro;</p>
<p>I will display &#8364;</p>
<p>I will display &#x20AC;</p>
```

◆ HTML에서 지원하는 일부 수학 기호

Char	Number	Entity	Description
∀	∀	∀	FOR ALL
∂	∂	∂	PARTIAL DIFFERENTIAL

∃	∃	∃	THERE EXISTS
∅	∅	∅	EMPTY SETS
∇	∇	∇	NABLA
∈	∈	∈	ELEMENT OF
∉	∉	∉	NOT AN ELEMENT OF
∋	∋	∋	CONTAINS AS MEMBER
∏	∏	∏	N-ARY PRODUCT
∑	∑	∑	N-ARY SUMMATION

◆ HTML에서 지원하는 일부 그리스 문자

Char	Number	Entity	Description
Α	Α	Α	GREEK CAPITAL LETTER ALPHA
Β	Β	Β	GREEK CAPITAL LETTER BETA
Γ	Γ	Γ	GREEK CAPITAL LETTER GAMMA
Δ	Δ	Δ	GREEK CAPITAL LETTER DELTA
Ε	Ε	Ε	GREEK CAPITAL LETTER EPSILON
Ζ	Ζ	Ζ	GREEK CAPITAL LETTER ZETA

◆ HTML에서 지원하는 기타 엔터티

Char	Number	Entity	Description
©	&#169;	&copy;	COPYRIGHT SIGN
®	&#174;	&reg;	REGISTERED SIGN
€	&#8364;	&euro;	EURO SIGN
™	&#8482;	&trade;	TRADEMARK
←	&#8592;	&larr;	LEFTWARDS ARROW
↑	&#8593;	&uarr;	UPWARDS ARROW
→	&#8594;	&rarr;	RIGHTWARDS ARROW
↓	&#8595;	&darr;	DOWNWARDS ARROW
♠	&#9824;	&spades;	BLACK SPADE SUIT
♣	&#9827;	&clubs;	BLACK CLUB SUIT
♥	&#9829;	&hearts;	BLACK HEART SUIT
♦	&#9830;	&diams;	BLACK DIAMOND SUIT

47. Using Emojis in HTML

- 이모티콘은 UTF-8 문자 집합의 문자입니다. 😄 😍 💖

◆ 이모티콘이란 무엇입니까?

- 그림 이모티콘은 이미지나 아이콘처럼 보이지만 실제로는 그렇지 않습니다.
- UTF-8(유니코드) 문자 집합의 문자(문자)입니다.
- UTF-8은 세계의 거의 모든 문자와 기호를 포함합니다.

◆ HTML 문자 집합 속성

- HTML 페이지를 올바르게 표시하려면 웹 브라우저가 페이지에 사용된 문자 집합을 알아야 합니다.
- 이것은 <meta>태그에 지정되어 있습니다 .

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

- 지정하지 않으면 UTF-8이 HTML의 기본 문자 집합입니다.

◆ UTF-8 문자

- 많은 UTF-8 문자는 키보드로 입력할 수 없지만 항상 숫자(엔티티 번호라고 함)를 사용하여 표시할 수 있습니다.

  ✓ A는 65

- ✓ B는 66
- ✓ C는 67

예시

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

<p>I will display A B C</p>
<p>I will display &#65; &#66; &#67;</p>

</body>
</html>
```

예시 설명

- ✓ <meta charset="UTF-8">요소는 문자 집합을 정의합니다.
- ✓ 문자 A, B, C는 숫자 65, 66, 67로 표시됩니다.

- 브라우저가 문자를 표시하고 있음을 이해하도록 하려면 엔티티 번호를 &#으로 시작하고 ; (세미콜론)으로 끝마치세요.

◆ 이모티콘 문자

이모티콘은 UTF-8 알파벳의 문자이기도 합니다.

- 😄은 128516
- 😍는 128525입니다
- 💘은 128151

예시

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

<h1>My First Emoji</h1>

<p>&#128512;</p>

</body>
</html>
```

- Emojis는 문자이기 때문에 HTML의 다른 문자와 마찬가지로 복사, 표시 및 크기 조정이 가능합니다.

예시

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

<h1>Sized Emojis</h1>

<p style="font-size:48px">
&#128512; &#128516; &#128525; &#128151;
</p>

</body>
</html>
```

- 전체 이모티콘 목록을 보려면 HTML Emoji Reference를 참조하세요 .

48. HTML Uniform Resource Locators

- URL은 웹 주소의 다른 단어입니다.
- URL은 단어(예: naver.com) 또는 인터넷 프로토콜(IP) 주소(예: 192.68.20.50)로 구성될 수 있습니다.
- 이름은 숫자보다 기억하기 쉽기 때문에 대부분의 사람들은 서핑할 때 이름을 입력합니다.

◆ URL - Uniform Resource Locator

- 웹 브라우저는 URL을 사용하여 웹 서버에서 페이지를 요청합니다.
- URL(Uniform Resource Locator)은 웹에서 문서(또는 기타 데이터)의 주소를 지정하는 데 사용됩니다.
- https://www.academy.com/html/default.asp 와 같은 웹 주소는 다음 구문 규칙을 따릅니다.

```
scheme://prefix.domain:port/path/filename
```

설명:

scheme	- 인터넷 서비스 유형을 정의합니다 (가장 일반적으로 http 또는 https 임).
prefix	- 도메인 접두사를 정의 합니다 (http의 기본값은 www 임).
domain	- 인터넷 도메인 이름을 정의합니다 (예: academy.com).
port	- 호스트의 포트 번호 를 정의합니다 (http의 기본값은 80).
path	- 서버의 경로 를 정의 합니다(생략된 경우: 사이트의 루트 디렉터리).
filename	- 문서 또는 리소스의 이름을 정의합니다.

◆ 일반적인 URL 체계

- 아래 표에는 몇 가지 일반적인 구성표가 나와 있습니다.

Scheme	Short for	Used for
http	HyperText Transfer Protocol	Common web pages. Not encrypted
https	Secure HyperText Transfer Protocol	Secure web pages. Encrypted
ftp	File Transfer Protocol	Downloading or uploading files
file		A file on your computer

◆ URL 인코딩

- URL은 ASCII 문자 집합을 사용하여 인터넷을 통해서만 보낼 수 있습니다 . URL에 ASCII 집합 이외의 문자가 포함되어 있으면 URL을 변환해야 합니다.
- URL 인코딩은 ASCII가 아닌 문자를 인터넷을 통해 전송할 수 있는 형식으로 변환합니다.
- URL 인코딩은 ASCII가 아닌 문자를 "%" 다음에 16진수로 대체합니다.
- URL은 공백을 포함할 수 없습니다. URL 인코딩은 일반적으로 공백을 더하기(+) 기호 또는 %20으로 바꿉니다.