



# Google HTML/CSS Style Guide (한글 번역)



이 문서는 2021 42 Seoul Hackathon 프로젝트 HTML & CSS Piscine 에서 42 Seoul 교육생들에게 HTML & CSS 코딩 style 지침으로 제공하고자 번역되었습니다. 불분명한 부분이 있다면 원문: [Google HTML/CSS Style Guide](#) 을 참고하세요.

## 1 Background

## 2 General

### 2.1 General Style Rules

#### 2.1.1 Protocol

### 2.2 General Formatting Rules

#### 2.2.1 Indentation

#### 2.2.2 Capitalization

#### 2.2.3 Trailing Whitespace

### 2.3 General Meta Rules

#### 2.3.1 Encoding

#### 2.3.2 Comments

#### 2.3.3 Action Items

## 3 HTML

### 3.1 HTML Style Rules

#### 3.1.1 Document Type

#### 3.1.2 HTML Validity

#### 3.1.3 Semantics

#### 3.1.4 Multimedia Fallback

#### 3.1.5 Separation of Concerns

#### 3.1.6 Entity References

#### 3.1.7 Optional Tags

#### 3.1.8 `type` Attributes

### 3.2 HTML Formatting Rules

#### 3.2.1 General Formatting

#### 3.2.2 HTML Line-Wrapping

#### 3.2.3 HTML Quotation Marks

## 4 CSS

### 4.1 CSS Style Rules

#### 4.1.1 CSS Validity

#### 4.1.2 ID and Class Naming

#### 4.1.3 ID and Class Name Style

#### 4.1.4 Type Selectors

#### 4.1.5 Shorthand Properties

#### 4.1.6 0 and Units

#### 4.1.7 Leading 0s

#### 4.1.8 Hexadecimal Notation

#### 4.1.9 Prefixes

#### 4.1.10 ID and Class Name Delimiters

#### 4.1.11 Hacks

### 4.2 CSS Formatting Rules

#### 4.2.1 Declaration Order

#### 4.2.2 Block Content Indentation

#### 4.2.3 Declaration Stops

#### 4.2.4 Property Name stops

#### 4.2.5 Declaration Block Separation

#### 4.2.6 Selector and Declaration Separation

#### 4.2.7 Rule Separation

#### 4.2.8 CSS Quotation Marks

### 4.3 CSS Meta Rules

#### 4.3.1 Section Comments

## Parting Words

# 1 Background

이 문서는 협업 효율, code quality 를 향상하기 위해, 인프라 지원을 가능하게 하기 위해 HTML 과 CSS 서식 및 스타일을 정의합니다. HTML & CSS raw, working 파일들 (GSS 파일들을 포함한) 에 적용됩니다. 코드 quality가 지켜지는 선에서 자유롭게 변경 및 간소화해서 사용하세요.

## 2 General

### 2.1 General Style Rules

#### 2.1.1 Protocol

HTTPS를 지원하지 않는게 아닌 이상, embedded resources( images, media files, style sheets, scripts, etc.) 에는 HTTPS([https:](https://))를 사용하세요.

```
<!-- Not recommended: omits the protocol -->
<script src="//ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>

<!-- Not recommended: uses HTTP -->
<script src="http://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
```

```
<!-- Recommended -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
```

```
/* Not recommended: omits the protocol */
@import '//fonts.googleapis.com/css?family=Open+Sans';

/* Not recommended: uses HTTP */
@import 'http://fonts.googleapis.com/css?family=Open+Sans';
```

```
/* Recommended */
@import 'https://fonts.googleapis.com/css?family=Open+Sans';
```

### 2.2 General Formatting Rules

#### 2.2.1 Indentation

들여쓰기는 `2 spaces` 를 사용하세요.

`tab` 들여쓰기 혹은 `tab` & `space` 혼용은 금지합니다.

```
<ul>
  <li>Fantastic
  <li>Great
</ul>
```

```
.example {
  color: blue;
}
```

## 2.2.2 Capitalization

소문자만 사용하세요.

모든 HTML element 명, attribute, attribute 값 (`text/CDATA` 가 아닌 이상), CSS selector, property, 그리고 property 값은 (`string` 이 아닌 이상) 소문자로 작성하세요.

```
<!-- Not recommended -->
<A HREF="/">Home</A>
```

```
<!-- Recommended -->

```

```
/* Not recommended */
color: #E5E5E5;
```

```
/* Recommended */
color: #e5e5e5;
```

## 2.2.3 Trailing Whitespace

diff를 복잡하게 만드는 불필요한 공백은 제거하세요.

```
<!-- Not recommended -->
<p>What?_
```

```
<!-- Recommended -->
<p>Yes please.
```

## 2.3 General Meta Rules

### 2.3.1 Encoding

UTF-8 encoding을 사용하세요.

BOM 없는 UTF-8 encoding이 적용된 editor 를 사용하세요.

`<meta charset="utf-8">` 로 HTML에 encoding을 명시하세요. style sheet에는 기본적으로 UTF-8이 적용 되었으므로 명시하지 않아도 됩니다.

encoding이 뭔지, 어떤 경우에 명시해야 하는지 궁금하다면 이 [링크](#)를 확인해보세요.

### 2.3.2 Comments

주석으로 코드가 어떤 목적을 위해 쓰였는지, 왜 특정 해법을 사용했는지 설명하세요. (프로젝트의 복잡도에 따라 상황이 다르기 때문에 필수 사항은 아닙니다).

### 2.3.3 Action Items

`todos` 와 `action items` 는 `TODO` 로 표시하세요.

`todo` 는 `TODO` 키워드로만 강조하세요. `@@` 와 같은 다른 서식은 금지합니다.

`contact`(user 명 혹은 mailing list)는 소괄호에 넣어 `TODO(contact)` 서식으로 추가하세요.

```
{# TODO(john.doe): revisit centering #}
<center>Test</center>
```

```
<!-- TODO: remove optional tags -->
<ul>
  <li>Apples</li>
  <li>Oranges</li>
</ul>
```

# 3 HTML

## 3.1 HTML Style Rules

### 3.1.1 Document Type

HTML5를 사용하세요.

모든 HTML 문서에는 HTML5(HTML syntax)가 선호됩니다: `<DOCTYPE html>`

(HTML은 `text/html` 형식으로 사용하는 것을 권장합니다. XHTML을 `application/xhtml+xml` 형식으로 사용하는 것은 HTML 보다 browser와 인프라 지원이 부족하고 최적화가 어렵습니다.

HTML에서 허용하지만 void element를 닫지 마세요. (`<br>` ✓ `<br />` ✗)

### 3.1.2 HTML Validity

W3C HTML validator 를 사용해 HTML code를 테스트하고 '인증된' 방식으로 HTML code를 작성하세요.

이 인증은 최소한의 quality check로 산업과 학업 현장에서 올바른 HTML 사용법의 기준을 제시해줍니다.

```
<!-- Not recommended -->
<title>Test</title>
<article>This is only a test.
```

```
<!-- Recommended -->
<!DOCTYPE html>
<meta charset="utf-8">
<title>Test</title>
<article>This is only a test.</article>
```

### 3.1.3 Semantics

HTML element를 목적에 맞게 사용하세요. heading element는 heading로, 문단은 `p` 로, hyperlink는 `a` 로 표기하세요.

HTML을 목적에 맞게 사용하는 것은 코드의 접근성과 효율성에 있어 매우 중요합니다.

```
<!-- Not recommended -->
<div onClick="goToRecommendations();">All recommendations</div>
```

```
<!-- Recommended -->
<a href="recommendations/">All recommendations</a>
```

### 3.1.4 Multimedia Fallback

멀티미디어에 대체 설명을 작성하세요.

사진, 영상, `canvas` 로 구현한 애니메이션에는 대체 설명을 작성하세요. 사진의 경우 사진의 의미를 설명하는 텍스트를 제공하세요. 영상/음성 매체의 경우는 가능하다면 대본 혹은 자막을 제공하세요.

대체 설명은 접근성에 있어 매우 중요합니다. 예를 들어 `@alt` 에 텍스트로 사진에 대한 설명이 제공되어있지 않다면 시각장애인은 HTML 이 전달하고자 하는 바를 온전히 이해할 수 없을 것입니다. 영상/음성 매체의 경우에도 사용자가 어떤 이유로든지 영상/매체가 전달하고자 하는 바를 직관적으로 이해하지 못하는 상황이 있을 수 있습니다.

(불필요한 중복이나 오직 외관적인 목적으로 사용된 사진에는 `alt=` 로 대체 설명을 제공하지 않아도 됩니다.)

```
<!-- Not recommended -->

```

```
<!-- Recommended -->

```

### 3.1.5 Separation of Concerns

구조(markup), 디자인(sytling), 동작(scripting)을 엄격하게 분리하세요. 문서와 템플릿은 오직 HTML로만, 디자인적인 요소는 오직 style sheet으로만, 모든 동작은 scripts로만 작성하세요. 최소한의 linking으로 이 세 영역의 접촉을 최소화하세요.

이는 유지보수 측면에서 매우 중요합니다. 항상 HTML 문서 및 템플릿을 바꾸는게 style sheets 혹은 scripts를 바꾸는 것보다 많은 문제를 일으킵니다.

```

<!-- Not recommended -->
<!DOCTYPE html>
<title>HTML sucks</title>
<link rel="stylesheet" href="base.css" media="screen">
<link rel="stylesheet" href="grid.css" media="screen">
<link rel="stylesheet" href="print.css" media="print">
<h1 style="font-size: 1em;">HTML sucks</h1>
<p>I've read about this on a few sites but now I'm sure:
  <u>HTML is stupid!!1</u>
<center>I can't believe there's no way to control the styling of
  my website without doing everything all over again!</center>

```

```

<!-- Recommended -->
<!DOCTYPE html>
<title>My first CSS-only redesign</title>
<link rel="stylesheet" href="default.css">
<h1>My first CSS-only redesign</h1>
<p>I've read about this on a few sites but today I'm actually
  doing it: separating concerns and avoiding anything in the HTML of
  my website that is presentational.
<p>It's awesome!

```

### 3.1.6 Entity References

팀이 사용하는 파일들, editor들이 모두 같은 encoding을 사용한다고 간주하고 `&mdash;`, `&rdquo;`, `#x263a;` 와 같은 HTML 예약어를 사용하지 마세요.

`<` 와 `&` 같이 HTML에서 특별한 의미를 갖는 문자일 경우와 control 문자들만 예외적으로 허용합니다.

### 3.1.7 Optional Tags

(선택사항) 선택적 tag들을 생략할 수 있습니다.

파일 크기 최적화와 검색을 편하게 하기 위한 목적으로 원한다면 생략할 수 있는 tag들이 있습니다. [HTML5 specification](#)에 생략 가능한 tag들이 명시 되어있습니다.

(이는 대부분의 웹 개발자들이 배운 내용과 다르기 때문에 보편적인 지침으로 자리 잡기에는 유예기간이 필요할 수 있습니다. 일관성과 간결함을 위해 생략 가능한 tag들을 생략한다면 선택적으로 일부 tag만 생략하는게 아니라 모두 생략하기를 권고합니다.)

```

<!-- Not recommended -->
<!DOCTYPE html>

```

```
<html>
  <head>
    <title>Spending money, spending bytes</title>
  </head>
  <body>
    <p>Sic.</p>
  </body>
</html>
```

```
<!-- Recommended -->
<!DOCTYPE html>
<title>Saving money, saving bytes</title>
<p>Qed.
```

### 3.1.8 `type` Attributes

style sheets로 CSS를 사용하지 않거나 scripts로 JavaScript를 사용하지 않는게 아니라면 style sheets 혹은 scripts를 위한 `type` attribute는 생략하세요. HTML5는 기본적으로 `type` attribute이 `text/css` 와 `text/javascript` 로 설정 되어있습니다.

```
<!-- Not recommended -->
<link rel="stylesheet" href="https://www.google.com/css/maia.css"
      type="text/css">
```

```
<!-- Recommended -->
<link rel="stylesheet" href="https://www.google.com/css/maia.css">
```

```
<!-- Not recommended -->
<script src="https://www.google.com/js/gweb/analytics/autotrack.js"
        type="text/javascript"></script>
```

```
<!-- Recommended -->
<script src="https://www.google.com/js/gweb/analytics/autotrack.js"></script>
```

## 3.2 HTML Formatting Rules

### 3.2.1 General Formatting

모든 `block`, `list` 혹은 `table` element는 개행으로 띄어쓰고, 모든 자식 element는 들여쓰세요.

이는 element의 styling에 관계없이 적용됩니다.

```
<blockquote>
  <p><em>Space</em>, the final frontier.</p>
</blockquote>
```

```
<ul>
  <li>Moe
  <li>Larry
  <li>Curly
</ul>
```

```
<table>
  <thead>
    <tr>
      <th scope="col">Income
      <th scope="col">Taxes
    </tr>
  <tbody>
    <tr>
      <td>$ 5.00
      <td>$ 4.50
    </tr>
  </tbody>
</table>
```

### 3.2.2 HTML Line-Wrapping

(선택사항) 줄이 길어질 경우 개행으로 쪼개주세요. HTML에는 줄 길이 제한이 없지만, 가독성을 높이기 위해 줄이 너무 길어질 경우 개행으로 분할하는 것을 추천합니다.

줄을 분할할 때, 따라오는 줄은 이전 줄보다 최소 `4 spaces` 들여써야합니다.

```
<md-progress-circular md-mode="indeterminate" class="md-accent"
  ng-show="ctrl.loading" md-diameter="35">
</md-progress-circular>
```

```
<md-progress-circular md-mode="indeterminate" class="md-accent"
  ng-show="ctrl.loading" md-diameter="35">
```

```
</md-progress-circular>
<md-progress-circular
  md-mode="indeterminate"
  class="md-accent"
  ng-show="ctrl.loading"
  md-diameter="35">
</md-progress-circular>
```

```
<md-progress-circular md-mode="indeterminate"
  class="md-accent"
  ng-show="ctrl.loading"
  md-diameter="35">
</md-progress-circular>
```

### 3.2.3 HTML Quotation Marks

attribute 값에는 쌍따옴표("")를 사용하세요.

```
<!-- Not recommended -->
<a class='maia-button maia-button-secondary'>Sign in</a>
```

```
<!-- Recommended -->
<a class="maia-button maia-button-secondary">Sign in</a>
```

## 4 CSS

### 4.1 CSS Style Rules

#### 4.1.1 CSS Validity

CSS validator에 bug가 있거나 proprietary syntax(e.g. -webkit-, -moz-, etc. 접두사가 붙는 특정 browser에 관련된 syntax)가 요구되지 않는 이상 [W3C CSS validator](#) 와 같은 도구로 '인증된' CSS code 를 작성하세요.

이 인증은 최소한의 quality check로 산업과 학업 현장에서 올바른 CSS 사용법의 기준을 제시해줍니다.

## 4.1.2 ID and Class Naming

의미있는 혹은 일반적인 ID/class 명을 사용하세요.

외관적인 특징을 수수께끼 같은 이름이 아니라 항상 element의 목적을 반영하는 ID/class 명을 사용하세요.

의미있는 ID/class 명을 사용하는게 가장 이해하기 쉽고 좋습니다. 마땅한 이름이 떠오르지 않을 때에만 일반적인 이름을 사용하세요.

```
/* Not recommended: meaningless */
#yee-1901 {}

/* Not recommended: presentational */
.button-green {}
.clear {}
```

```
/* Recommended: specific */
#gallery {}
#login {}
.video {}

/* Recommended: generic */
.aux {}
.alt {}
```

## 4.1.3 ID and Class Name Style

ID/class 명은 필요한 만큼 길게, 하지만 가능한 짧게 쓰세요. 가능한 간략하게 무엇을 위한 ID/class 인지 전달하세요.

간략한 ID/class 명은 코드를 더 이해하기 쉽게 만들어 주고, 코드 효율성을 높여줍니다.

```
/* Not recommended */
#navigation {}
.atr {}
```

```
/* Recommended */
#nav {}
.author {}
```

## 4.1.4 Type Selectors

불가피한 상황이 아니라면, element 명을 ID/class 명과 붙여 쓰지 마세요.

성능 개선을 위해 불필요한 상위 selector 사용은 피해야합니다.

```
/* Not recommended */
ul#example {}
div.error {}
```

```
/* Recommended */
#example {}
.error {}
```

## 4.1.5 Shorthand Properties

가능한 property는 축약해서 사용하세요.

CSS에는 `font` 와 같은 축약 가능한 property들이 있습니다. 효율성을 높이고 코드를 더 이해하기 쉽게 하기 위해 한 property만 사용되었더라도 축약해서 사용하세요.

```
/* Not recommended */
border-top-style: none;
font-family: palatino, georgia, serif;
font-size: 100%;
line-height: 1.6;
padding-bottom: 2em;
padding-left: 1em;
padding-right: 1em;
padding-top: 0;
```

```
/* Recommended */
border-top: 0;
font: 100%/1.6 palatino, georgia, serif;
padding: 0 1em 2em;
```

## 4.1.6 0 and Units

반드시 필요한게 아니라면 "0"이후 단위는 생략하세요.

```
flex: 0px; /* This flex-basis component requires a unit. */
flex: 1 1 0px; /* Not ambiguous without the unit, but needed in IE11. */
margin: 0;
padding: 0;
```

### 4.1.7 Leading 0s

소수점 앞에 나오는 "0"은 생략하세요. -1과 1 사이의 값의 소수점 앞에 "0"은 생략하세요.

```
font-size: .8em;
```

### 4.1.8 Hexadecimal Notation

색상 표기시 3 문자 16진수 표기법이 가능하다면 사용하세요.

```
/* Not recommended */
color: #eebbcc;
```

```
/* Recommended */
color: #ebc;
```

### 4.1.9 Prefixes

(선택사항) 특정 application을 위한 selector에는 적절한 접두사를 붙여주세요.

큰 프로젝트에 사용되는 혹은 다른 프로젝트나 외부 사이트에 embed 되는 코드에는 ID/class 명에 접두사를 넣어주세요. 짧은 고유의 식별자에 dash를 붙여 사용하세요.

```
.adw-help {} /* AdWords */
#maia-note {} /* Maia */
```

### 4.1.10 ID and Class Name Delimiters

ID/class 명은 하이픈(-)으로 단어를 구분해서 쓰세요. 이는 코드를 이해하기 쉽게 만들어주고 검색을 쉽게 해줍니다.

```
/* Not recommended: does not separate the words "demo" and "image" */
.demoimage {}

/* Not recommended: uses underscore instead of hyphen */
.error_status {}
```

```
/* Recommended */
#video-id {}
.ads-sample {}
```

### 4.1.11 Hacks

사용자의 browser를 감지하는 CSS "hacks"에 의존하지 마세요. CSS filters, workarounds 혹은 hacks에 의존하는 것은 효율적이고 관리하기에 용이한 코드로 유지하는데 도움이 되지 않습니다.

## 4.2 CSS Formatting Rules

### 4.2.1 Declaration Order

알파벳 순으로 선언하세요.

통일성 있는 코드를 짤 수 있고 좀 더 쉽게 기억하거나 유지보수 할 수 있습니다.

특정 브라우저를 위한 접두사는 무시하고 정렬하세요. 하지만 다양한 브라우저를 고려한 접두사를 붙일때, 접두사의 순서는 알파벳 순을 유지해야 합니다. (e.g. -moz는 -webkit 앞에 옵니다).

```
background: fuchsia;
border: 1px solid;
-moz-border-radius: 4px;
-webkit-border-radius: 4px;
border-radius: 4px;
color: black;
text-align: center;
text-indent: 2em;
```

### 4.2.2 Block Content Indentation

모든 블록 내용을 들여쓰기 하세요.

블록 안 블록 내용 또한 들여쓰기를 하세요. 이는 블록의 의존성을 보여주고 이해하기 쉽게 만들어줍니다.

```
@media screen, projection {  
  
  html {  
    background: #fff;  
    color: #444;  
  }  
  
}
```

### 4.2.3 Declaration Stops

모든 선언 이후 세미콜론을 붙이는것은 코드를 일관성 있게 만들어주고, 확장성이 좋아집니다.

```
/* Not recommended */  
.test {  
  display: block;  
  height: 100px  
}
```

```
/* Recommended */  
.test {  
  display: block;  
  height: 100px;  
}
```

### 4.2.4 Propert Name stops

일관성을 위해 항상 property와 값 사이에 공백을 넣으세요. (property와 ':'은 붙여쓰세요.)

```
/* Not recommended */  
h3 {  
  font-weight:bold;  
}
```

```
/* Recommended */  
h3 {
```

```
font-weight: bold;
}
```

## 4.2.5 Declaration Block Separation

마지막 selector와 선언 block을 여는 중괄호는 **1 space** 띄어쓰세요.

```
/* Not recommended: missing space */
#video{
  margin-top: 1em;
}

/* Not recommended: unnecessary line break */
#video
{
  margin-top: 1em;
}
```

```
/* Recommended */
#video {
  margin-top: 1em;
}
```

## 4.2.6 Selector and Declaration Separation

항상 각 selector와 선언 사이에 개행을 넣으세요.

```
/* Not recommended */
a:focus, a:active {
  position: relative; top: 1px;
}
```

```
/* Recommended */
h1,
h2,
h3 {
  font-weight: normal;
  line-height: 1.2;
}
```

## 4.2.7 Rule Separation

각 rule 사이에 하나의 빈 줄을 넣으세요.

```
html {
  background: #fff;
}

body {
  margin: auto;
  width: 50%;
}
```

## 4.2.8 CSS Quotation Marks

property 값이나 attribute selector를 위해 큰 따옴표 보다는 작은 따옴표를 쓰세요.

URL(`url()`) 을 위해서 따옴표를 쓰지 마세요.

예외: 만약 `@charset` 을 쓰기 원한다면, 큰 따옴표를 쓰세요. (작은 따옴표는 불허합니다.)

```
/* Not recommended */
@import url("https://www.google.com/css/maia.css");

html {
  font-family: "open sans", arial, sans-serif;
}
```

```
/* Recommended */
@import url(https://www.google.com/css/maia.css);

html {
  font-family: 'open sans', arial, sans-serif;
}
```

## 4.3 CSS Meta Rules

### 4.3.1 Section Comments

(선택사항) section 주석을 사용하여 각 section을 분류하세요. 각 section은 빈 줄로 구분하세요.

```
/* Header */  
  
#adw-header {}  
  
/* Footer */  
  
#adw-footer {}  
  
/* Gallery */  
  
.adw-gallery {}
```

## Parting Words

일관성 있는 코드를 쓰세요.

코드를 작성하고 있다면, 잠깐 주위에 있는 동료들의 코드를 보고 style을 정하세요. 만약 그들이 모든 연산자 앞뒤에 `space` 를 입력한다면, 당신도 그렇게 하세요. 만약 그들의 주석이 hash marks로 둘러싸여 있다면, 당신의 주석도 동일하게 작성하세요.

style 지침이 존재하는 이유는 '공통된 코딩 어휘/용어'를 사용해서 사람들이 당신이 '코드를 어떻게 작성하고 있는지'가 아니라 '코드로 무엇을 말하고 있는지'에 집중하게 하기 위함입니다. 우리는 사람들이 '공통된 어휘/용어'를 알 수 있게 하기 위해 global style 지침을 제공합니다. 하지만 자체적인 style 지침도 중요합니다. 만약 당신이 작성한 코드가 동료들의 코드와 상당히 다르다면, 코드를 읽는 사람들의 흐름이 끊기겠죠? 이를 지양하세요.