

# 네이버 코딩 컨벤션 (HTML/CSS)

Ver 0.1.0

---

# 저작권

Copyright © 2015 Naver Corp. All Rights Reserved.

이 문서는 Naver(주)의 지적 자산이므로 Naver(주)의 승인 없이 이 문서를 다른 용도로 임의 변경하여 사용할 수 없습니다. 이 문서는 정보 제공의 목적으로만 제공됩니다. Naver(주)는 이 문서에 수록된 정보의 완전성과 정확성을 검증하기 위해 노력하였으나, 발생할 수 있는 내용상의 오류나 누락에 대해서는 책임지지 않습니다. 따라서 이 문서의 사용이나 사용 결과에 따른 책임은 전적으로 사용자에게 있으며, NHN(주)는 이에 대해 명시적 혹은 묵시적으로 어떠한 보증도 하지 않습니다. 관련 URL 정보를 포함하여 이 문서에서 언급한 특정 소프트웨어 상품이나 제품은 해당 소유자의 저작권법을 따르며, 해당 저작권법을 준수하는 것은 사용자의 책임입니다. Naver(주)는 이 문서의 내용을 예고 없이 변경할 수 있습니다.

---

# 목차

1. 가이드의 취지 .....	1
2. 운영 규칙 .....	2
2.1. 문의와 요청 경로 .....	2
2.2. 프로젝트별로 정의할 파일 .....	2
라이선스 파일 포함 .....	2
프로젝트별 정책 파일 포함 .....	2
3. 파일 공통 요건 .....	3
3.1. 파일 인코딩은 UTF-8 .....	3
3.2. 새줄 문자는 LF .....	3
4. 기본 작성 규칙 [General Rules] .....	4
4.1. 인코딩 [Encoding] .....	4
4.2. 들여쓰기 [Indentation] .....	4
4.3. 대소문자 [Capitalization] .....	5
4.4. 행의 끝 [End of Lines] .....	5
4.5. 폴더구조 [Folder structures] .....	5
루트 폴더의 네이밍 .....	5
기본 폴더 구조 .....	5
확장형 폴더 구조(sass 및 grunt 등의 사용) .....	6
5. HTML 작성 규칙 [HTML Style Rules] .....	7
5.1. Doctype [Doctype] .....	7
5.2. 유효성 검사 [HTML Validity] .....	7
5.3. 시멘틱 마크업 [Semantics] .....	7
5.4. 구조와 표현의 분리 [Separation of Concerns] .....	7
5.5. 대체텍스트 제공 [Alternative text] .....	8
5.6. 엔티티 캐릭터터 [Entity Characters] .....	8
5.7. 속성의 생략 [Omit attributes] .....	8
boolean attributes의 속성값 생략 .....	8
type의 생략 .....	9
6. HTML 문법 규칙 [HTML Formatting Rules] .....	10
6.1. 들여쓰기와 개행[HTML Indentation and new line] .....	10
6.2. 공백 [Whitespaces] .....	10
6.3. 따옴표 [HTML Quotation marks] .....	10
6.4. 빈줄(Empty Lines) .....	11
6.5. 주석 [Comments] .....	11
7. CSS 작성 규칙 [CSS Style Rules] .....	12
7.1. 유효성 검사 [CSS Validity] .....	12
7.2. 속성의 축약 [Shorthand Properties] .....	12
7.3. 단위의 생략 [Omit Units] .....	12
7.4. 색상값의 표기법 [Hexadecimal Notation] .....	13
7.5. id/class 네이밍 규칙 [Id and class naming] .....	13
7.6. 이미지 대체 기법 [Image Replacement] .....	14
8. CSS 문법 규칙 [CSS Formatting Rules] .....	16

8.1. 들여쓰기 [CSS Indentation] .....	16
8.2. 빈칸과 세미콜론 [Space and semicolon] .....	16
빈칸 .....	16
세미콜론(;) .....	16
8.3. 따옴표 [CSS Quotation Marks] .....	17

---

## 1. 가이드의 취지

이 가이드는 프로젝트 구성원 간의 코드 형식에 대한 합의 수준을 높이기 위한 목적으로 작성되었다. 가이드를 적용하는 프로젝트에서 다음과 같은 이득이 있기를 기대한다.

첫째, 기본적인 코드 형식에 대해서 프로젝트마다 초기에 반복 논의하는 시간을 줄인다. 가이드에서 설명한 보편적인 요소를 바탕으로 프로젝트별로 예외나 정교한 규칙에 대한 논의를 짧은 시간 안에 할 수 있도록 한다. 단순한 코드 형식보다 깊이 있는 규칙을 정하는 단계로 빨리 진입하도록 돕는다.

둘째, 코드 리뷰 과정에서 핵심로직에 집중할 수 있게 한다. 정교하지 못한 규칙으로 프로젝트를 진행하면 코드 리뷰 때 형식을 맞추기 위한 지적과 반영에 시간을 소모한다. 버그 수정, 기능 추가를 할 때도 줄, 공백을 바꾼 부분으로 인한 DIFF가 많이 보인다면 변경의 핵심을 파악하는데 방해가 된다. 컨벤션 가이드는 그런 요소를 최소화하여 핵심적인 변경을 파악하고 리뷰하는 시간을 줄여준다.

네이버의 공통 컨벤션 가이드는 외부의 가이드와 대비해서 아래와 같은 장점을 추구한다.

첫째, 외부의 가이드보다 적극적으로 개발자의 의견이 반영된다. 네이버의 구성원이 가이드 내용, 룰체크 설정을 유지 보수하기 때문이다.

둘째, 오픈소스 컨트리뷰션과 같은 방식으로 가이드의 내용을 수정 제안할 수 있다. 가이드는 프로그램 소스와 유사한 방식으로 가이드를 관리한다. ASCIIDOC 형식인 텍스트 파일로 작성하고 GIT으로 버전을 관리한다.

이 가이드에서 네이버의 개발자 공동체에 의해 유지되고 발전되는 기준을 담아가고자 한다.

## 2. 운영 규칙

### 2.1 문의와 요청 경로

가이드 내용에 대한 문의와 의견은 <https://oss.navercorp.com/engineering/coding-convention/issues> 에 새로운 이슈로 등록한다. 이슈를 등록할 때 제목은 `[언어][규칙 식별자]`로 시작한다. 예시는 다음과 같다.

```
[Java][avoid-1-char-var] 바람직한 사례를 추가
```

규칙 식별자는 각 규칙의 제목 다음에 표기되어 있다.

### 2.2 프로젝트별로 정의할 파일

#### 라이선스 파일 포함

[file-licence]

프로젝트의 루트 디렉토리에 `LICENSE`라는 파일명으로 라이선스를 명시한다.

#### 프로젝트별 정책 파일 포함

[file-convention]

프로젝트의 루트 디렉토리에 CONVENTION.\* 파일명으로 프로젝트별 규칙을 명시한다. 파일의 확장자는 제한을 두지 않으나 LICENSE.md, LICENSE.adoc 등 해당 파일의 포맷을 나타내는 관례를 따른다. 텍스트 에디터로 열어볼 수 있는 포맷을 권장한다.

공통 가이드의 내용 외에 프로젝트에 특화된 컨벤션을 정의하지 않는다면 생략가능하다.

해당 파일에는 아래와 같은 내용이 포함될 수 있다.

- 공통 가이드에 내용 중 해당 프로젝트에서는 예외로 할 내용
  - 예) 테스트 코드의 함수 이름에서는 "\_"를 허용한다.
- 디렉토리/패키지 구성 관례
  - 예) 통합테스트는 /src/main/integration-test에 포함시킨다.
- 해당 프로젝트에 쓰는 라이브러리/프레임워크에 특화된 내용
  - 예) jquery의 bind() 함수대신 on() 함수를 사용한다.

---

## 3. 파일 공통 요건

### 3.1 파일 인코딩은 UTF-8

[encoding-utf8]

모든 소스, 텍스트 문서 파일의 인코딩은 UTF-8로 통일한다.

### 3.2 새줄 문자는 LF

[newline-lf]

Unix 형식의 새줄 문자(newline)인 LF(Line Feed, 0x0A)을 사용한다. Windows 형식인 CRLF가 섞이지 않도록 편집기와 GIT 설정 등을 확인한다.

## 4. 기본 작성 규칙 [General Rules]

### 4.1 인코딩 [Encoding]

UTF-8을 사용하고, 코드 작성시 도입부에 인코딩 방식을 선언한다.

HTML <sup>1</sup>.

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

or

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

CSS <sup>2</sup>.

```
@charset "UTF-8";
```



#### Note

에디터에서 소스코드 저장시 선언된 인코딩과 동일한지 확인한다.

### 4.2 들여쓰기 [Indentation]

들여쓰기는 탭을 이용하고, 탭의 크기는 공백 4칸으로 한다.

공백으로 들여쓰기 하지 않도록 주의한다.

나쁜 예.

```
<div>
# <h1>마크업 코딩 컨벤션</h1>
# <p>에디터 설정에서 탭의 간격을 2칸으로 설정.</p>
</div>

<div>
####<h1>마크업 코딩 컨벤션</h1>
####<p>공백으로 들여쓰기 하지 않는다.</p>
</div>
```

좋은 예.

```
<div>
# <h1>마크업 코딩 컨벤션</h1>
# <p>들여쓰기는 탭을 이용하고, 1탭의 크기는 공백 4칸.</p>
</div>
```



#### Note

탭으로 들여쓰기는 중첩이 깊어질 경우 소스코드의 가독성에 문제가 발생할 수 있으므로 엘리먼트의 중첩에 유의한다.

<sup>1</sup>html의 인코딩 선언 방법 <http://www.w3.org/International/questions/qa-html-encoding-declarations>

<sup>2</sup>CSS에서의 인코딩 선언 방법 <http://www.w3.org/International/questions/qa-css-charset>



## 4.3 대소문자 [Capitalization]

모든 코드는 소문자로만 사용한다.

문자값으로 사용되는 경우를 제외하고 HTML의 엘리먼트, 속성, 속성값 그리고 CSS의 셀렉터, 속성, 속성값은 반드시 소문자로 작성한다.

나쁜 예.

```
<IMG SRC="img/logo_naver.png" ALT="Naver" WIDTH="210" HEIGHT="78">

div{color: #FADADA;}
.nanum {
  Font-Family: NanumGothic,Sans-serif;
}
```

좋은 예.

```


div {
  color: #fadada;
}
.nanum {
  font-family: NanumGothic,Sans-serif;
}
```



### Note

CSS에서는 대소문자 구분을 하지만, HTML에서는 대소문자 구분을 하지 않는다.  
id/class 속성은 HTML문서에 포함되어 대소문자 구분이 되지 않으므로 소문자로만 작성한다.

## 4.4 행의 끝 [End of Lines]

모든 행의 끝은 반드시 개행되어야 한다.<sup>3</sup>



### Note

행의 정의에 따라 파일의 맨 마지막 행도 개행문자로 끝나야 하기 때문에 빈줄을 하나 추가한다.

## 4.5 폴더구조 [Folder structures]

### 루트 폴더의 네이밍

루트 폴더명은 반드시 프로젝트명 또는 서비스명으로 하고, 축약하지 않는다.

모바일 등의 별도 환경에 따라 저장소가 구분되는 경우 기기에 따라 prefix를 추가할 수 있으나,  
데스크탑 환경은 별도 prefix 사용 없이 프로젝트명 또는 서비스명으로 사용한다.

### 기본 폴더 구조

```
mail
└─ src
```

<sup>3</sup>POSIX 명세에 따른 "행"의 정의 [http://pubs.opengroup.org/onlinepubs/000095399/basedefs/xbd\\_chap03.html#tag\\_03\\_205](http://pubs.opengroup.org/onlinepubs/000095399/basedefs/xbd_chap03.html#tag_03_205)

```
└ img
└ css
└ js
```

### 확장형 폴더 구조(sass 및 grunt 등의 사용)

SASS 및 grunt 등의 preprocessor 사용시 추가적인 폴더구조가 필요할수 있다.  
이 경우에도 최종 산출 소스의 폴더 구조는 기본 폴더 구조와 최대한 동일하도록 한다.

```
mail
└ src
  └ img
  └ css
  └ [scss]
└ [dist]
  └ ...
  └ ...
└ [doc]
└ [guide]
  ...
  ...
```



#### Note

SASS 폴더는 예외적으로 CSS 폴더와 동일한 레벨에 위치하도록 한다.

## 5. HTML 작성 규칙 [HTML Style Rules]

### 5.1 Doctype [Doctype]

HTML5 doctype <sup>1</sup>을 사용하고, XHTML은 사용하지 않는다.

```
<!doctype html>
```



Note

XHTML보다 HTML이 브라우저 지원과 성능 최적화에 좀 더 유리하다.

### 5.2 유효성 검사 [HTML Validity]

HTML은 W3C Validation <sup>2</sup>을 통과해야 한다.



Note

유효성 검사를 통해 품질 측정이 가능하고, 표준명세에 따른 요구 사항과 제약 사항에 대한 학습이 가능하다.

### 5.3 시멘틱 마크업 [Semantics]

의미에 맞는 엘리먼트를 사용한다.

나쁜 예.

```
<div onclick="goHome();">Home</div>
```

좋은 예.

```
<a href="home/">Home</a>
```



Note

의미에 맞는 엘리먼트의 사용은 접근성 보장과 검색 엔진 최적화를 위해 필수적이고, 문서의 구조 파악을 용이하게 한다.

### 5.4 구조와 표현의 분리 [Separation of Concerns]

구조(HTML)와 표현(CSS)을 분리하여 작성한다.

나쁜 예.

```
<h1 style="font-size:1.2em;color:#333">구조와 표현의 분리</h1>
<p>html과 css를 <b>분리</b>하여 작성해주세요.</p>
<center><u>유지보수</u>가 어렵습니다.</center>
```

좋은 예.

<sup>1</sup>doctype 표준 규칙 <http://www.w3.org/TR/html-markup/syntax.html#doctype-syntax>

<sup>2</sup>W3C Validator <http://validator.w3.org/>

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

<h1 class="title">구조와 표현의 분리</h1>
<p class="description">html과 css를 <strong>분리</strong>하여 작성해주세요.</p>
<p><em>유지보수</em>가 쉬워집니다.</p>
```



## Note

HTML과 CSS/JS의 접점이 적을수록 유지보수의 이점이 있다.

## 5.5 대체텍스트 제공 [Alternative text]

이미지나 동영상, 캔버스를 이용한 애니메이션 등의 멀티미디어 콘텐츠에는 그를 설명해 줄 수 있는 대체텍스트를 반드시 포함해야 한다.

나쁜 예.

```

```

좋은 예.

```

```



## Note

순전히 디자인을 위한 요소인 경우 css를 이용하여 처리하고,  
대체텍스트가 중복으로 제공되는 경우 alt="" 와 같이 빈 속성을 추가한다.

## 5.6 엔티티 캐릭터 [Entity Characters]

HTML의 문법상 특별한 의미를 지닌 문자들(<, >, &, 공백문자[&nbsp;], 따옴표)을 제외한 엔티티 캐릭터의 사용은 지양한다.



## Note

처럼 문법상 영향을 주지 않는 특수문자를 엔티티캐릭터로 표현하는것은, 소스코드의 가독성을 해치기만 할 뿐 특별한 이점이 없다.

## 5.7 속성의 생략 [Omit attributes]

### boolean attributes의 속성값 생략

특정 속성값을 가지지 않는 boolean attributes<sup>3</sup>의 속성값은 생략한다.

나쁜 예.

```
<input type="radio" checked="checked">
<input type="button" disabled="disabled">
<select>
  <option value="1" selected="selected">옵션1</option>
</select>
```

<sup>3</sup><http://www.w3.org/TR/html5/infrastructure.html#boolean-attribute>

좋은 예.

```
<input type="radio" checked>
<input type="button" disabled>
<select>
  <option value="1" selected>옵션1</option>
</select>
```



#### Note

boolean attributes는 속성을 추가하는 것으로 true값을 가지게 되고, false값 지정을 위해서는 속성을 완전히 제거해야만 한다.

속성값의 표시가 불필요한 속성으로 속성값을 생략한다.

### type의 생략

link, script 엘리먼트의 type 속성은 생략한다.

나쁜 예.

```
<link rel="stylesheet" href="http://www.naver.com/css/main.css" type="text/css">
<script src="http://www.naver.com/js/main.js" type="text/javascript">
```

좋은 예.

```
<link rel="stylesheet" href="http://www.naver.com/css/main.css">
<script src="http://www.naver.com/js/main.js">
```



#### Note

HTML5 독타입을 사용하는 경우 link의 mime type은 "text/css", script의 mime type은 "text/javascript"로 기본값이 설정되어 있다.

## 6. HTML 문법 규칙 [HTML Formatting Rules]

### 6.1 들여쓰기와 개행[HTML Indentation and new line]

모든 block 요소는 개행 후 들여쓰기 한다.

나쁜 예.

```
<dl><dt>Title :</dt><dd>Markup Coding Convention</dd></dl>
```

좋은 예.

```
<div>
  <p>
    <strong>Markup</strong> Coding Convention
  </p>
</div>
```



Note

inline 요소는 작업자가 선택적으로 줄바꿈 하는 것으로 한다.

### 6.2 공백 [Whitespaces]

닫기 엘리먼트 이후의 공백처럼 불필요한 공백이 코드에 포함되지 않도록 한다.

나쁜 예.

```
<a href="www.naver.com">네이버#</a>

<a href="www.naver.com">네이버</a>#####
```

좋은 예.

```
<a href="www.naver.com">네이버</a>
```



Note

불필요한 공백은 소스의 DIFF에 영향을 줄수 있다.

Editor에서 제공하는 "Invisible Character 보기" 기능을 활용하면 불필요한 공백이 있는지 쉽게 확인할수 있다.

### 6.3 따옴표 [HTML Quotation marks]

엘리먼트의 속성 값은 반드시 쌍따옴표(Double Quote)를 이용하여 감싼다.

나쁜 예.

```
<table width=100% cellpadding=0 border=1>
  ...
  ...
</table>

<button onclick='run();'>실행</button>
```

좋은 예.

```
<div id="header" class="logoless">
  ...
  ...
</div>

<button onclick="alert('실행');">실행</button>
```

## 6.4 빈줄(Empty Lines)

코드 그룹간에 빈줄을 추가하여 가독성을 높일수 있도록 한다.

빈줄은 2줄 이상 연속으로 추가하지 않는다.

## 6.5 주석 [Comments]

코드 그룹간 구분을 위해 주석을 추가하거나, 코드에 대한 설명이 필요한 경우 주석을 사용한다.

코드에 대한 설명은 가급적 길지 않게 하고, 주석을 사용하지 않더라도 예측이 가능한 네이밍을 이용해야한다.

```
<!-- header -->
<div id="header">
  <!-- 해당 서비스명으로 텍스트를 변경하세요. -->
  <h1>서비스명</h1>
  <p>나는 자랑스런 태극기 앞에..</p>
</div>

<!-- container -->
<div id="container">
  ...
</div>
```



### Note

주석의 지나친 사용은 코드의 가독성을 해칠수 있고, 파일의 용량을 무겁게 할수 있다.

## 7. CSS 작성 규칙 [CSS Style Rules]

### 7.1 유효성 검사 [CSS Validity]

CSS는 가능한 한 유효한 코드로 작성되어야 한다.

유효성 검사는 W3C CSS validator <sup>1</sup>를 기준으로 한다.

### 7.2 속성의 축약 [Shorthand Properties]

상하,좌우 값이 동일한 경우 속성값을 축약한다.

나쁜 예.

```
div {  
  margin: 0 0 0 0;  
  padding: 0 20px 0 20px;  
  border-style: solid;  
  border-width: 1px;  
  border-color: #000;  
}
```

좋은 예.

```
div {  
  margin: 0;  
  padding: 0 20px;  
  border: 1px solid #000;  
}
```

### 7.3 단위의 생략 [Omit Units]

속성값이 0인 경우 단위를 생략한다.

나쁜 예.

```
div {  
  width: 0px;  
  height: 0px;  
  background-position: 0px 0px;  
  font-size: 0em;  
  letter-spacing: 0px;  
}
```

좋은 예.

```
div {  
  width: 10px;  
  height: 0;  
  background-position: 0 0;  
  font-size: 0;  
  letter-spacing: 0;  
}
```

<sup>1</sup><http://jigsaw.w3.org/css-validator/>



```
}
```

## 7.4 색상값의 표기법 [Hexadecimal Notation]

동일한 값으로 이루어진 16진수 컬러 값은, 축약하여 3자리로 표기한다.

나쁜 예.

```
div {
  color: #666666;
  background-color: #ffdd00;
  border: 1px solid #ddccdd;
}
```

좋은 예.

```
div {
  color: #666;
  background-color: #fd0;
  border: 1px solid #dcd;
}
```

## 7.5 id/class 네이밍 규칙 [Id and class naming]

레이아웃에 사용되는 엘리먼트<sup>2</sup>와 품요소의 레이블 연결을 위해 id를 사용할 수 있으나, id 셀렉터를 이용한 style의 선언은 지양하도록 한다. id/class의 네이밍은 의미를 알수있는 단어를 사용하고 과도하게 축약하지 않도록 한다.

또한 특정 형태나 속성에 종속되는 네이밍은 스펙의 확장/변경시 대응이 용이치 않기 때문에 지양한다.

단어간의 구분자는 언더바(\_)를 사용하고, CamelCase나 하이픈(-)은 사용하지 않는다.

나쁜 예.

```
#wrap{
  .... //id 셀렉터를 이용한 style
}

.ly_atcp {
  .... //class명의 과도한 축약
}

.asideContents {
  .... //camel case 사용
}

.mt0 {
  margin-top: 0; //특정 속성에 종속되는 네이밍
}

.green_button {
  .... //특정 형태에 종속되는 네이밍
}
```

<sup>2</sup>#wrap, #header, #container, #contents, #footer

```
.btn5 {
  .... //의미를 알기 어려운 네이밍
}
```

좋은 예.

```
.wrap{
  ....
}

.auto_complete {
  ....
}

.aside_contents {
  ....
}

.column.column_first {
  ....
}

.button_login {
  ....
}
```

## 7.6 이미지 대체 기법 [Image Replacement]

이미지 대체 기법 사용시 콘텐츠의 내용에 부합하는 올바른 텍스트 엘리먼트를 제공해야하고, 텍스트 숨김처리를 위해 작성된 CSS코드는 접근성과 성능에 부합하는 코드로 작성되어야 한다.

나쁜 예.

```
/* 대체텍스트를 제공하지 않음 */
<button class="icon_write"></button>

/* 성능이슈가 발생할수 있는 CSS코드 */
.icon_write {
  background: url(...);
  width: 20px;
  height: 20px;
  text-indent: -999px;
}
```

좋은 예.

```
<button class="icon_write"><span class="blind">글쓰기</span></button>

<style>
.blind {
  position: absolute;
  clip: rect(0 0 0 0);
  width: 1px;
```

```
height: 1px;
margin: -1px;
overflow: hidden;
}
.icon_write {
background: url(...);
width: 20px;
height: 20px;
}
</style>
```



#### Note

성능상 критичный 이슈<sup>3</sup>가 발생할 수 있는 text-indent:-999px 또는 left:-999px 를 이용한 대체기법은 사용하지 않는다.

---

<sup>3</sup><http://yobi.navercorp.com/NaverSpeed-TF/NaverSpeed-TF/issue/16>

## 8. CSS 문법 규칙 [CSS Formatting Rules]

### 8.1 들여쓰기 [CSS Indentation]

모든 블록 콘텐츠<sup>1</sup>는 들여쓰기를 한다.

```
@media screen {
  .container {
    width: 100%;
    display: block;
  }
}

@keyframes showup {
  from {
    transform: translate3d(0, 20px, 0);
    opacity: 0;
  }

  to {
    transform: translate3d(0, 0, 0);
    opacity: 1;
  }
}
```

### 8.2 빈칸과 세미콜론 [Space and semicolon]

#### 빈칸

- 선택터와 중괄호({}) 사이에 빈칸을 1칸 추가한다.
- 가독성을 위해 속성과 속성값 사이의 콜론(:) 뒤에는 빈칸 1칸을 추가한다.
- 속성값이 콤마(,)로 구분될 때에는 콤마 뒤에 빈칸 1칸을 추가한다.

#### 세미콜론(,)

속성 값 이후에는 항상 세미콜론(;)을 추가한다.  
마지막 속성 값 이후에도 세미콜론을 생략하지 않는다.

틀린 예.

```
.wrap{
  background-color:rgba(255,255,255,1);
  width:100%;
  height:100%
}
```

좋은 예.

```
.wrap {
```

<sup>1</sup><http://www.w3.org/TR/CSS21/syndata.html#block>

```
background-color: rgba(255, 255, 255, 1);
width: 100%;
height: 100%;
}
```



#### Note

적절한 빈칸의 추가는 가독성 향상 효과가 있다.

공백과 세미콜론등을 제거하여 성능 최적화하는 방법보다는 배포시 minify 처리하는것이 더 효율적이다.

## 8.3 따옴표 [CSS Quotation Marks]

- 속성 선택터(attribute selectors)와 속성값은 홑따옴표를 사용한다.
- url속성값은 따옴표를 넣지 않는다<sup>2</sup>
- 예외로, @charset 선언시에는 반드시 쌍따옴표를 사용한다.<sup>3</sup>

틀린 예.

```
@charset 'utf-8';
input[type="radio"] {
  background: url('http://img.naver.net/logo.png') no-repeat;
  font-family: "nanum gothic", sans-serif;
}
```

좋은 예.

```
@charset "utf-8";
input[type='radio'] {
  background: url(http://img.naver.net/logo.png) no-repeat;
  font-family: 'nanum gothic', sans-serif;
}
```

<sup>2</sup>URI값의 따옴표 추가는 선택사항이다. <http://www.w3.org/TR/CSS2/syndata.html#value-def-uri>

<sup>3</sup>@charset 표기시 반드시 쌍따옴표를 사용하는것이 스펙 <http://www.w3.org/TR/CSS21/syndata.html#charset>