

# 개발자 도구

Developer tools

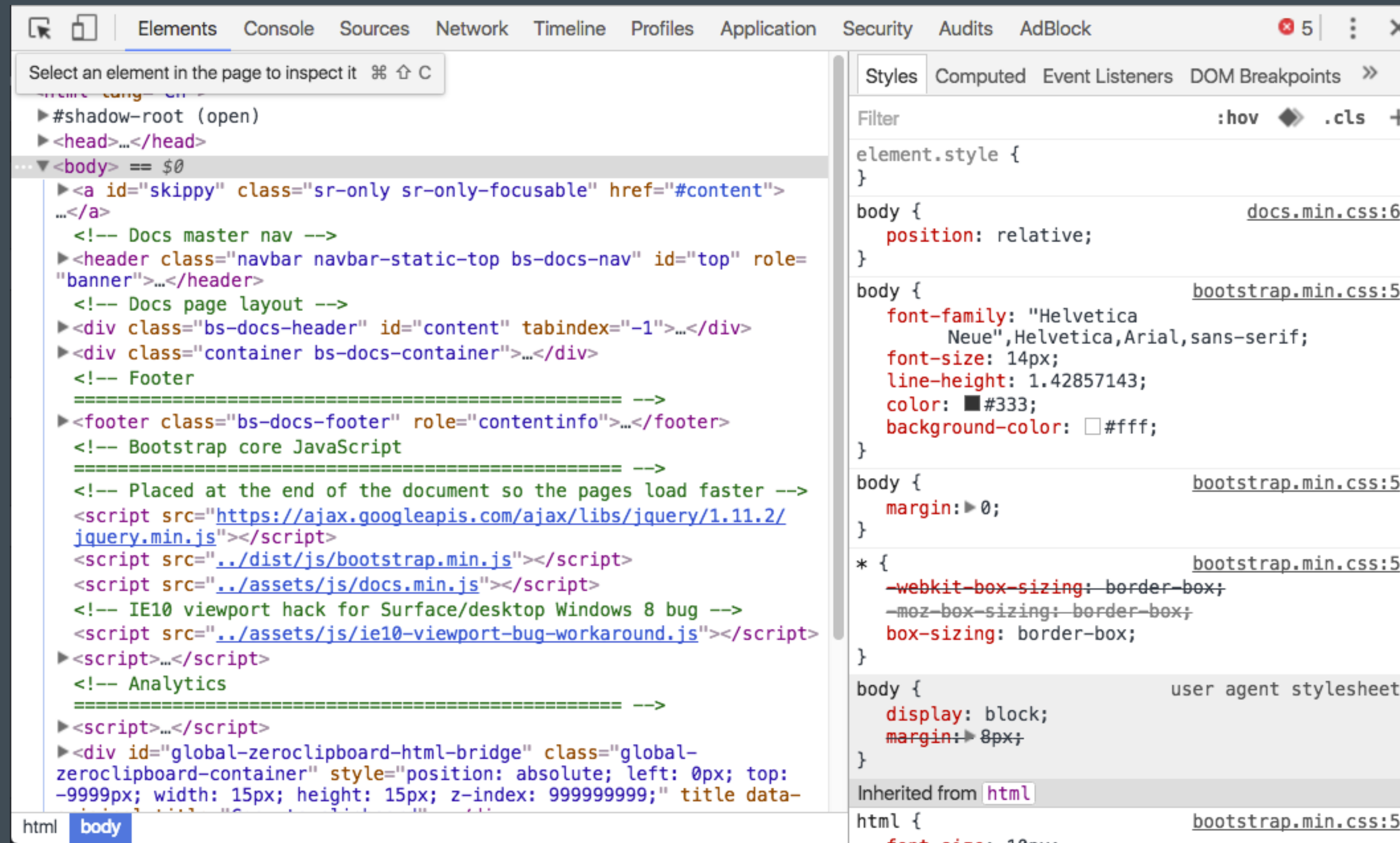
브라우저에서 HTML/CSS요소를 디버깅하는 법에 대해 알아봅니다

# 요소(Elements)

IE의 경우 DOM 탐색기 또는 HTML, 파이어폭스의 경우 검사기(Inspector)

F12 or Ctrl + Shift + I  
Cmd + Alt + I

현재 렌더링 된  
HTML요소들을 보여줍니다



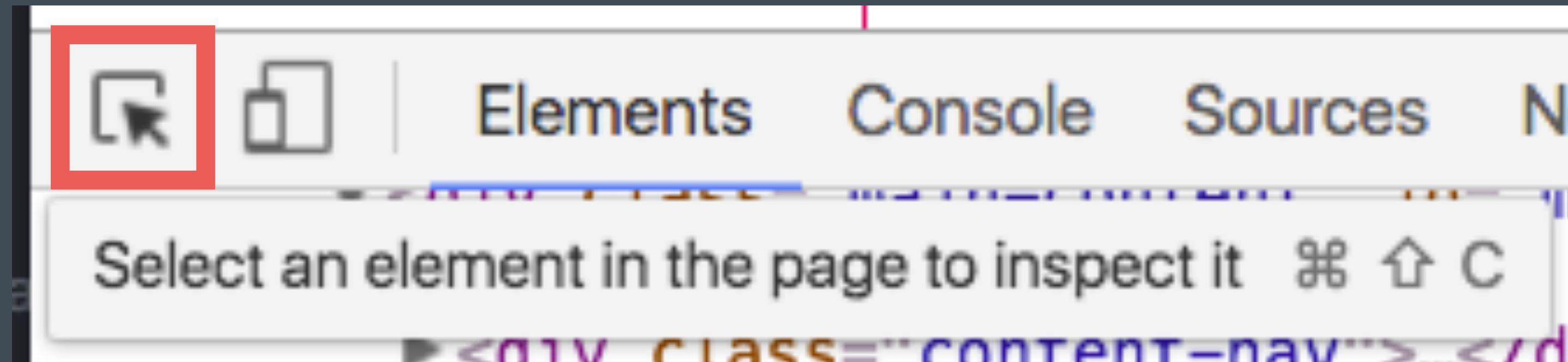
선택한 요소에 대한  
CSS값을 보여줍니다

우측에서 직접 CSS속성을 추가/변경/삭제하며 실시간으로 결과를 확인할 수 있습니다.  
좌측에서 직접 HTML요소를 추가/변경하는것도 가능합니다.

# 요소(Elements)

IE의 경우 DOM 탐색기 또는 HTML, 파이어폭스의 경우 검사기(Inspector)

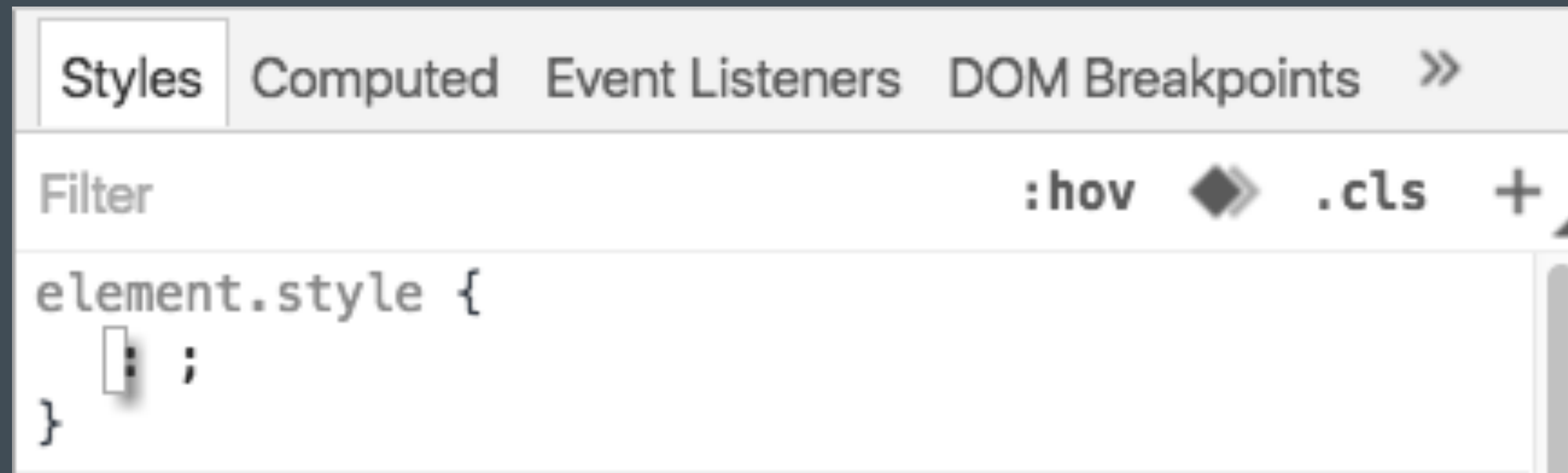
요소 선택  
cmd + shift + c



자주 사용하는 단축키이며  
외워두시는게 좋습니다

# 요소(Elements) - 스타일(Styles)

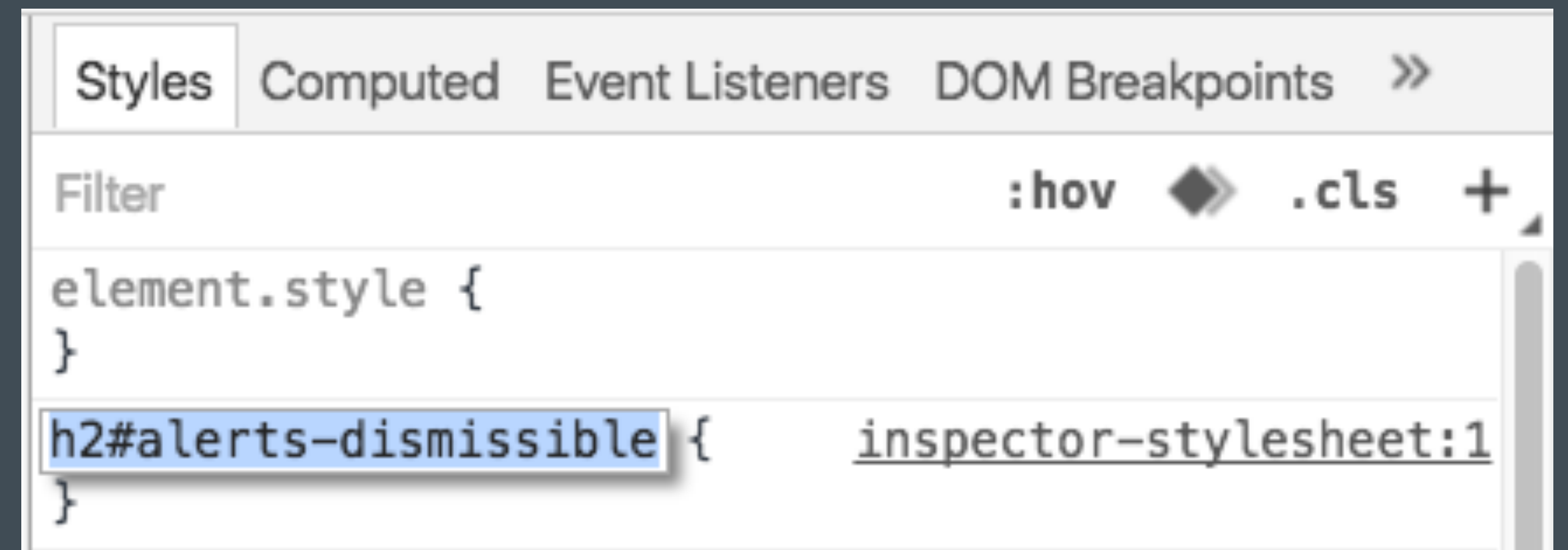
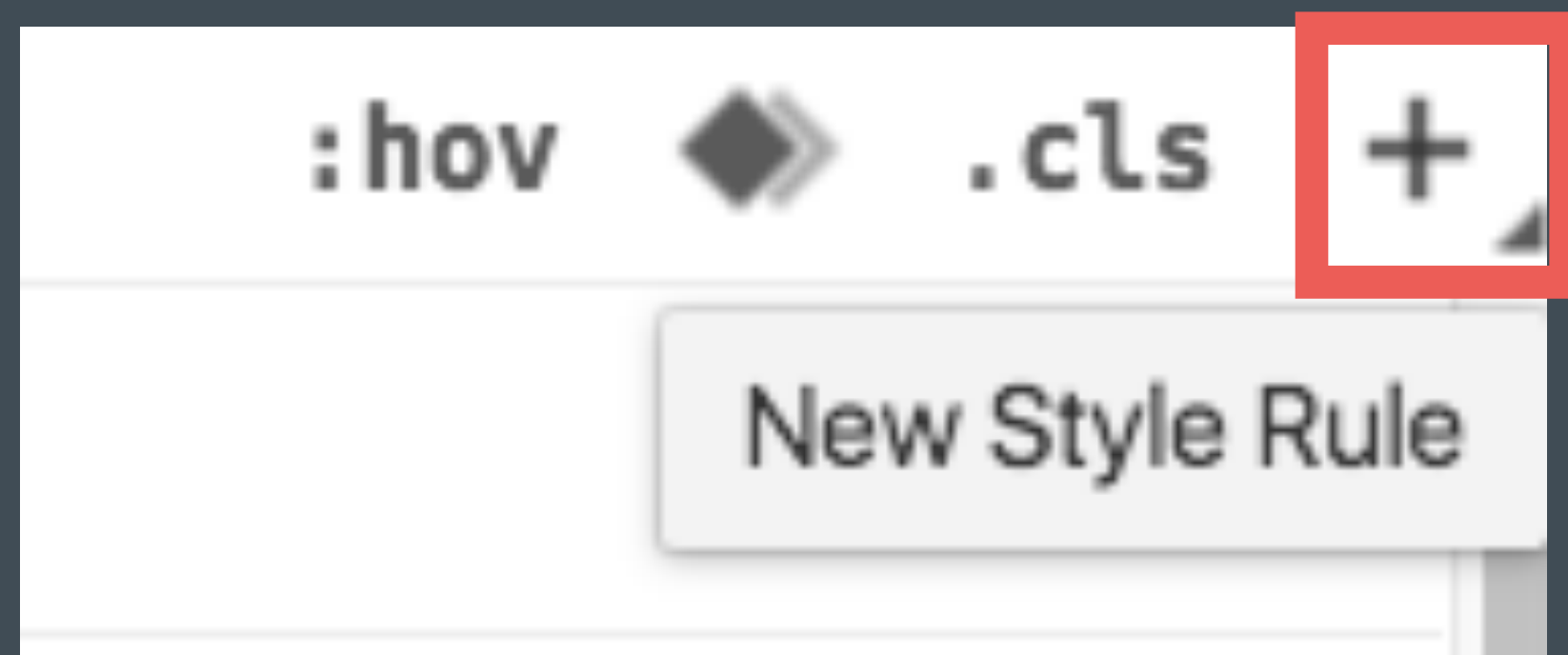
IE의 경우 DOM 탐색기 또는 HTML, 파이어폭스의 경우 검사기(Inspector)



element.style에 넣은 값은 Inline스타일로 적용됩니다.

# 요소(Elements) - 스타일(Styles)

IE의 경우 DOM 탐색기 또는 HTML, 파이어폭스의 경우 검사기(Inspector)

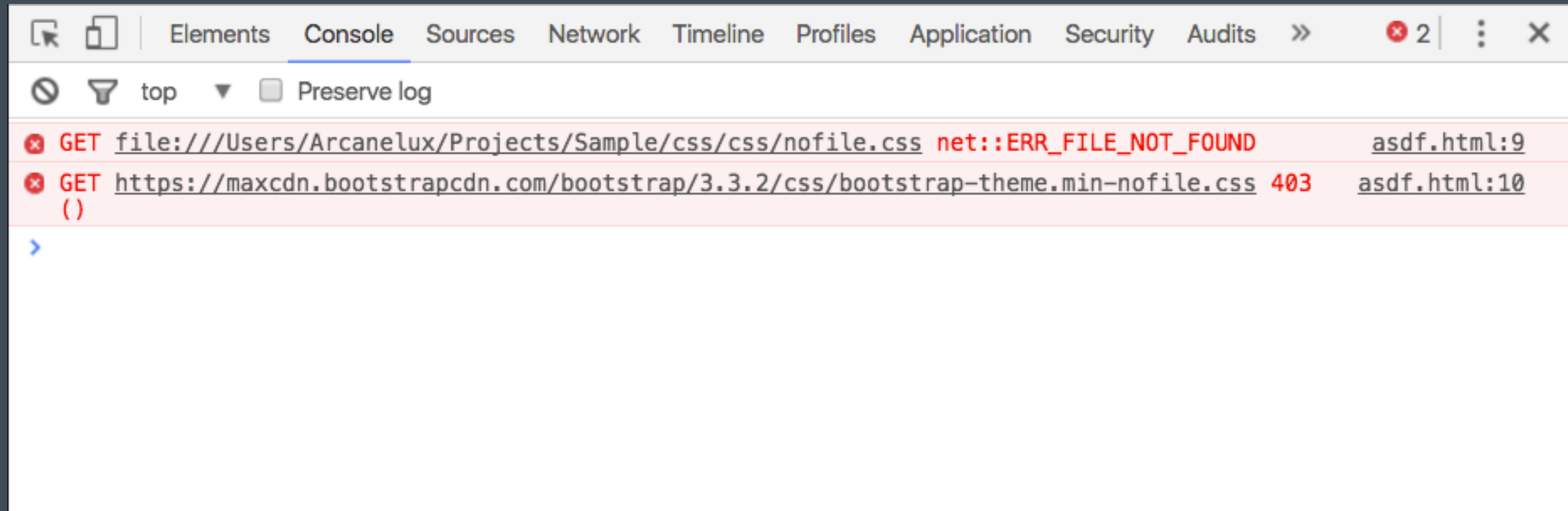


New Style Rule버튼을 누르면, 현재 요소를 특정하는 선택자를 자동으로 생성해줍니다.



# 콘솔(Console)

실시간으로 내부의 자바스크립트를 테스트하거나, 로그를 출력해줍니다

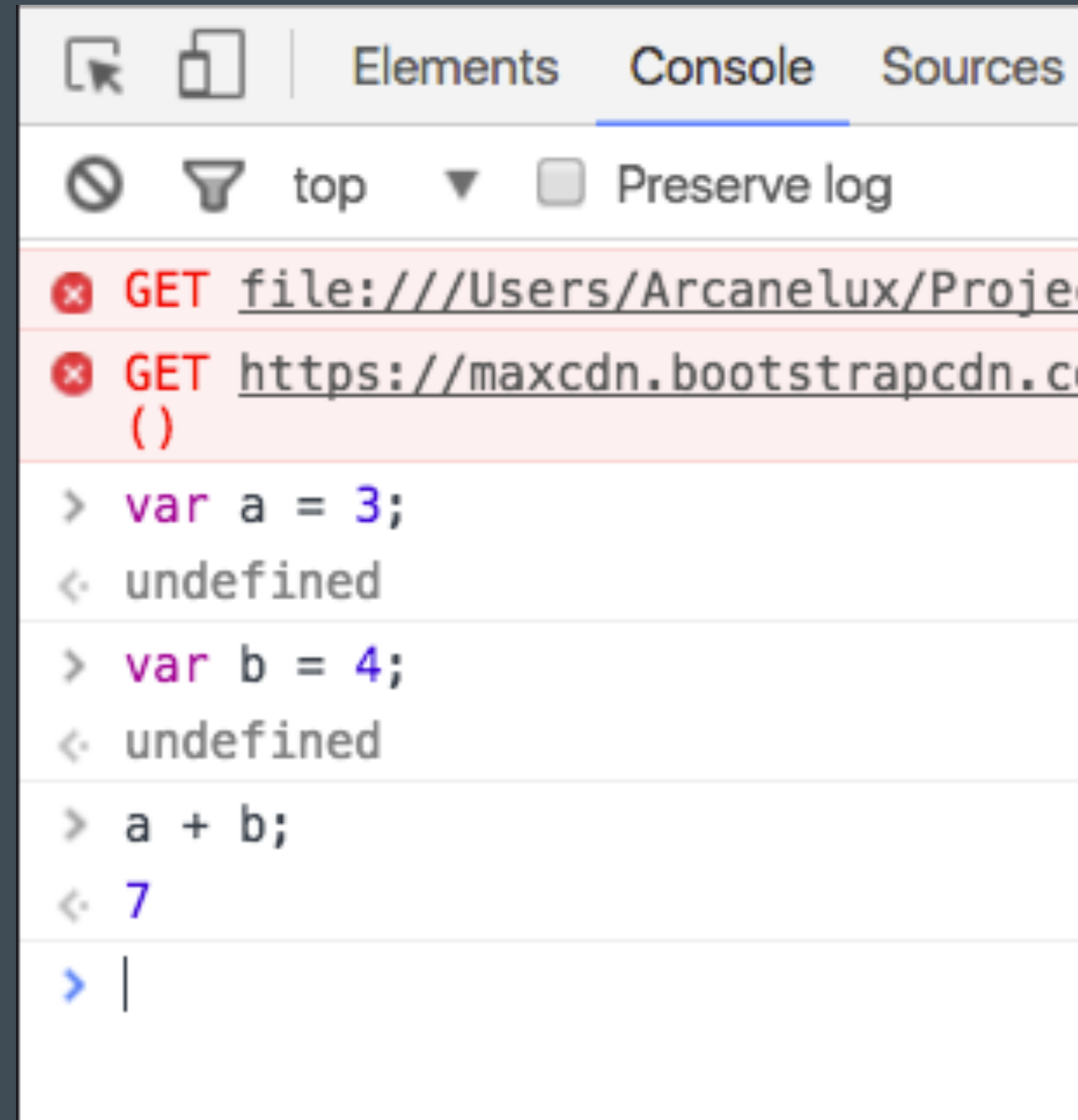


링크로 불러오려는 파일이 없는 경우, 이곳에 에러메세지가 표시됩니다.

css파일이 적용이 안되거나, 이미지파일이 출력되지 않을 경우 먼저 Console화면을 확인해보는것이 좋습니다.

# 콘솔(Console)

실시간으로 내부의 자바스크립트를 테스트하거나, 로그를 출력해줍니다



파이썬 인터프리터와 비슷하게, 현재 브라우저에서 자바스크립트를 실행할 수 있습니다.  
HTML페이지에서 자바스크립트 파일을 로드한 경우, 로드한 모듈도 사용 가능합니다.

# Form 요소

form

데이터를 입력하거나 전송할 때 사용하는 HTML요소에 대해 알아봅니다



# form 요소

form

## form태그

```
<form action="" method="get">  
  <label for="username">ID</label>  
  <input type="text" name="username">  
</form>
```

form은 브라우저(클라이언트)에서 서버로 데이터를 전송하기 위해 사용하는 태그입니다.

# form - method

method 속성

## method

폼에서 서버로 데이터를 전송하는 방식을 결정

### GET

URL에 데이터를 담아 전달

### POST

URL과는 별도로 데이터를 전달

아이디/패스워드와 같은 중요한 정보는 GET방식으로 전달하지 않습니다.

# form - action

action 속성

**action**

form에서 데이터를 전송할 URL

# input 요소

input types

input의 type들

```
<input type="text" id="username">  
<input type="password" id="password">  
<input type="radio" id="radio">  
<input type="checkbox" id="checkbox">  
<input type="button">  
<input type="file" id="file">  
<input type="submit">  
<input type="reset">  
<input type="hidden" id="hidden" value="hiddenValue">
```

데이터 전송  
폼 초기화

input요소는 값을 가지거나, form에 영향을 줍니다

# input 요소

input 주요 속성들

input의 속성들

```
<input type="text" value="disabled" disabled>  
<input type="text" value="readonly" readonly>  
<input type="text" size="3">  
<input type="text" maxlength="10">  
<input type="checkbox" checked="checked">  
<input id="radio1" type="radio" name="agree" checked="checked">  
<input id="radio2" type="radio" name="agree">
```

같은 이름을 가진 radio요소는 동시에 체크되지 않습니다

input요소는 값을 가지거나, form에 영향을 줍니다

# select 요소

select

select 태그

```
<select name="number" id="select-id">  
  <option value="1">First</option>  
  <option value="2">Second</option>  
  <option value="3">Third</option>  
  <option value="4">Fourth</option>  
</select>
```

select태그는 여러개의 주어진 값 중 일부를 선택하는 역할을 합니다



# select 요소

select 요소의 속성

multiple 속성

```
<select multiple="multiple">  
  <option value="apple">Apple</option>  
  <option value="banana">Banana</option>  
  <option value="orange">Orange</option>  
</select>
```

기본적으로 select요소는 option의 값을 한 개만 선택할 수 있습니다.

multiple속성을 가질 경우, ctrl (맥의 경우 cmd), shift키를 이용해 여러개의 값을 선택할 수 있습니다.

# select 요소

select 요소의 속성

size 속성

```
<select size="2">  
  <option value="apple">Apple</option>  
  <option value="banana">Banana</option>  
  <option value="orange">Orange</option>  
</select>
```

한 번에 option을 몇 개 보여줄지 정합니다.

# button 요소

button types

button 요소

```
<button type="submit">submit type button</button>
```

```
<button type="reset">reset type button</button>
```

```
<button type="button">button type button</button>
```

button요소는 input요소의 같은 type을 대체할 수 있습니다.

# fieldset, legend 요소

fieldset, legend

fieldset의 제목을 나타냅니다

form요소들을 담는  
틀을 나타냅니다

fieldset과 legend 요소

```
<fieldset>
  <legend>Login</legend>
  <label>username : <input type="text"></label>
  <label>password : <input type="password"></label>
</fieldset>
```

legend요소는 fieldset의 첫 번째 자식으로 사용해야합니다.  
fieldset은 다른 fieldset을 중첩해서 자식으로 가질 수 있습니다.

배운내용을 복습하며 문서를 작성해봅시다

form-default.html

form-styled.html / form-styled.css

# Sass

Syntactically Awesome Stylesheet

구문적 놀라운 스타일 시트(...)



# Sass는?

Syntactically Awesome Stylesheet

## CSS전처리기 (Pre-processor)

Sass와 같은 'CSS확장 언어'의 파일을 웹 브라우저에서 해석할 수 있는  
css파일로 만들어주는 처리기

# Sass 기본구조

Sass basic structure

중괄호를 쓰지 않음

```
div.container  
  padding: 15px  
  margin: 0
```

요소간 상하관계는 들여쓰기로 구분

```
p#main-title  
  font-size: 16px  
  font-weight: bold
```

```
.fixed  
  position: fixed  
  bottom: 10px  
  right: 10px
```

# Sass의 출력 스타일 - expanded

Sass output style

Sass

```
div.container
  padding: 15px
  margin: 0

p#main-title
  font-size: 16px
  font-weight: bold

.fixed
  position: fixed
  bottom: 10px
  right: 10px
```

CSS

```
div.container {
  padding: 15px;
  margin: 0;
}

div.container p#main-title {
  font-size: 16px;
  font-weight: bold;
}

div.container .fixed {
  position: fixed;
  bottom: 10px;
  right: 10px;
}
```

# Sass의 출력 스타일 - nested

Sass output style

Sass

```
div.container
  padding: 15px
  margin: 0

p#main-title
  font-size: 16px
  font-weight: bold

.fixed
  position: fixed
  bottom: 10px
  right: 10px
```

CSS

```
div.container {
  padding: 15px;
  margin: 0; }
div.container p#main-title {
  font-size: 16px;
  font-weight: bold; }
div.container .fixed {
  position: fixed;
  bottom: 10px;
  right: 10px; }
```

# Sass의 출력 스타일 - compact

Sass output style

Sass

```
div.container
  padding: 15px
  margin: 0

p#main-title
  font-size: 16px
  font-weight: bold

.fixed
  position: fixed
  bottom: 10px
  right: 10px
```

CSS

```
div.container { padding: 15px; margin: 0; }
div.container p#main-title { font-size: 16px; font-weight: bold; }
div.container .fixed { position: fixed; bottom: 10px; right: 10px; }
```

# Sass의 출력 스타일 - compressed

Sass output style

Sass

```
div.container
  padding: 15px
  margin: 0

p#main-title
  font-size: 16px
  font-weight: bold

.fixed
  position: fixed
  bottom: 10px
  right: 10px
```

CSS

```
div.container{padding:15px;margin:0}div.container p#main-title{font-size:16px;font-weight:bold}div.container .fixed{position:fixed;bottom:10px;right:10px}
```



# sass-autocompile

atom package

atom 패키지 설치  
sass-autocompile

# sass-autocompile

atom packages - Settings

Compile on Save

Compile with 'compressed'

Compile with 'compact'

Compile with 'nested'

Compile with 'expanded'

저장시 자동 컴파일

Compressed 방식 파일 생성

Compact 방식 파일 생성

Nested 방식 파일 생성

Expanded 방식 파일 생성

# sass-autocompile

atom packages - Settings

상대 경로 지정  
'../css/\$1.css'

# Sass 문법

Sass syntax

Sass 사용법에 대해 간단히 알아봅니다

주석

Comments

//주석내용

또는

/\* 주석내용 \*/ (CSS기본)  
을 사용합니다

# 주석

## Comments

```
// 한 줄 주석은 Sass에서만 지원하며,  
// CSS컴파일 시 나타나지 않습니다.  
div.container  
  padding: 15px  
  margin: 0  
  
/* 기존에 CSS에서 사용하는 주석 형태는 컴파일 시에도 나타납니다 */  
p#main-title  
  font-size: 16px  
  font-weight: bold  
  
/*  
  여러 줄 주석은  
  기존 CSS주석 형태로만 사용 가능합니다  
*/  
.fixed  
  position: fixed  
  bottom: 10px  
  right: 10px
```



# 주석

Comments

```
@charset "UTF-8";
div.container {
    padding: 15px;
    margin: 0;
    /* 기존에 CSS에서 사용하는 주석 형태는 컴파일 시에도 나타나게 됩니다 */
    /*
        여러 줄 주석은
        기존 CSS주석 형태로만 사용 가능합니다
    */
}

div.container p#main-title {
    font-size: 16px;
    font-weight: bold;
}

div.container .fixed {
    position: fixed;
    bottom: 10px;
    right: 10px;
}
```

# 중첩

Nested

중괄호를 쓰지 않음

```
.container  
  width: 1100px  
  margin: 0 auto
```

한 번의 들여쓰기마다  
한 단계씩의  
하위 요소가 됨을  
뜻합니다

```
#page  
  width: 800px  
  border: 1px solid #eee  
  padding: 10px
```

```
p.section-title  
  font-size: 20px  
  text-align: center
```

```
p.section-content  
  font-size: 12px  
  color: #999
```

# 부모 참조 선택자 (&)

Referencing parent selectors

```
#page
width: 800px
border: 1px solid #eee
padding: 10px
```

Sass

a

```
text-decoration: none
&:hover
color: red
```

```
.link-container &
font-size: 30px
```

&은 자신의  
부모선택자를  
참조합니다

중괄호를 쓰지 않음

```
#page {
width: 800px;
border: 1px solid #eee;
padding: 10px;
}
```

```
#page a {
text-decoration: none;
}
```

```
#page a:hover {
color: red;
}
```

```
.link-container #page a {
font-size: 30px;
}
```

CSS

& 앞에 선택자가 있을 경우,  
해당 선택자 다음에 현재 위치에서의 부모선택자를 참조합니다

# 중첩 속성

Nested properties

Sass

```
.container  
  margin:  
    left: auto  
    right: auto
```

속성 다음에 -(dash)  
로 이어지는 속성은  
아래로 내려  
작성할 수 있습니다

CSS

```
.container {  
  margin-left: auto;  
  margin-right: auto;  
}
```

# 선택자 상속 (@extend)

Selector inheritance

```
.btn
  background-color: #cdcdcd
  font-weight: bold
  color: white
  padding: 5px 20px
```

Sass

```
.btn-ok
  @extend .btn
  background-color: #d9edf7
```

```
.btn-cancel
  @extend .btn
  background-color: #bbb
```

상속받은 선택자가 .btn의 설정을 함께 공유합니다

```
.btn, .btn-ok, .btn-cancel {
  background-color: #cdcdcd;
  font-weight: bold;
  color: white;
  padding: 5px 20px;
}
```

CSS

```
.btn-ok {
  background-color: #d9edf7;
}

.btn-cancel {
  background-color: #bbb;
}
```

상속을 사용하면 같은 속성에서 일부만 바뀐 요소를 쉽게 컨트롤 할 수 있습니다.

# 대체 선택자 (%)

Placeholder selector

```
%btn
  background-color: #cdcdcd
  font-weight: bold
  color: white
  padding: 5px 20px
```

```
Sass .btn-ok
      @extend %btn
      background-color: #d9edf7
```

```
.btn-cancel
  @extend %btn
  background-color: #bbb
```

대체선택자로 선언된 구문은 CSS로 해석되지 않습니다

%btn에 지정한 속성은 CSS에 나타나지 않습니다

```
.btn-ok, .btn-cancel {
  background-color: #cdcdcd;
  font-weight: bold;
  color: white;
  padding: 5px 20px;
}
```

CSS

```
.btn-ok {
  background-color: #d9edf7;
}

.btn-cancel {
  background-color: #bbb;
}
```

# 변수 (\$)

Variables

```
$padding: 10px  
$bg-color: #ecec  
$title-font-weight: bold
```

변수명은 \$로 시작합니다

```
div.container  
  background-color: $bg-color  
  padding: $padding  
  margin: 0
```

자주 쓰이는 color값을 저장합니다

```
p#main-title  
  font-size: 16px  
  font-weight: $title-font-weight
```

중복으로 쓰이는 값에 유용합니다

```
p#sub-title  
  font-size: 14px  
  font-weight: $title-font-weight
```

```
.fixed  
  position: fixed  
  padding: $padding  
  bottom: 10px  
  right: 10px
```

# 변수의 데이터 유형

Variables

유형	설명
숫자 (Number)	정수, 소수, 단위숫자 ex) 1.0, 2.3em, 50px
문자 (String)	작은따옴표 혹은 큰따옴표로 감싸진 문자 ex) 'images/025.png', "Sass사용법"
색상 (Colors)	색상 정보 ex) #ff2342, rgb(255,233,210)
참/거짓값 (Boolean)	참 또는 거짓 값 true, false
리스트 (List)	공백이나 콤마로 구분해서 나열 2px solid gray, Helvetica, Sans-serif
맵 (Map)	파이썬의 dict와 유사 ex: {color: white, font-size: 12px}
비어있음 (Null)	값 초기화시 사용



# Sass파일 호출

## Import

현재 파일 위치를 기준으로 불러옵니다

```
@import 'variables'
```

```
%btn
```

```
background-color: $brand-primary
```

```
font-weight: bold
```

```
color: white
```

```
padding: 5px 20px
```

CSS파일을 불러올 때는 확장자를 입력하여야 하지만, Sass파일을 불러올 때는 확장자를 입력하지 않아도 됩니다.