

# 파이썬 웹 크롤링



# 웹 크롤링

- 스크래핑(scraping)

- HTTP를 통해 웹 사이트의 내용을 긁어다 원하는 형태로 가공하는 것
- 웹 사이트의 데이터를 수집하는 모든 작업

- 크롤링(crawling)

- 인터넷 상에 존재하는 자료를 스크래핑(크롤링)을 통해 수집하여 데이터를 파싱하여 원하는 정보를 추출하는 것
- 크롤러는 조직적, 자동화된 방법으로 웹을 탐색하는 프로그램으로 크롤러가 하는 작업을 크롤링이라고 함

- 파싱(parsing)

- 웹 페이지에서 원하는 데이터를 특정 패턴이나 순서로 추출하여 정보를 가공하는 것



# 웹 크롤링 예

```

1 from urllib.request import urlopen
2 from bs4 import BeautifulSoup
3
4 def request(url):
5     response = urlopen(url)
6     byte_data = response.read()
7     text_data = byte_data
8     #text_data = byte_data.decode('utf-8')
9     return text_data
10
11 def select_rank(data):
12     bs = BeautifulSoup(data, 'html.parser')

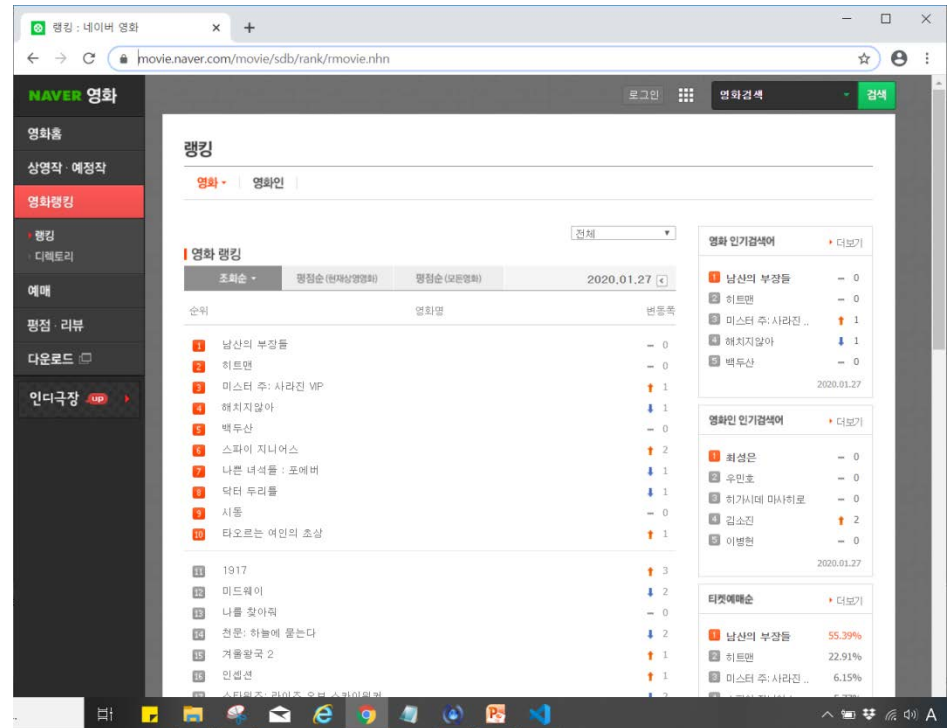
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

1 위: 남산의 부장들
2 위: 히트맨
3 위: 미스터 주: 사라진 VIP
4 위: 해치지말아
5 위: 백두산
6 위: 스파이 지니어스
7 위: 나쁜 녀석들 : 포에버
8 위: 닥터 두리틀
9 위: 시동
10 위: 타오르는 여인의 초상
11 위: 1917
12 위: 미드웨이
13 위: 나를 찾아줘
14 위: 천문: 하늘에 묻는다
15 위: 겨울왕국 2
16 위: 인센션
17 위: 스타워즈: 라이즈 오브 스카이워커
18 위: 미성년
19 위: 기생충
20 위: 악인전
21 위: 포드 v 페라리
22 위: 하이큐!! 땅 vs 하늘
23 위: 사마에게
24 위: 사바하
25 위: 가장 보통의 연애
26 위: 오즈의 마법사: 요술구두와 말하는 책
27 위: 닥터 슬립
28 위: 터미네이터: 다크 페이트

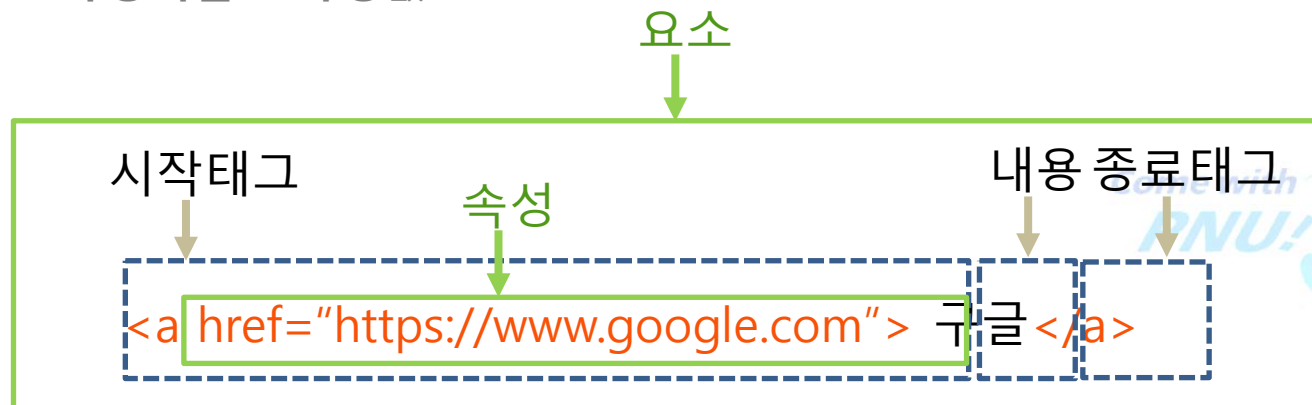
```



# 웹 페이지 기초(HTML)

# HTML(HyperText Markup Language)

- 마크 업을 사용하여 웹 페이지의 구조를 설명
- HTML 태그(Tag)
  - HTML 문서를 구성하는 명령어
  - <태그>로 작성
- HTML 요소(Element)
  - 시작태그와 종료태그 사이의 모든 내용
- HTML 속성(Attribute)
  - 요소의 추가 정보를 제공
  - 시작태그에 추가
  - 속성이름="속성값 "



# HTML 기본 태그 정리

태그(Tag)	설명	비고
<!-- -->	주석처리	
<!DOCTYPE html>	문서 타입 정의 (HTML5문서 정의)	
<a>	하이퍼링크 정의	
<body>	문서의 본문 정의	
 	줄바꿈 태그로 해당 태그를 만나면 줄을 바꿈	종료 태그 없음
<div>	문서의 섹션정의	
<h1>~<h6>	제목을 나타내면 <h1>가 제일 중요한 제목	자동 단락 나눔
<head>	문서에 관한 정보 정의	
<hr>	수평줄을 그어줌	종료 태그 없음
<img>	이미지 표시	
<li>	목록의 아이템 표시	
<ol>	순서가 있는 목록태그로 아이템은 <li>태그로 표시	
<p>	문단태그로 단락을 나눔	
<title>	문서 타이틀 정의(웹 브라우저의 툴바나 타이틀바에 표시)	
<ul>	순서가 없는 목록태그로 아이템은 <li>태그로 표시	

# 블록(Block)/인라인(Inline)요소

- 블록(Block) 요소

- 줄 바꿈이 일어나는 형태로 영역의 너비가 상위 영역의 전체 너비  
를 사용
- <div>
- <p>,<ul>,<ol>,<li>,<h1>~<h6>

- 인라인(Inline) 요소

- 새 줄에서 시작되지 않고 필요한 만큼 너비를 차지
- <span>,<a>,<img>

```
<div style="background-color:yellow">블록(inline) 요소 </div>  
<span style="background-color:blue">인라인(inline)</span>요소
```



블록(inline) 요소

인라인(inline) 요소



# 시맨틱 웹(Semantic Web)

- 시맨틱 웹(Semantic Web)

- 의미론적인 웹이라는 뜻
- 컴퓨터가 웹사이트를 단순한 코드의 구성이 아닌 의미를 가진 사이트라는걸 알 수 있게 만드는 것
- HTML5 시맨틱 태그(Semantic tag) 지원
  - 컴퓨터가 정보를 이해하고, 논리적인 추론까지 할 수 있는 구조를 만들기 위해 추가된 태그



# HTML5 시맨틱 태그(Semantic tag)

- **<header>**
  - 페이지나 섹션의 머리말 표현
  - 페이지 제목, 페이지를 소개하는 간단한 설명
- **<nav>**
  - 하이퍼링크들을 모아 놓은 특별한 섹션
  - 페이지 내 목차를 만드는 용도
- **<section>**
  - 문서의 장(chapter, section) 혹은 절을 구성하는 역할
  - 일반 문서에 여러 장이 있듯이 웹 페이지에 여러 <section> 가능
  - 제목태그(<h1>~<h6>)를 사용하여 절 혹은 섹션의 주제 기입
- **<article>**
  - 본문과 연관 있지만, 독립적인 콘텐츠를 담는 영역
  - 혹은 보조 기사, 블로그 포스트, 댓글 등 기타 독립적인 내용
  - <article>에 담는 내용이 많은 경우 여러 <section> 둘 수 있음
- **<aside>**
  - 본문에서 약간 벗어난 노트나 팁
  - 신문, 잡지에서 주요 기사 옆 관련 기사, 삽입 어구로 표시된 논평 등
  - 페이지의 오른쪽이나 왼쪽에 주로 배치
- **<footer>**
  - 꼬리말 영역, 주로 저자나 저작권 정보



# 실습



The screenshot shows a web browser window with the title 'HTML연습'. The address bar contains the text 'Google에서 검색하거나 URL을 입력하세요.' Below the address bar, the main heading is '자유 소프트웨어(Free Software)'. There are three bullet points with links: '원칙1', '원칙2', and '원칙3'. The text below explains that free software has no restrictions on use, modification, or distribution, and that its source code is publicly available. It also notes that while most free software is provided for free, it is not the same as 'free' in the sense of 'free of charge'. At the bottom, there is a logo for the 'FREE SOFTWARE FOUNDATION'.

HTML연습

Google에서 검색하거나 URL을 입력하세요.

## 자유 소프트웨어(Free Software)

- [원칙1](#)
- [원칙2](#)
- [원칙3](#)

소프트웨어의 소스가 공개되어 사용, 수정, 배포 등에 제한이 없는 소프트웨어입니다.

즉, 소스 코드가 공개되어 누구나 소프트웨어를 자유롭게 수정할 수 있고, 자유롭게 복제 및 배포가 가능한 소프트웨어입니다.

자유 소프트웨어는 대부분 무료로 제공됨으로 무료로 제공되는 프리웨어와 혼동이 되지만 전혀 다른 개념입니다.

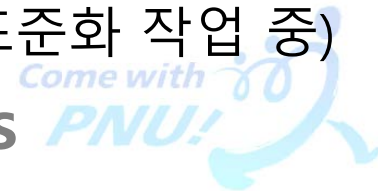
- 소프트웨어의 작동 원리를 연구하고 이를 자신의 필요에 맞게 변경시킬 수 있는 자유
- 소프트웨어를 이웃과 함께 공유하기 위해서 이를 복제하고 배포할 수 있는 자유
- 소프트웨어를 향상시키고 이를 공동체 전체의 이익을 위해서 다신 환원시킬 수 있는 자유

 **FREE SOFTWARE**  
FOUNDATION

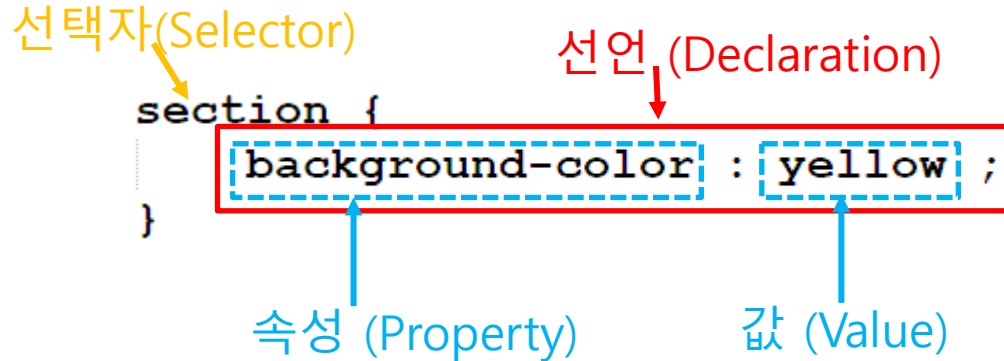
# 웹 페이지 기초(CSS)

# CSS3 스타일 시트

- CSS(Cascading Style Sheet)
  - CSS로 작성된 코드를 스타일 시트(style sheet)라고 부름
  - HTML 문서의 색이나 모양 등 외관을 꾸미는 언어
    - 디자인, 레이아웃 및 다양한 장치 및 화면 크기에 대한 디스플레이의 변형을 포함하여 웹 페이지의 스타일을 정의하는데 사용
  - 현재 CSS3 : CSS level 3
    - CSS1 -> CSS2 -> CSS3 -> CSS4(현재 표준화 작업 중)
  - <https://www.w3schools.com/css>



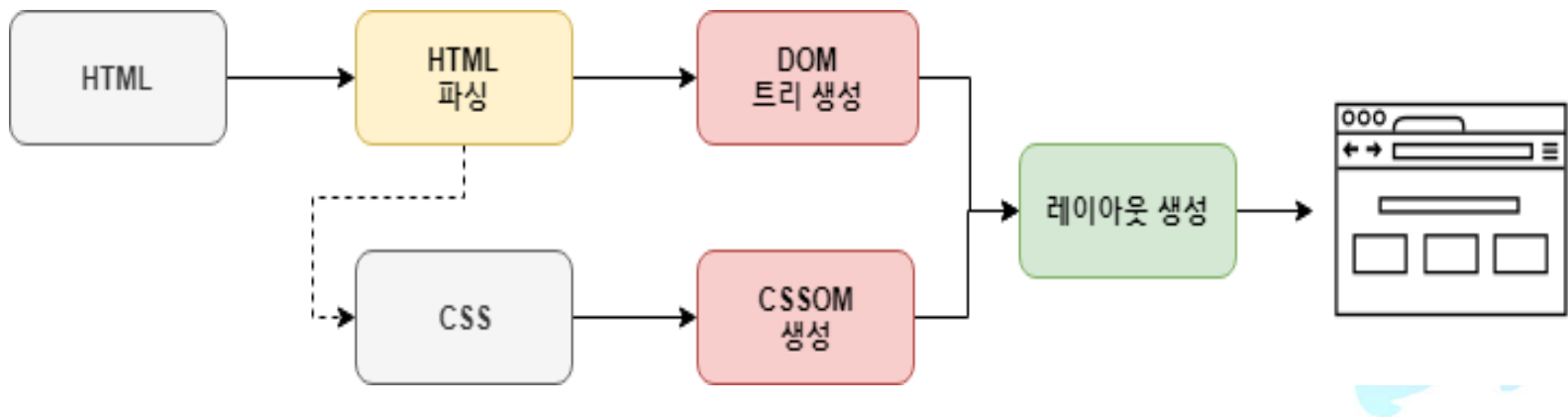
# CSS의 구성



- **선택자(selector)**
  - 스타일을 고칠 HTML 요소
  - 고칠 스타일은 여러 개의 선언으로 지정가능
  - 반드시 `}`로 묶어야 함
- **선언(declaration)**
  - 콜론(:)으로 구분 된 CSS 속성 이름과 값이 포함
  - 세미콜론(;)으로 끝남

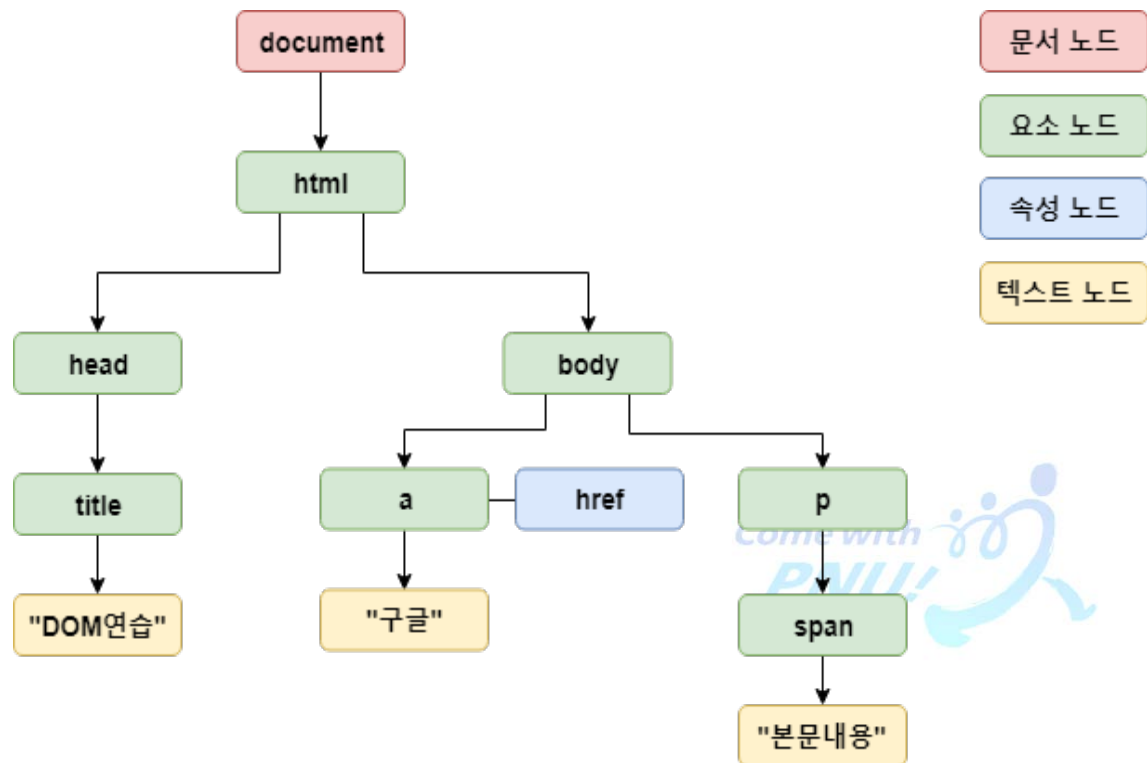
# DOM (문서 객체 모델; Document Object Model)

- 문서의 구조화된 표현을 제공하며 웹 페이지의 객체 지향 표현
- 구조화된 문서는 DOM을 사용하여 트리 구조를 얻어낼 수 있음



# DOM 예시

```
1 <html>
2   <head>
3     <title>DOM연습</title>
4   </head>
5   <body>
6     <a href="http://goole.com">구글</a>
7     <p>
8       <span>본문내용</span>
9     </p>
10  </body>
11 </html>
```



# 선택자의 종류

- **요소 선택자(Element Selector)**
  - HTML 요소 이름을 기반으로 요소 선택
  - HTML 태그 이름사용
  - 예) `section { ... }`
  - `*`는 전체 요소 지정
- **클래스 선택자(Class Selector)**
  - `class` 속성을 가진 요소를 선택
  - 마침표(.) 다음에 `class`속성의 값 사용
  - 예) `.c1 { ... }`
- **아이디 선택자(ID Selector)**
  - `id` 속성을 가진 요소를 선택
  - 해시(#) 다음에 `id`속성의 값 사용
  - 예) `#s1 { ... }`





# 선택자의 종류

- 속성 선택자

- 각 태그가 가지고 있는 그 속성에 접근하는 방식

- 종류

- 태그[속성]

- 속성 이름에 해당되는 속성을 가진 태그를 선택
- 예) `a[href] { .. }`

- 태그[속성="속성값"]

- 속성이 속성값인 태그를 선택
- 예) `a[href="#m1"] {...}`

- 태그[속성\*="속성값"]

- 지정된 어트리뷰트 값을 포함하는 요소를 선택
- 예) `a[href*="#m"] {...}`

- 태그[속성^="속성값"]

- 지정된 어트리뷰트 값으로 시작하는 요소를 선택
- 예) `a[href^="#m"] {...}`

- 태그[속성\$="속성값"]

- 지정된 어트리뷰트 값으로 시작하는 요소를 선택
- 예) `a[href$="1"] {...}`



# 선택자의 종류

- 형제 선택자

- 형제 관계(동위 관계)에서 뒤에 위치하는 요소를 선택

- 종류

- 선택자A + 선택자B

- 선택자A의 형제 요소 중 선택자A 바로 뒤에 위치하는 선택자B 요소를 선택
    - 예) nav + section { ... }

- 선택자A ~ 선택자B

- 선택자A의 형제 요소 중 선택자A 뒤에 위치하는 선택자B 요소를 모두 선택
    - 예) nav ~ section { ... }



# 셀렉터 조합하기-계층 접근용 셀렉터

- 2 개 이상의 셀렉터 조합
  - 조합에 적합한 HTML 태그에만 적용
- 조상/자손 셀렉터(Anccestor/descendent selector)
  - 자손 관계인 2 개 이상의 태그 나열

예) `ul li{ color : dodgerblue; }`

- `<ul>`의 자손 `<li>`에 적용되는 스타일 시트

- 부모/자식 셀렉터(Parent/child selector)
  - 부모 자식 관계인 두 셀렉터를 '`>`' 기호로 조합

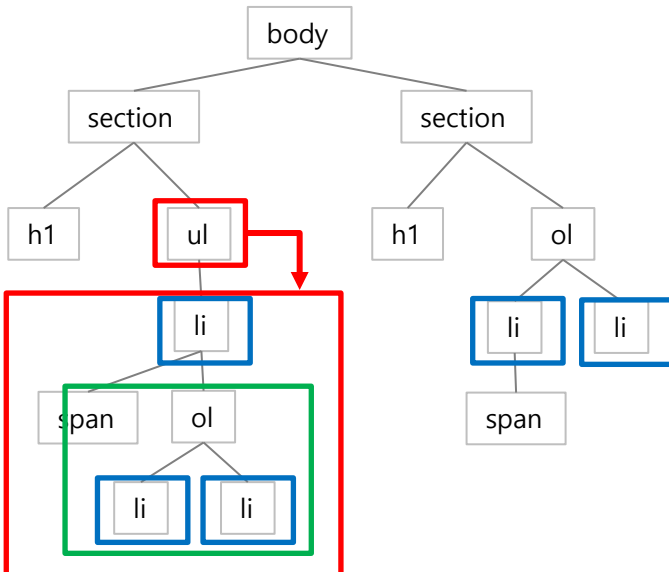
예) `ul > li{ color : dodgerblue; }`

- `<ul>`의 직계 자식인 `<li>`에 적용되는 스타일 시트



# 셀렉터 조합하기-계층 접근용 셀렉터

```
<section id="s1">
  <h1>&lt;header&gt; </h1>
  <ul>
    <li>페이지나 섹션의 <span>머리말</span> 표현
    <ol>
      <li>페이지 제목</li>
      <li>페이지를 소개</li>
    </ol>
  </li>
</ul>
</section>
<section id="s2">
  <h1>&lt;nav&gt; </h1>
  <ol>
    <li><span>하이퍼링크들을 모아 놓은 특별한 섹션</span>
    <li>페이지 내 목차를 만드는 용도</li>
  </ol>
</section>
```



```
li {
  color : blue ;
}
```



```
<header>
  • 페이지나 섹션의 머리말 표현
  1. 페이지 제목
  2. 페이지를 소개

<nav>
  1. 하이퍼링크들을 모아 놓은 특별한 섹션
  2. 페이지 내 목차를 만드는 용도
```

```
ul li {
  color : red ;
}
```



```
<header>
  • 페이지나 섹션의 머리말 표현
  1. 페이지 제목
  2. 페이지를 소개

<nav>
  1. 하이퍼링크들을 모아 놓은 특별한 섹션
  2. 페이지 내 목차를 만드는 용도
```

```
li > ol {
  color : green ;
}
```



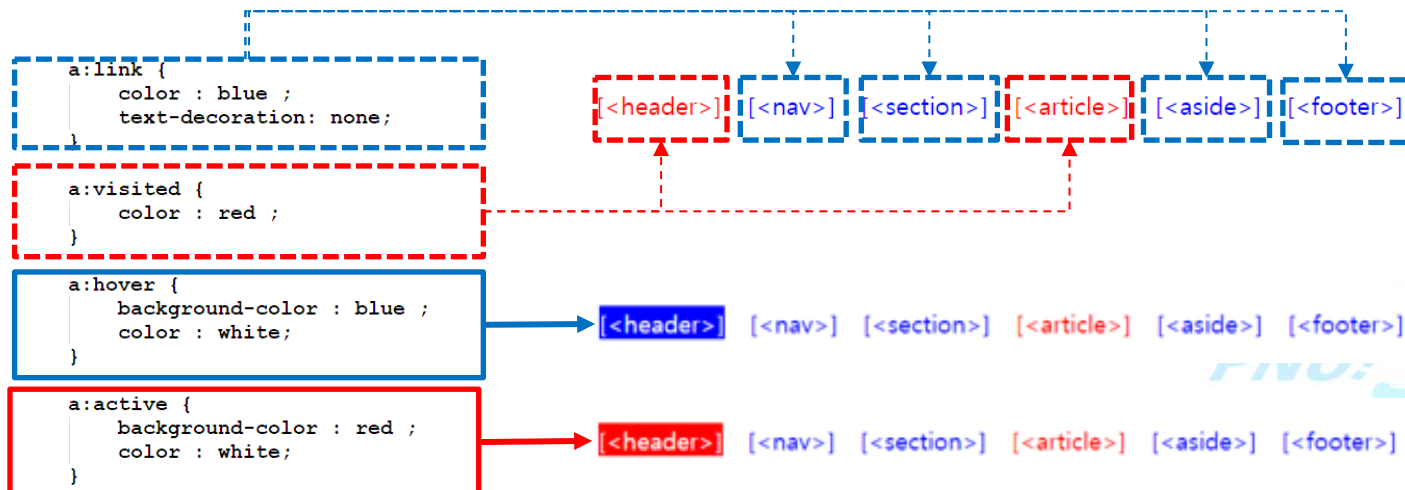
```
ul ol {
  color : green ;
}
```

```
<header>
  • 페이지나 섹션의 머리말 표현
  1. 페이지 제목
  2. 페이지를 소개

<nav>
  1. 하이퍼링크들을 모아 놓은 특별한 섹션
  2. 페이지 내 목차를 만드는 용도
```

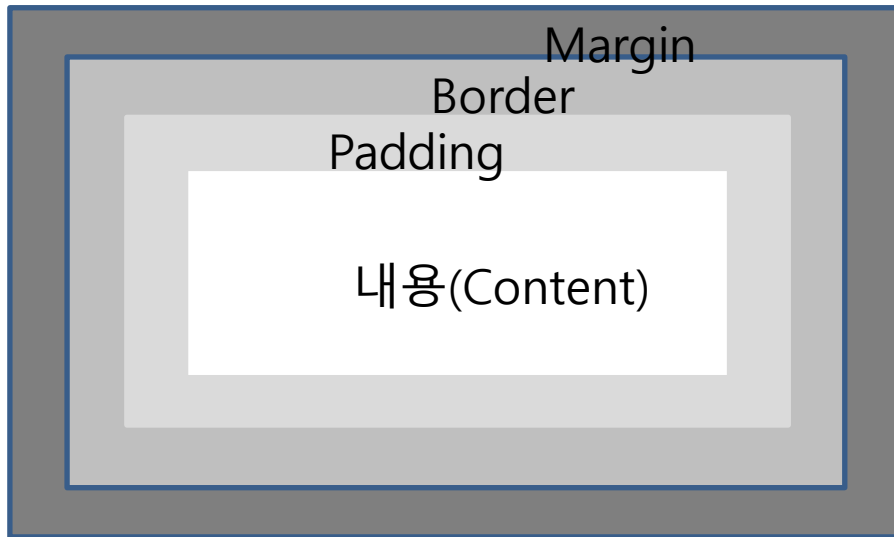
# 가상 클래스(Pseudo-class)

- 선택될 요소의 특별한 상태를 지정하기 위해 웹 문서에 존재하지 않는 임의의 선택자
- 하이퍼링크 요소 a와 관련된 가상클래스
  - **a:link**
    - 하이퍼링크 요소 중 아직 방문하지 않은 하이퍼링크에 적용
  - **a:visited**
    - 하이퍼링크 요소 중 한번 이상 방문한 하이퍼링크에 적용
  - **a:hover**
    - 하이퍼링크 요소에 마우스를 올려 놓았을 때 적용
  - **a:active**
    - 하이퍼링크 요소를 클릭했을 때 적용

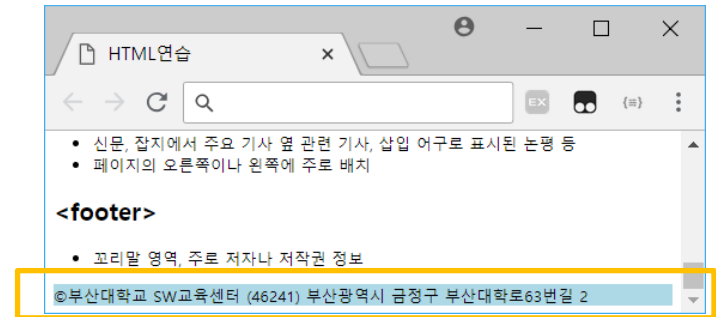


# 박스 모델(Box Model)

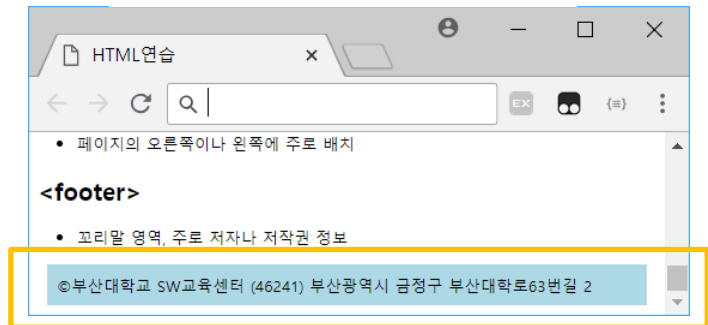
- HTML 요소의 실제 내용, 패딩, 테두리, 여백으로 구성



```
header, footer {  
    background-color : lightblue;  
}
```



```
header, footer {  
    background-color : lightblue;  
    padding : 10px;  
    border : 10px;  
    margin : 10px;  
}
```



# 여백 지정

- 안쪽 여백

- padding 속성으로 지정

- 1개의 값 : 4면 모두 동일한 값
    - 2개의 값: 상단하단과 왼쪽오른쪽 값
    - 3개의 값 : 상단, 왼쪽오른쪽, 하단 값
    - 4개의 값 : 상단, 오른쪽, 하단, 왼쪽 값

- 각 면에 대하여 지정

- padding-top, padding-right, padding-bottom, padding-left

- 바깥쪽 여백

- margin 속성으로 지정

- 1개의 값 : 4면 모두 동일한 값
    - 2개의 값: 상단하단과 왼쪽오른쪽 값
    - 3개의 값 : 상단, 왼쪽오른쪽, 하단 값
    - 4개의 값 : 상단, 오른쪽, 하단, 왼쪽 값

- 각 면에 대하여 지정

- margin-top, margin-right, margin-bottom, margin-left



# 배경 지정

- 요소의 배경색 지정

- background-color : 색상 값 ;

- 색상 값 지정 방법

- 색상 이름을 사용하여 지정

- [https://www.w3schools.com/colors/colors\\_names.asp](https://www.w3schools.com/colors/colors_names.asp)
    - background-color : lightblue;

- RGB 색상

- 빨강,초록,파란을 혼합하여 사용
    - background-color : rgb(173,216,230);

- RGBA 색상

- RGB 색상에 0~1까지 알파 값을 이용하여 투명도를 지정
    - background-color : rgb(173,216,230,0.5);

- HEX (16진수) 값 사용

- background-color : #add8e6;





# 테두리 지정

- **border**
  - 요소의 테두리 두께 지정
- **border-style**
  - dotted - 점선 테두리
  - dashed - 점선 테두리
  - solid - 단색 테두리
  - double - 이중 테두리
  - none - 테두리 없음
- **border-width**
  - 테두리 너비
- **border-color**
  - 테두리 색상
- **테두리 스타일, 너비, 색상**
  - 2개의 값 지정
    - 위쪽 아래 테두리
    - 왼쪽 오른쪽 테두리
  - 4개의 값 지정
    - 위쪽 테두리
    - 오른쪽 테두리
    - 아래쪽 테두리
    - 왼쪽 테두리

<header>

- 페이지나 섹션의 머리말 표현
- 페이지 제목, 페이지를 소개하는 간단한 설명

<header>

- 페이지나 섹션의 머리말 표현
- 페이지 제목, 페이지를 소개하는 간단한 설명


<header>

- 페이지나 섹션의 머리말 표현
- 페이지 제목, 페이지를 소개하는 간단한 설명

```
#s1 {  
  color : blue ;  
  margin : 10px;  
  border : solid 10px green ;  
}
```

```
#s1 {  
  color : blue ;  
  margin : 10px;  
  border-style : solid ;  
  border-width : 10px ;  
  border-color : green ;  
}
```

```
#s1 {  
  color : blue ;  
  margin : 10px;  
  border-style : solid dotted;  
  border-width : 10px 2px ;  
  border-color : green red;  
}
```

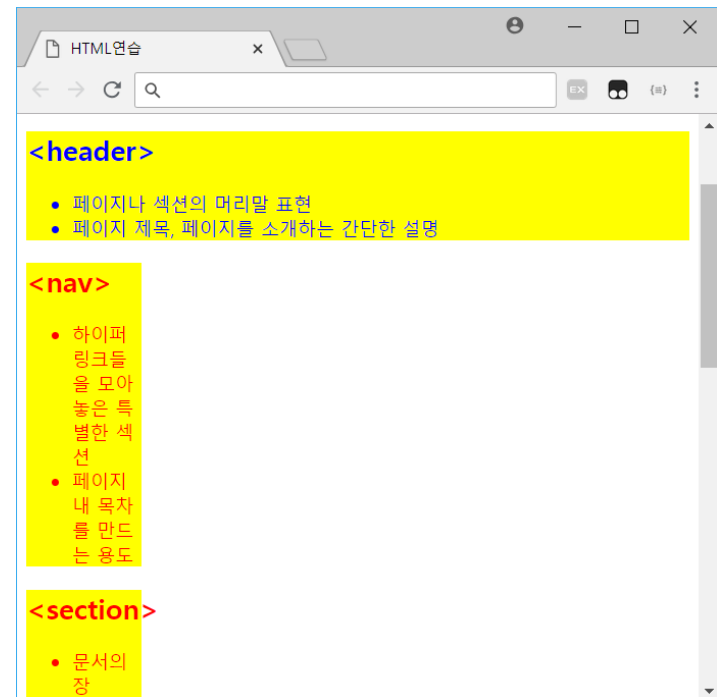
Come with 

```
#s1 {  
  color : blue ;  
  margin : 10px;  
  border-style : solid dotted none double;  
  border-width : 10px 6px 3px 1px;  
  border-color : green red tomato blue;  
}
```

# 너비와 높이 지정

- 너비 : width
- 높이 : height
- %, px, cm로 지정 가능

```
#s1 {  
    color : blue ;  
    background-color : yellow;  
    width : 100% ;  
}  
  
.c1 {  
    color : red ;  
    background-color : yellow;  
    width : 100px ;  
}
```



# 텍스트

- 텍스트 색상
  - color
  - 색상이름, HEX 값, RGB 값
- 텍스트 정렬
  - text-align
  - 가운데 정렬 : text-align: center;
  - 왼쪽 정렬 : text-align: left;
  - 오른쪽 정렬 : text-align: right;
  - 양쪽 정렬 : text-align: justify;
- 글꼴 크기
  - font-size
  - px과 em으로 표시
    - em은 가변단위로 1em은 현재 지정된 폰트의 크기를 말함
    - 1em = 16px



# 목록 스타일

- 목록의 아이템 모양

- ul

- list-style-type: circle;
    - list-style-type: square;

- ol

- list-style-type: upper-roman;
    - list-style-type: lower-alpha;

```
ul {  
    list-style-type: circle;  
}  
  
ol {  
    list-style-type: upper-roman;  
}
```

## <header>

- 페이지나 섹션의 머리말 표현
- 페이지 제목, 페이지를 소개하는 간단한 설명

## <nav>

- I. 하이퍼링크들을 모아 놓은 특별한 섹션
- II. 페이지 내 목차를 만드는 용도

# display 속성

- HTML 요소를 어떻게 보여줄 지 결정

- HTML 요소는 기본 값을 가짐

- 예) div, p, ul : 블록(block)요소
- 예) span, a : 인라인(inline)요소

- display 속성을 이용하여 기본값 변경

- display:none;
  - 영역을 찾아하지 않고 보이지 않음
- display:block;
  - 블록 영역으로 기본적으로 브라우저 전체 너비가 적용되고 줄바꿈 적용
  - 가로(width)와 세로(height)를 지정할 수 있음
- display:inline;
  - 인라인 영역으로 요소의 내용만큼만 너비가 적용되고 줄바꿈이 적용되지 않음
  - 가로(width)와 세로(height)를 지정할 수 없음
- display:inline-block;
  - 블록과 인라인 영역의 중간 형태로 크기를 변경할 수 있고 줄바꿈이 적용되지 않음
  - 가로(width)와 세로(height)를 지정할 수 있음

기본:inline `[<header>]` `[<nav>]` `[<section>]` `[<article>]` `[<aside>]` `[<footer>]`

Inline-block적

```
a {  
  width : 100px;  
  height : 25px;  
  padding : 5px;  
  text-align : center ;  
  display : inline-block;  
}
```

`[<header>]` `[<nav>]` `[<section>]` `[<article>]` `[<aside>]` `[<footer>]`

Come with  
PNU!



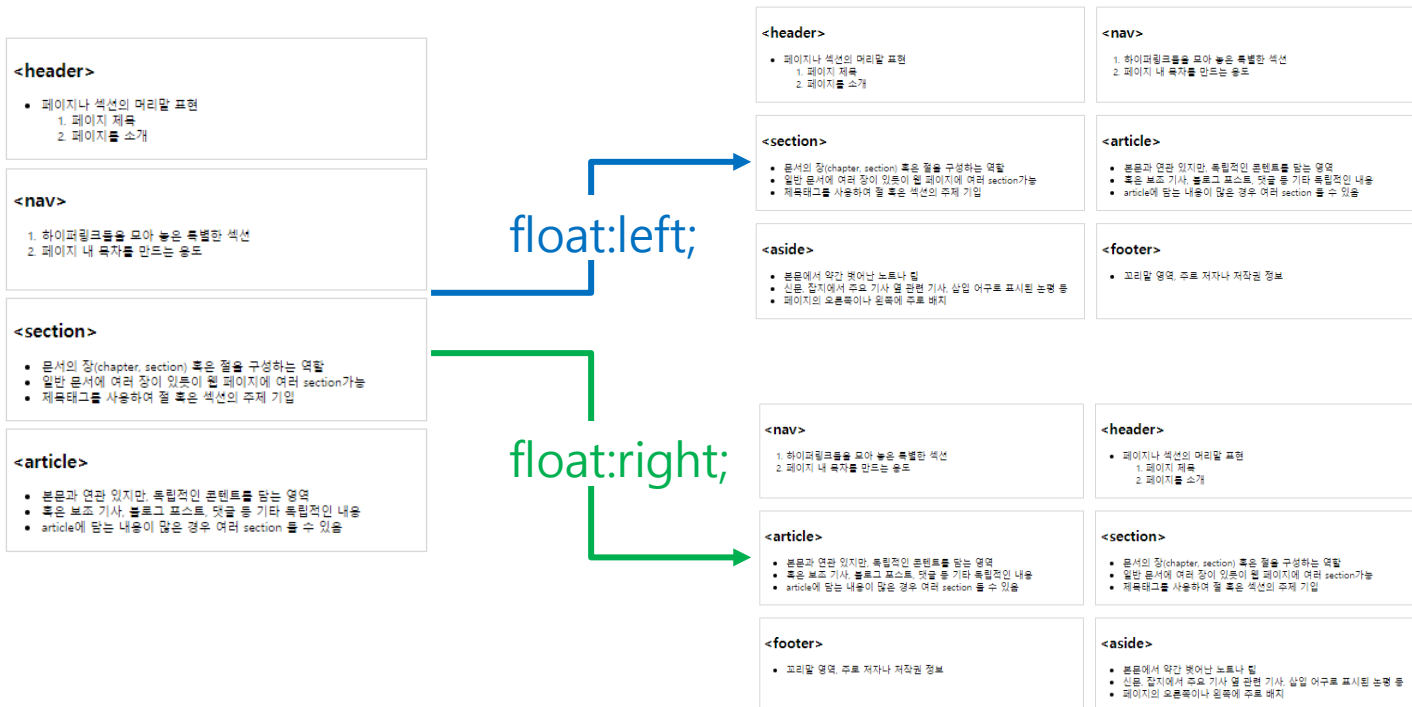
# float 속성/ clear 속성

## • float 속성

- 특정 요소를 정렬하여 흐르듯이 배치
- float:left; 왼쪽으로 배치
- float:right; 오른쪽으로 배치

## • clear 속성

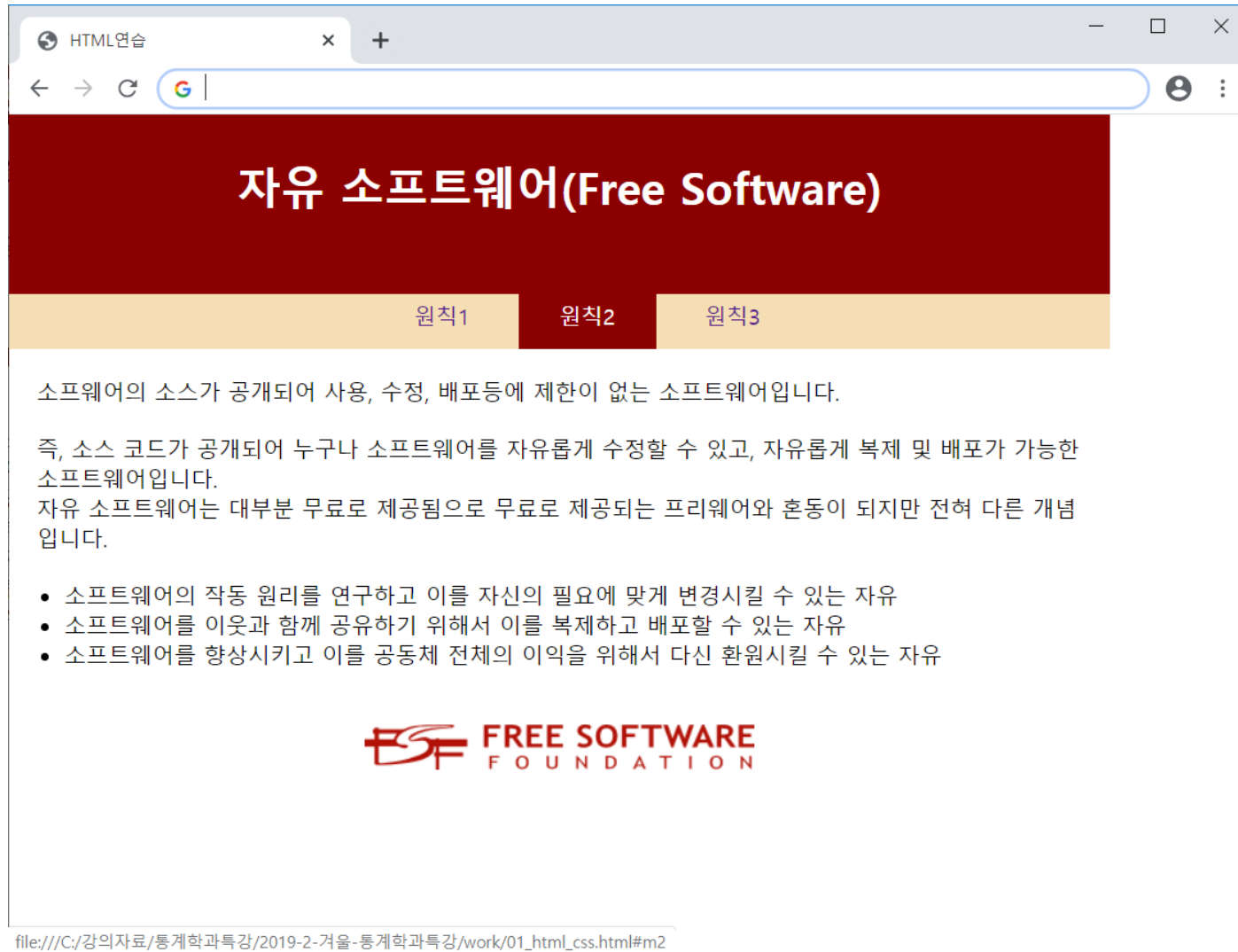
- float 속성을 해제하고 줄바꿈



# box-sizing 속성

- HTML 블록 요소들의 실제 너비와 높이는 너비와 높이 값에 안쪽 여백, 바깥쪽 여백, 테두리 값을 모두 더해서 표시됨
- **box-sizing: border-box;**
  - 실제 너비와 높이를 지정한 너비와 높이로 고정
  - 즉 안쪽 여백, 바깥쪽 여백, 테두리 값을 지정하더라도 지정한 너비와 높이만큼 표시되도록 함

# 실습



HTML연습

## 자유 소프트웨어(Free Software)


원칙1    원칙2    원칙3

소프트웨어의 소스가 공개되어 사용, 수정, 배포 등에 제한이 없는 소프트웨어입니다.

즉, 소스 코드가 공개되어 누구나 소프트웨어를 자유롭게 수정할 수 있고, 자유롭게 복제 및 배포가 가능한 소프트웨어입니다.

자유 소프트웨어는 대부분 무료로 제공됨으로 무료로 제공되는 프리웨어와 혼동이 되지만 전혀 다른 개념입니다.

- 소프트웨어의 작동 원리를 연구하고 이를 자신의 필요에 맞게 변경시킬 수 있는 자유
- 소프트웨어를 이웃과 함께 공유하기 위해서 이를 복제하고 배포할 수 있는 자유
- 소프트웨어를 향상시키고 이를 공동체 전체의 이익을 위해서 다신 환원시킬 수 있는 자유

 **FREE SOFTWARE**  
FOUNDATION

file:///C:/강의자료/통계학과특강/2019-2-겨울-통계학과특강/work/01\_html\_css.html#m2



# 크롬 개발자 도구

HTML연습 x +

← → ↻ Google에서 검색하거나 URL을 입력하세요.

## 자유 소프트웨어(Free Software)

h1 800 × 43.2

원칙1 원칙2 원칙3

소프트웨어의 소스가 공개되어 사용, 수정, 배포 등에 제한이 없는 소프트웨어입니다.

즉, 소스 코드가 공개되어 누구나 소프트웨어를 자유롭게 수정할 수 있고, 자유롭게 복제 및 배포가 가능한 소프트웨어입니다.

자유 소프트웨어는 대부분 무료로 제공됨으로 무료로 제공되는 프리웨어와 혼동이 되지만 전혀 다른 개념입니다.

- 소프트웨어의 작동 원리를 연구하고 이를 자신의 필요에 맞게 변경시킬 수 있는 자유
- 소프트웨어를 이웃과 함께 공유하기 위해서 이를 복제하고 배포할 수 있는 자유
- 소프트웨어를 향상시키고 이를 공동체 전체의 이익을 위해서 다시 환원시킬 수 있는 자유

**FREE SOFTWARE**  
FOUNDATION

Elements Console Sources Network

```
<!doctype html>
<html>
  <head>...</head>
  <body>
    <header>
      <h1>자유 소프트웨어(Free Software)</h1>
    </header>
    <nav>...</nav>
    <section> == $0
      <p>
        소프트웨어의 소스가 공개되어 사용, 수정, 배포 등에 제한이 없는 소프트
        웨어입니다.
      </p>
    </section>
    <section>...</section>
    <footer>...</footer>
  </body>
</html>
```

html body section

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

```
element.style {
}
header, nav, section, footer {
  width: 800px;
}
* {
  margin: 0px;
}
section {
  display: block;
}
Inherited from html
html {
  color: -internal-root-color;
}
```

margin -  
border -  
padding -  
800 × 128  
color -  
rgb(0, 0, 0)  
display block

Console What's New x

Highlights from the Chrome 79 update