

# Web, HTML, CSS

yujingaya  
2017 Fall

# 다루는 내용

## Web

Server, Client

HTTP Protocol

## HTML

Trend: Semantic Web

Hypermedia

## CSS

Cascading Rule

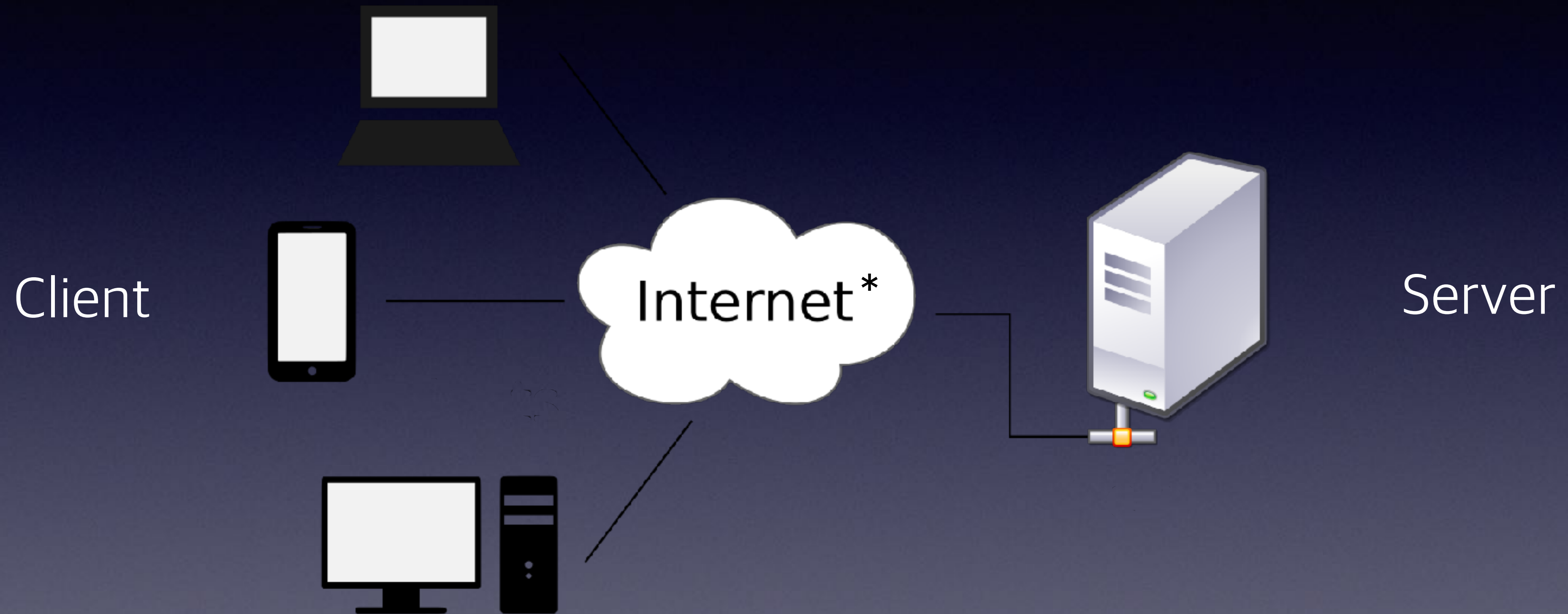
Web

# Tim Berners-Lee

Inventor of the Web



# Server Client Model



\* (막간 홍보) Internet 에서 일어나는 일들이 궁금하다면 여름에 열리는 Wheel Seminar 를 들으세요!

# 첫 웹 페이지

Hypertext

웹 상의 어떤 문서든 링크로  
접근할 수 있게 하자!

## World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

### [What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

### [Help](#)

on the browser you are using

### [Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,X11 [Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#) )

### [Technical](#)

Details of protocols, formats, program internals etc

### [Bibliography](#)

Paper documentation on W3 and references.

### [People](#)

A list of some people involved in the project.

### [History](#)

A summary of the history of the project.

### [How can I help ?](#)

If you would like to support the web..

### [Getting code](#)

Getting the code by [anonymous FTP](#) , etc.

# 어떻게 서버를 찾아가는가: URL

<https://yujingaya.com:8080/wiki/index.html?board=love&page=3#title>

**Protocol:** http, https, ftp, mailto, ...

**Hostname:** google.com, netflix.com, ...

**Path:** /wiki, /w/URL, ...

**Port:** :8080, :8000, :60138

**Query:** ?board=love&page=3

**Fragment:** #title

# 통신 방법: HTTP

## Client Request

```
GET /index.html HTTP/1.1
Host: www.example.com
Accept-Language: en
```

Method: GET, POST, PUT, DELETE, ...

Header: Date, Content-Type, ...

Status Code: 200, 404, 500, ...

## Server Response

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
```

```
<html>
<head>
  <title>An Example Page</title>
</head>
<body>
  Hello World, this is a very simple HTML document.
</body>
</html>
```

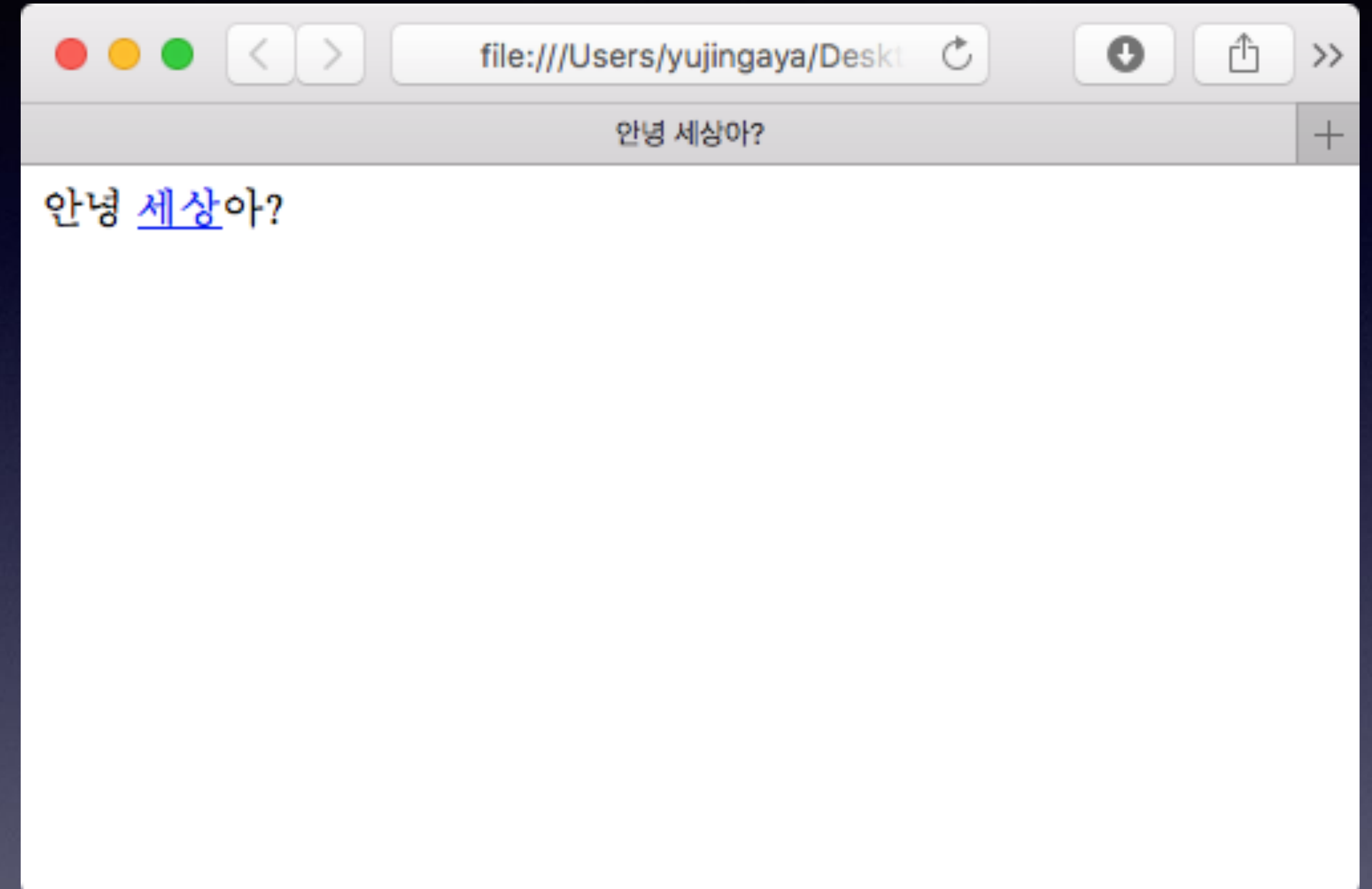


# HTML

HyperText Markup Language

# 기본적인 구조

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>안녕 세상아? </title>
  </head>
  <body>
    안녕 <a href="google.com" target="_blank">세상</a>아?
  </body>
</html>
```



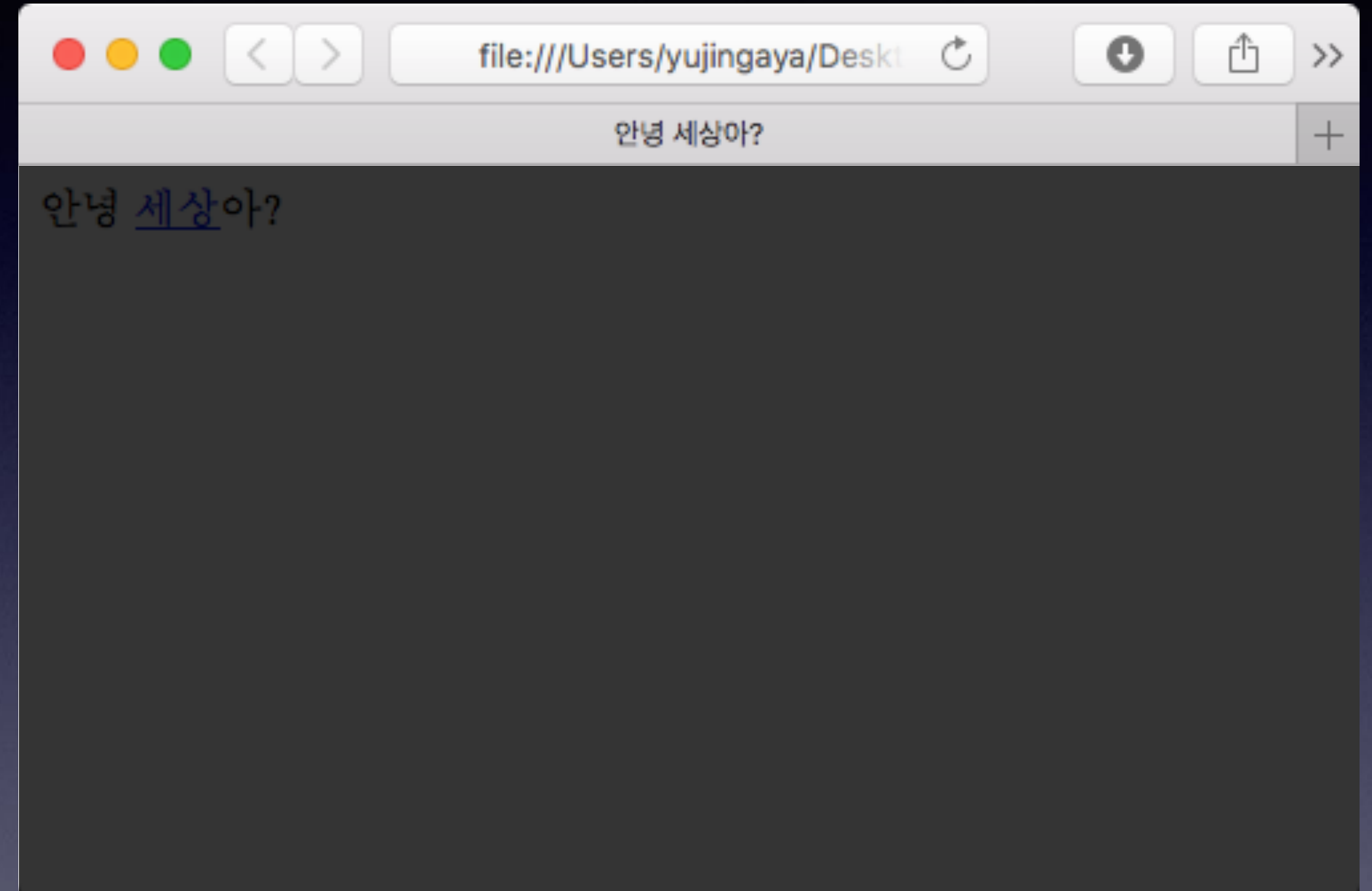
**Document Type:** HTML 버전 명시. (HTML5 의 경우 html)

**Tag:** 미리 정의 된 태그 (뒤에서 다룹니다)

**Attribute:** 태그의 기능을 추가 (이 링크는 google.com 으로 가는 링크야)

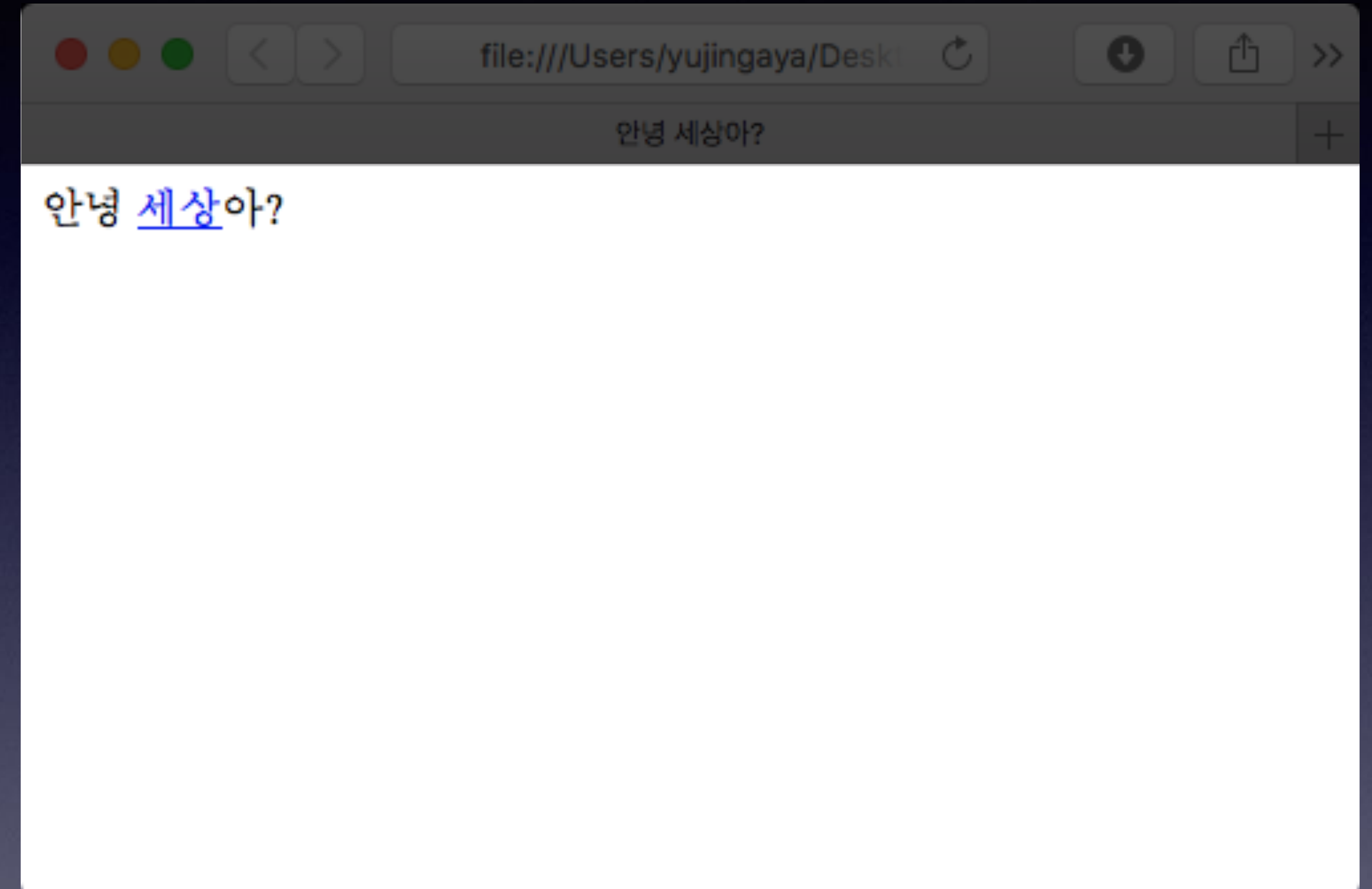
# Head: 부가 정보

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>안녕 세상아? </title>
  </head>
  <body>
    안녕 <a href="google.com" target="_blank">세상</a>아?
  </body>
</html>
```



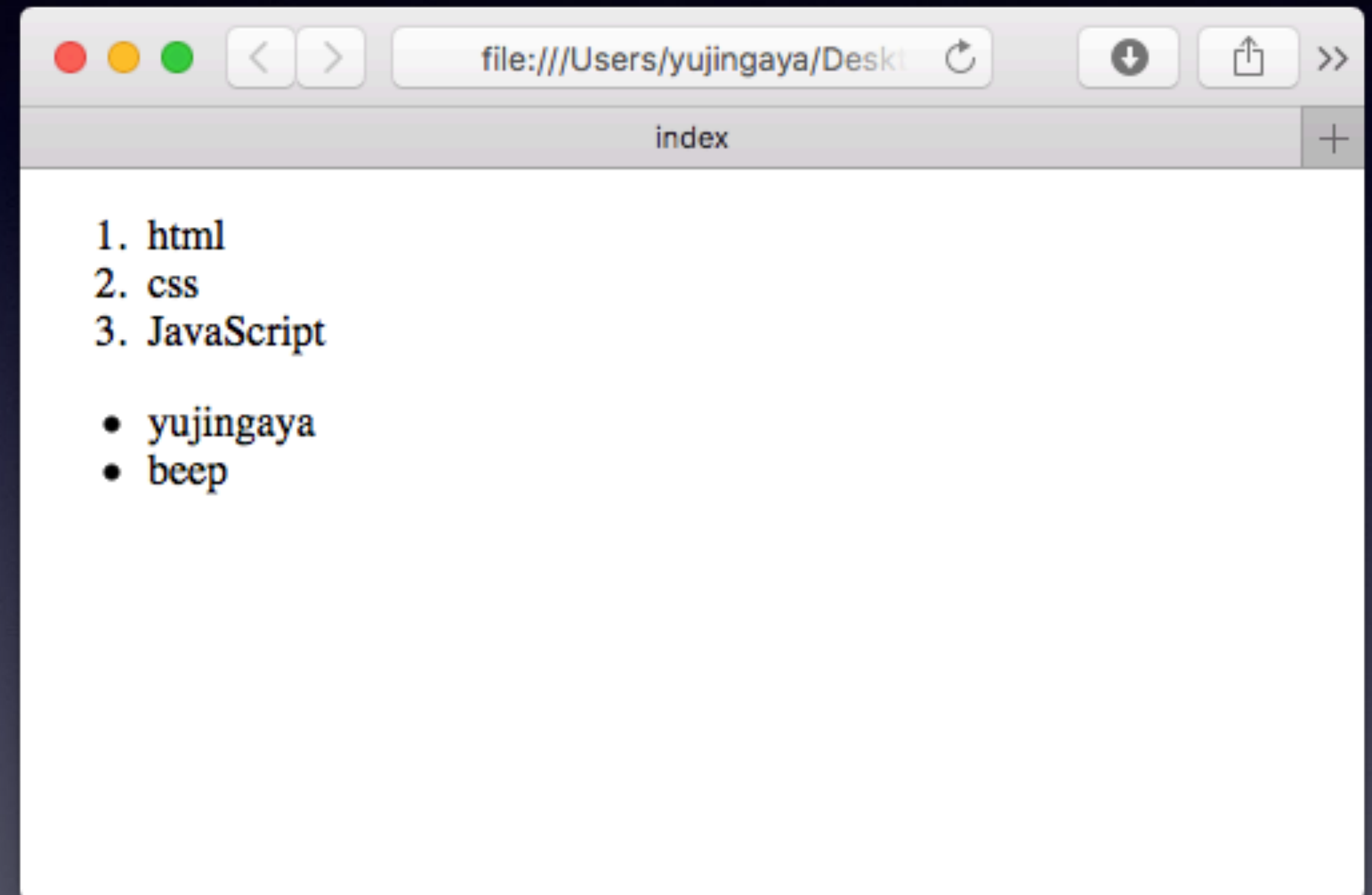
# Body: 보여지는 부분

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title> 안녕 세상아? </title>
  </head>
  <body>
    안녕 <a href="google.com" target="_blank">세상</a>아?
  </body>
</html>
```



# List: ol, ul

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
  </head>
  <body>
    <ol>
      <li>html</li>
      <li>css</li>
      <li>JavaScript</li>
    </ol>
    <ul>
      <li>yujingaya</li>
      <li>beep</li>
    </ul>
  </body>
</html>
```



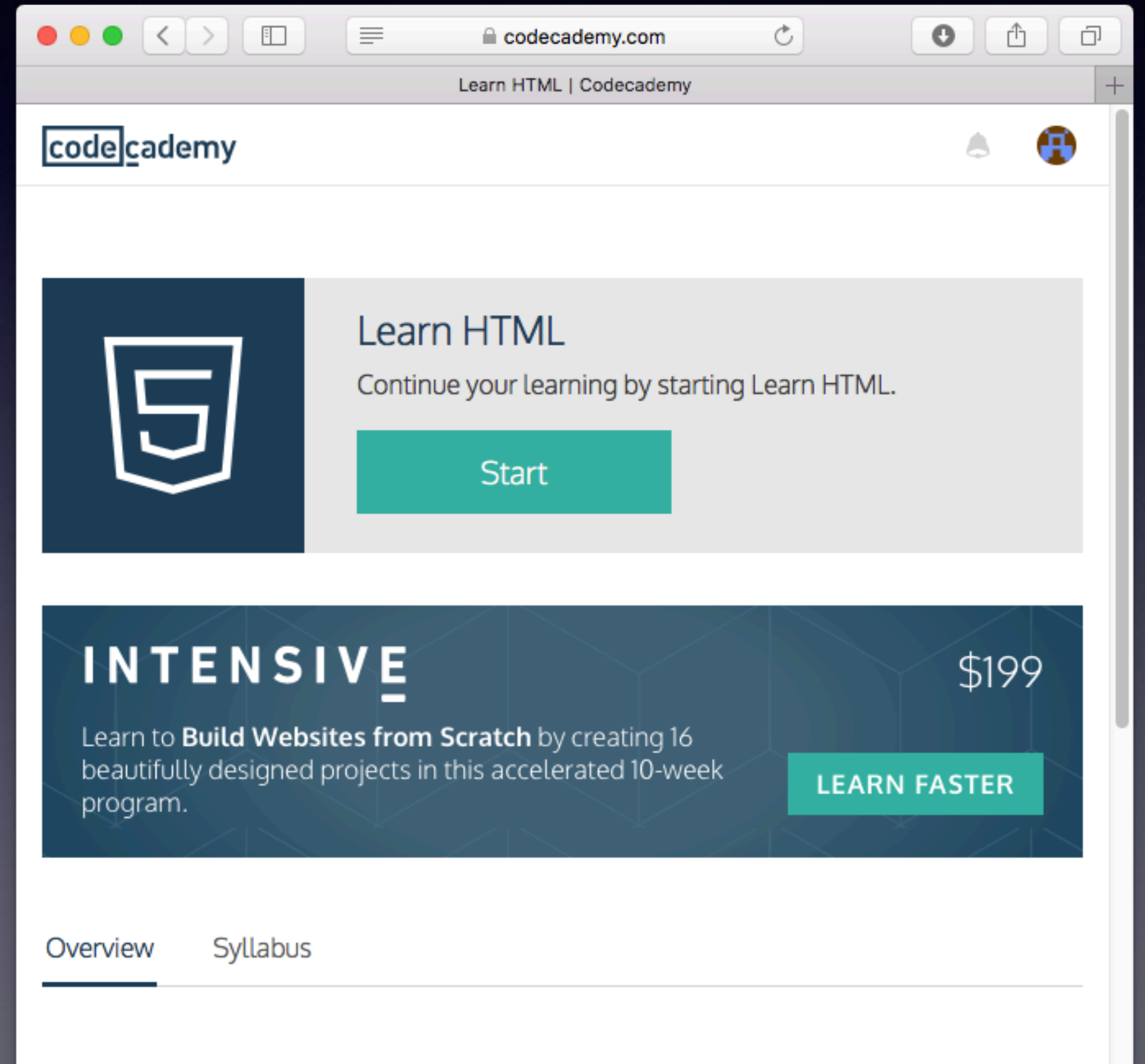
# Heading, Paragraph

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>멋진 신입생 세미나</h1>
    <p>이건 본문입니다</p>
  </body>
</html>
```



# 다른 태그들을 알아보려면?

- <http://htmlreference.io>: 사전
- <http://caniuse.com>: xx 브라우저에서 지원 되나요
- <https://www.codecademy.com/learn/learn-html> (숙제!!! 약 2시간 걸림)

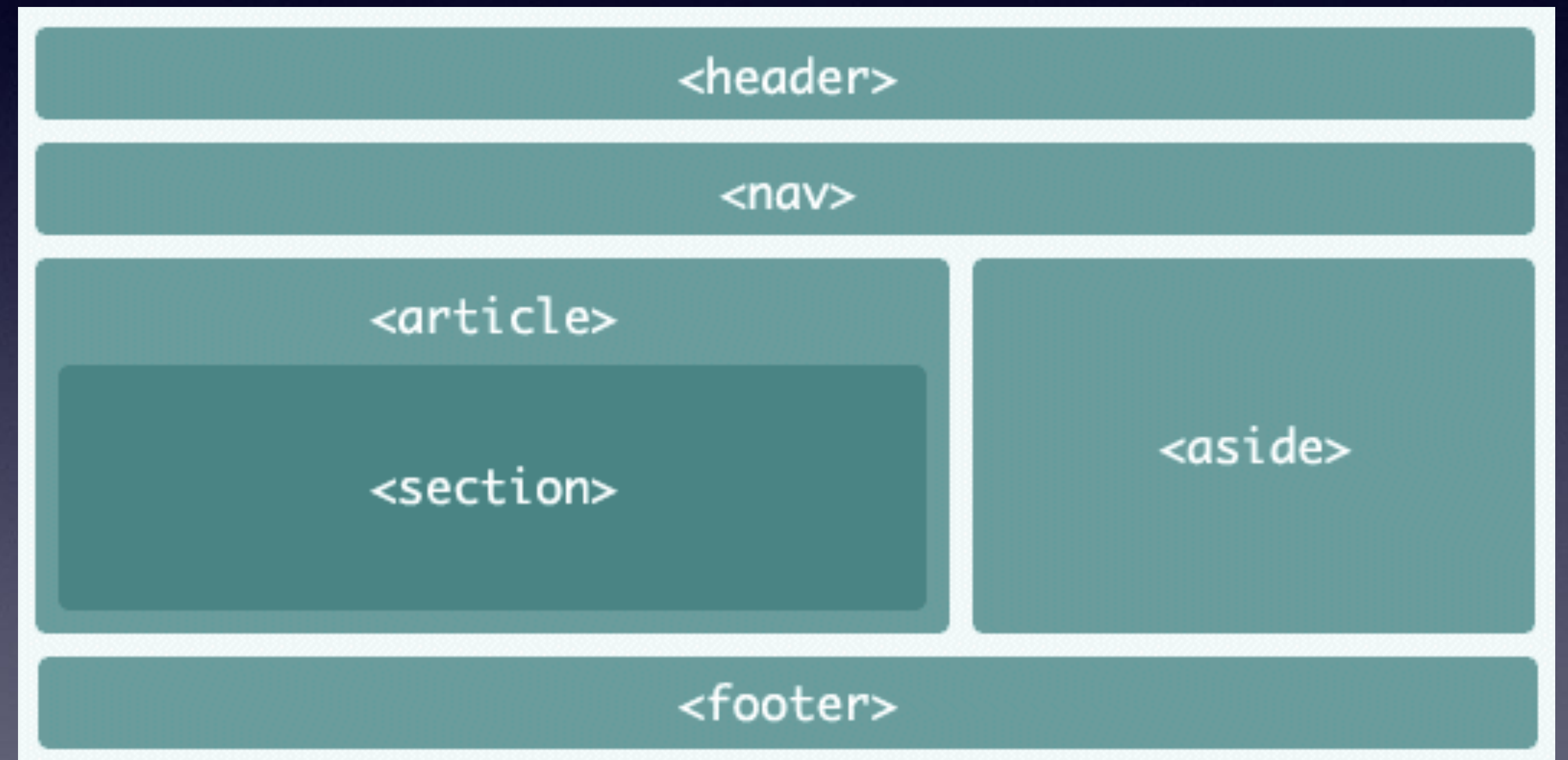


# Trend: Semantic Web

대-충 모양만 똑같으면 되지!



의미에 맞는 태그를 써야  
(접근성) 시각 장애인도 읽을 수 있고  
(SEO) 검색 엔진에도 잘 걸린다구!





# Trend: Hypermedia

글자, 그림만 잘 전달!



NETFLIX



링크는 모든 걸 향할 수 있어!

동영상, 음악

상거래, 지도, SNS

WebGL, WebAssembly



twitch

# CSS

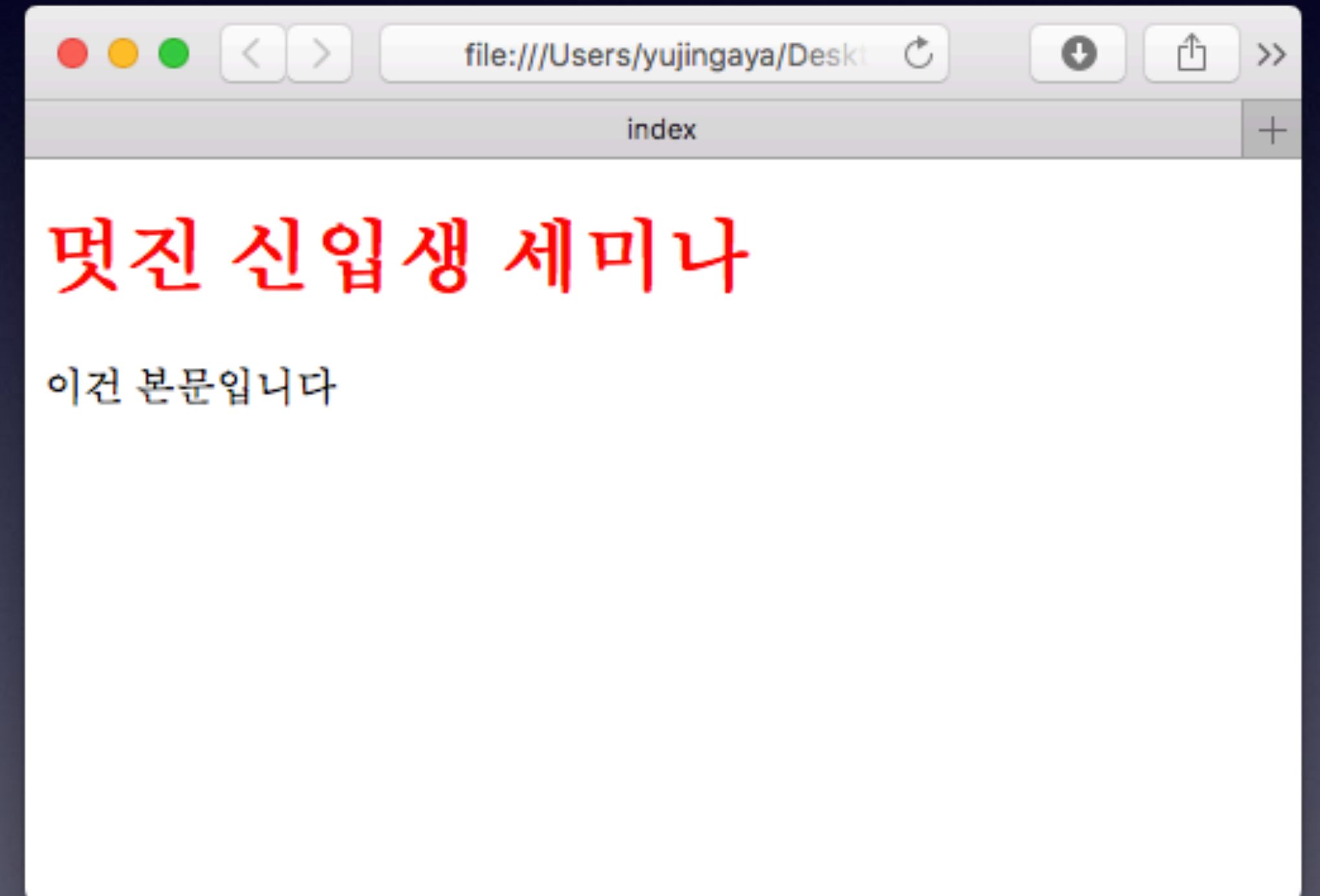
Cascading Style Sheets

(폭포처럼) 떨어져 내리는

# 못생긴 기본 스타일.. 직접 꾸미려면?

## Inline Style

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1 style="color: red;">멋진 신입생 세미나</h1>
    <p>이건 본문입니다</p>
  </body>
</html>
```



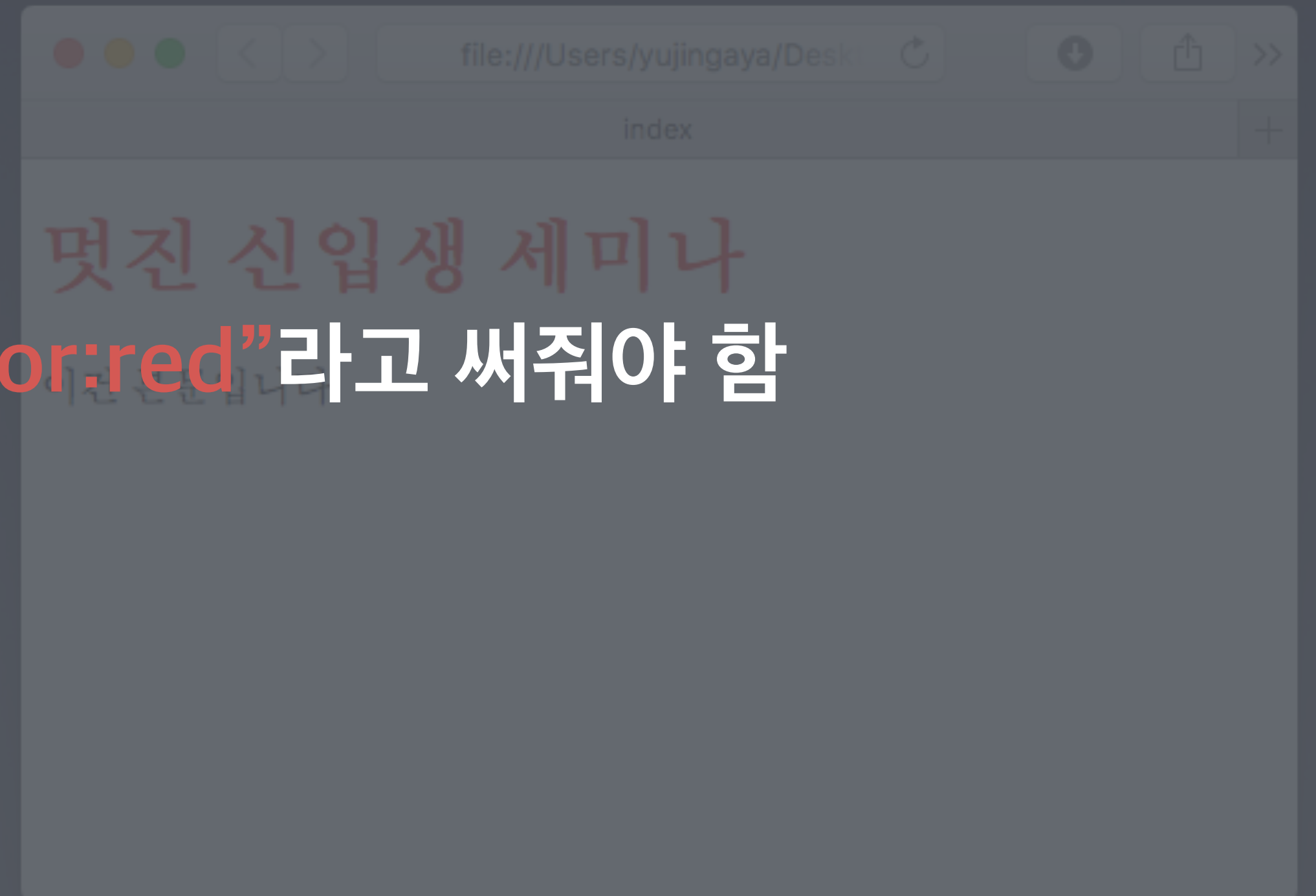
# 못생긴 기본 스타일.. 직접 꾸미려면?

## Inline Style

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1 style="color: red;">멋진 신입생 세미나</h1>
    <p>이건 본문입니다</p>
  </body>
</html>
```

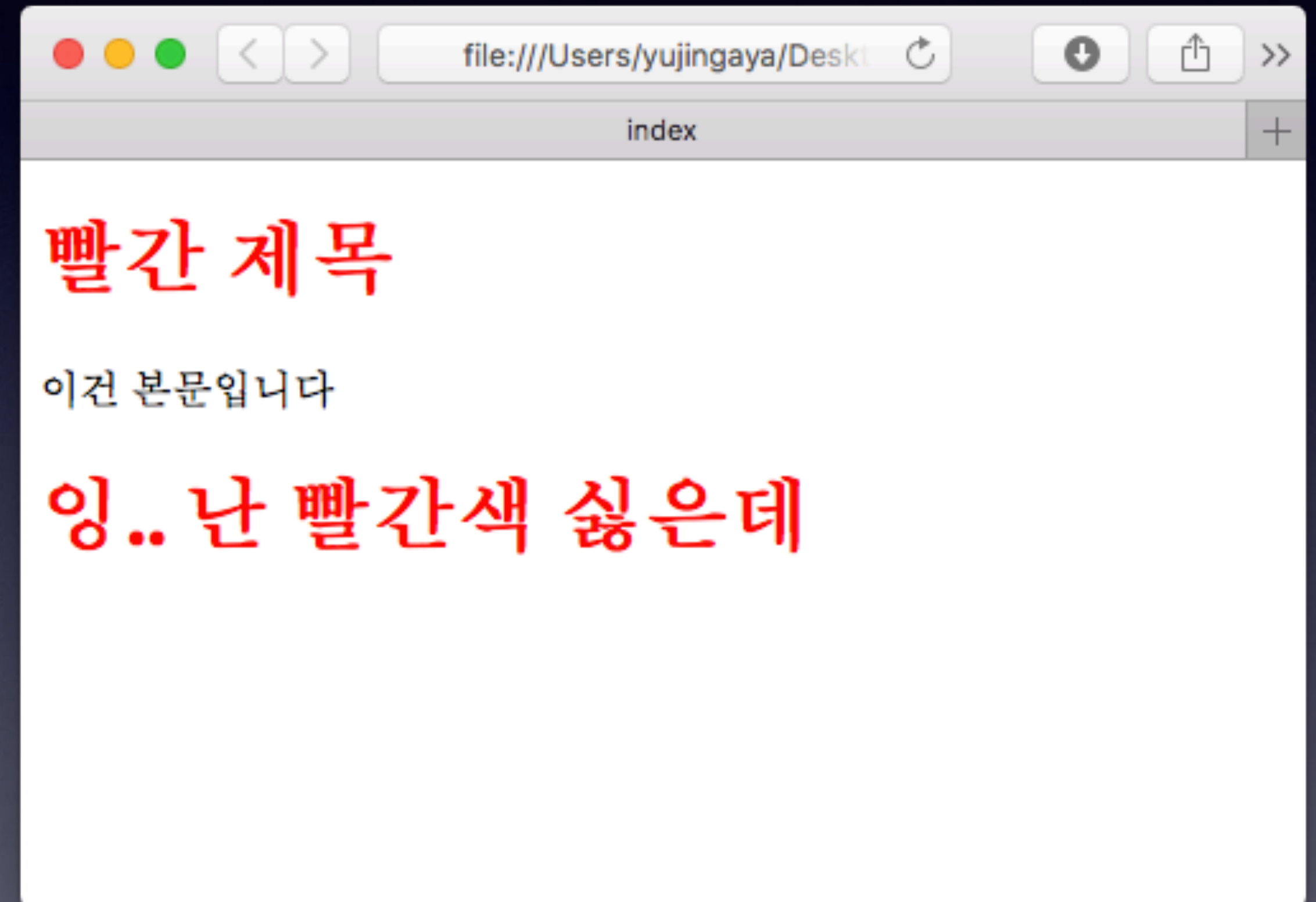
## 단점

스타일을 줄 때마다 계속 **style="color:red"**라고 써줘야 함



# <style> tag

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <style>
      h1 {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1>빨간 제목</h1>
    <p>이건 본문입니다</p>
    <h1>잉.. 난 빨간색 싫은데</h1>
  </body>
</html>
```

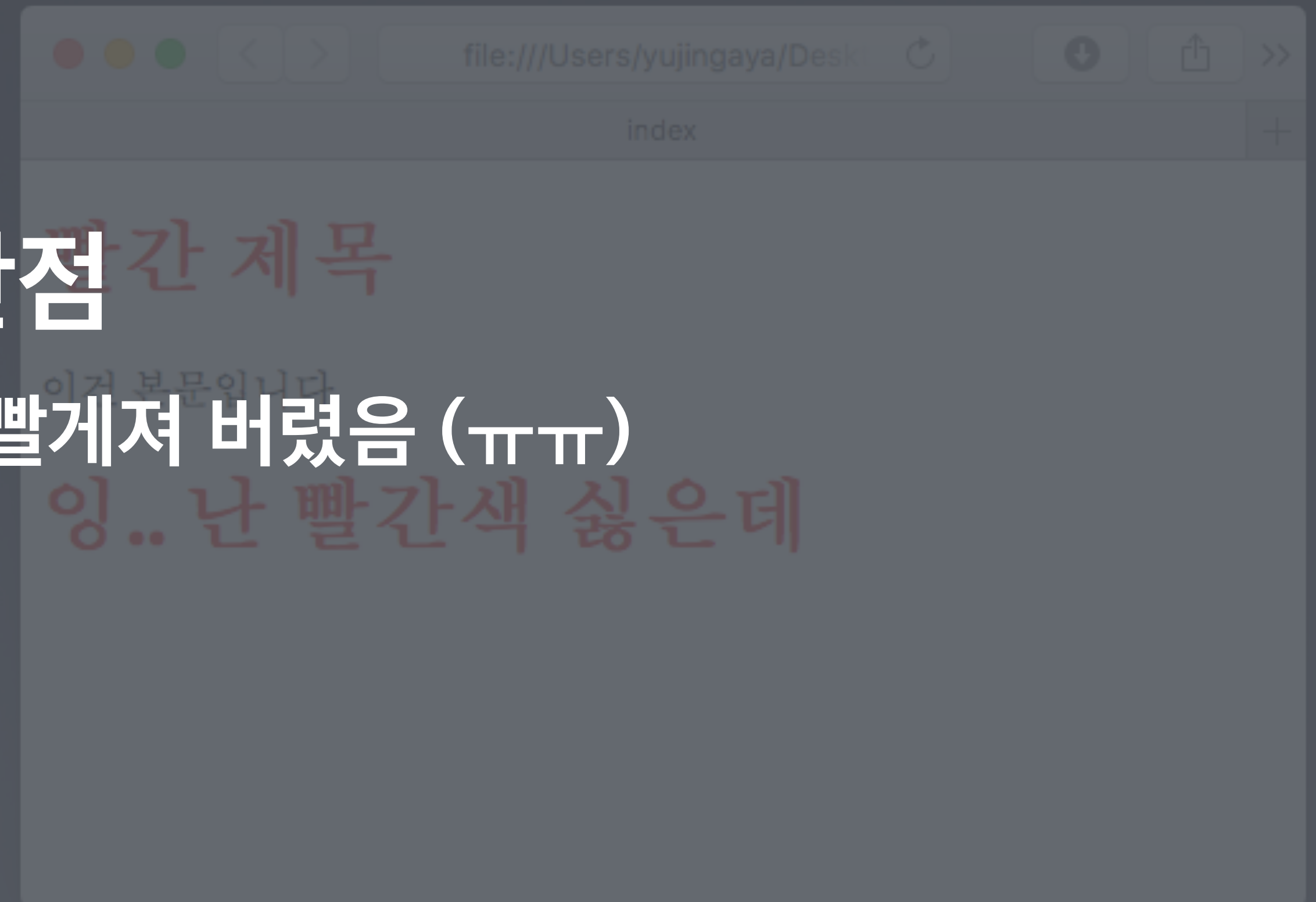


# <style> tag

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <style>
      h1 {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1>빨간 제목</h1>
    <p>이건 본문입니다</p>
    <h1>잉.. 난 빨간색 싫은데</h1>
  </body>
</html>
```

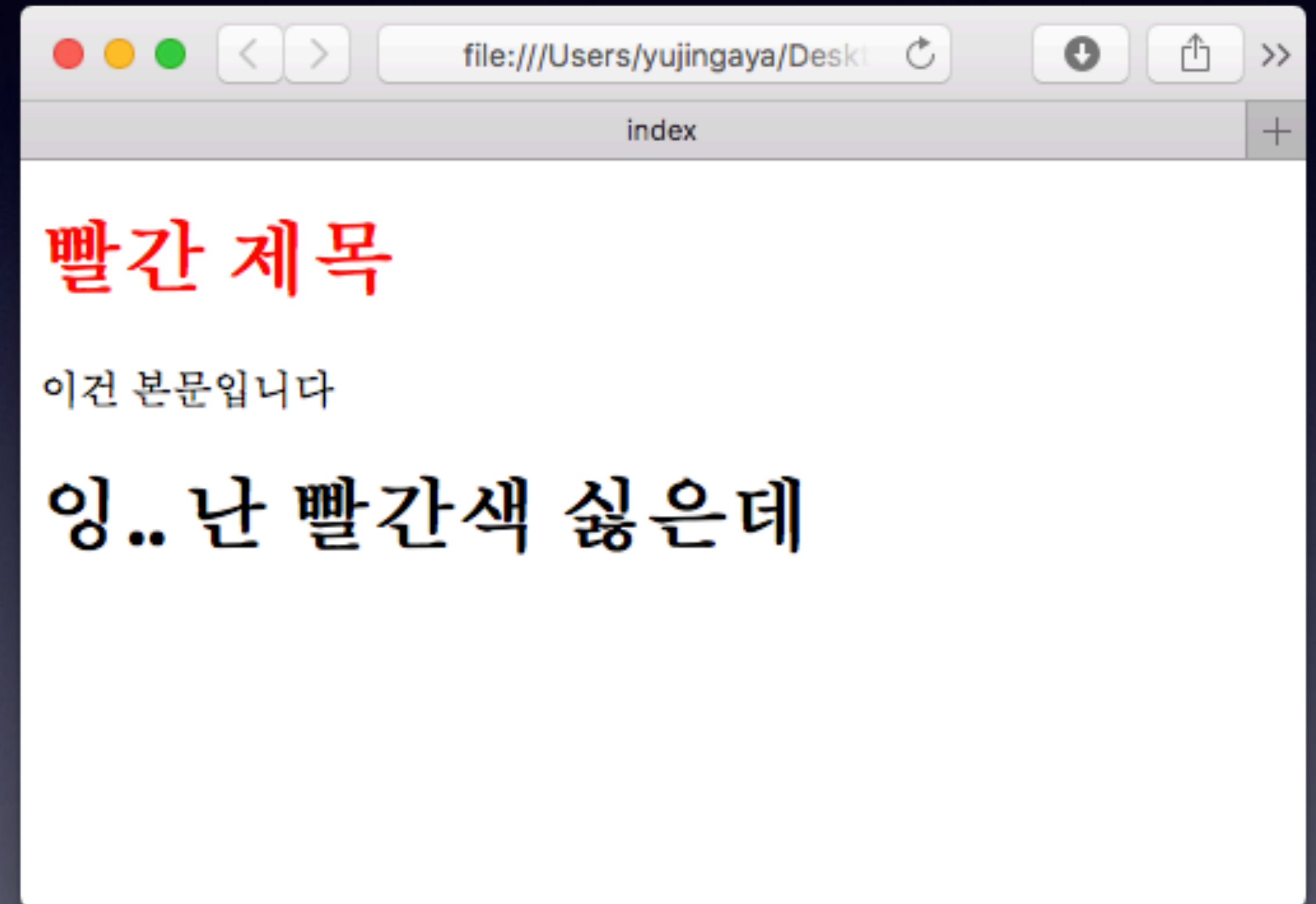
## 단점

모든 h1 태그가 다 빨게져 버렸음 (ㅠㅠ)



# <style> tag with class, id

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <style>
      h1.red-title {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1 class="red-title">빨간 제목</h1>
    <p>이건 본문입니다</p>
    <h1>잉.. 난 빨간색 싫은데</h1>
  </body>
</html>
```



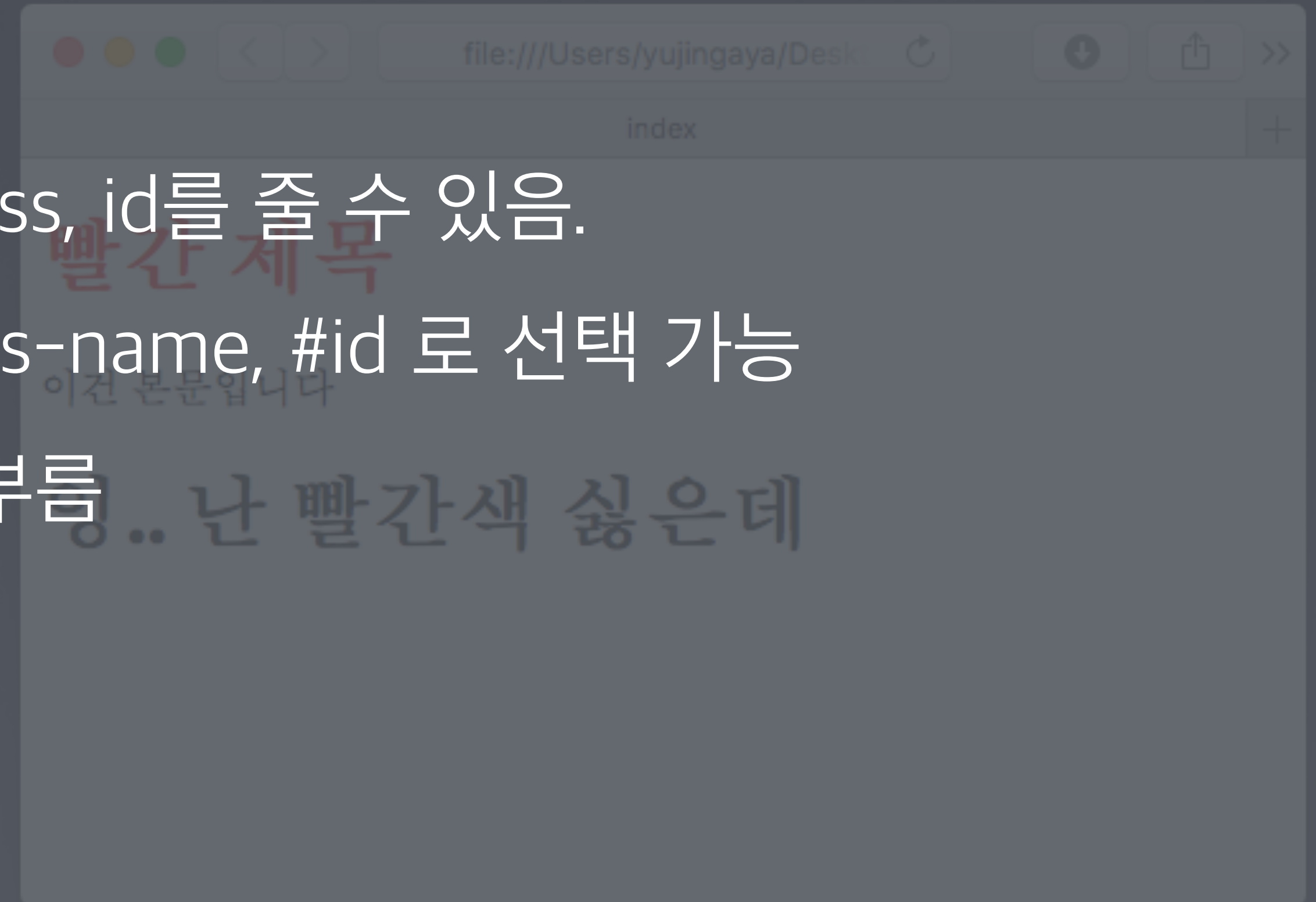
# <style> tag with class, id

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <style>
    h1 .red-title {
      color: red;
    }
  </style>
</head>
<body>
  <h1 class="red-title">빨간 제목</h1>
  <p>이건 본문입니다</p>
  <h1>잉.. 난 빨간색 싫은데</h1>
</body>
</html>
```

1. 태그에 attribute 로 class, id를 줄 수 있음.

2. <style> 안에서는 .class-name, #id 로 선택 가능

3. 이런걸 selector 라고 부름





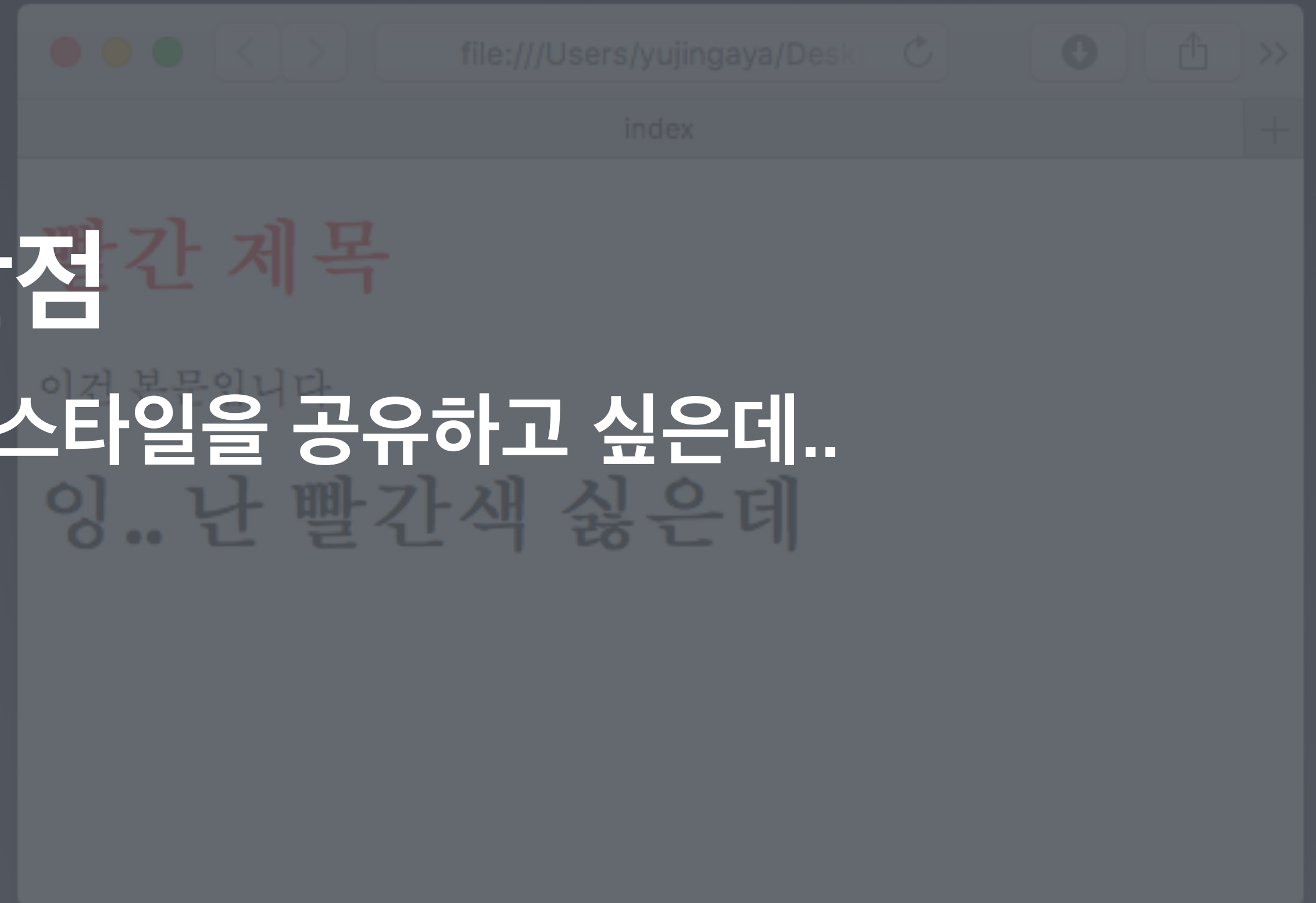
# <style> tag with class, id

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <style>
      h1 .red-title {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1 class="red-title">빨간 제목</h1>
    <p>이건 본문입니다</p>
    <h1>잉.. 난 빨간색 싫은데</h1>
  </body>
</html>
```

## 단점

여러 html 파일에서 같은 스타일을 공유하고 싶은데..

잉.. 난 빨간색 싫은데



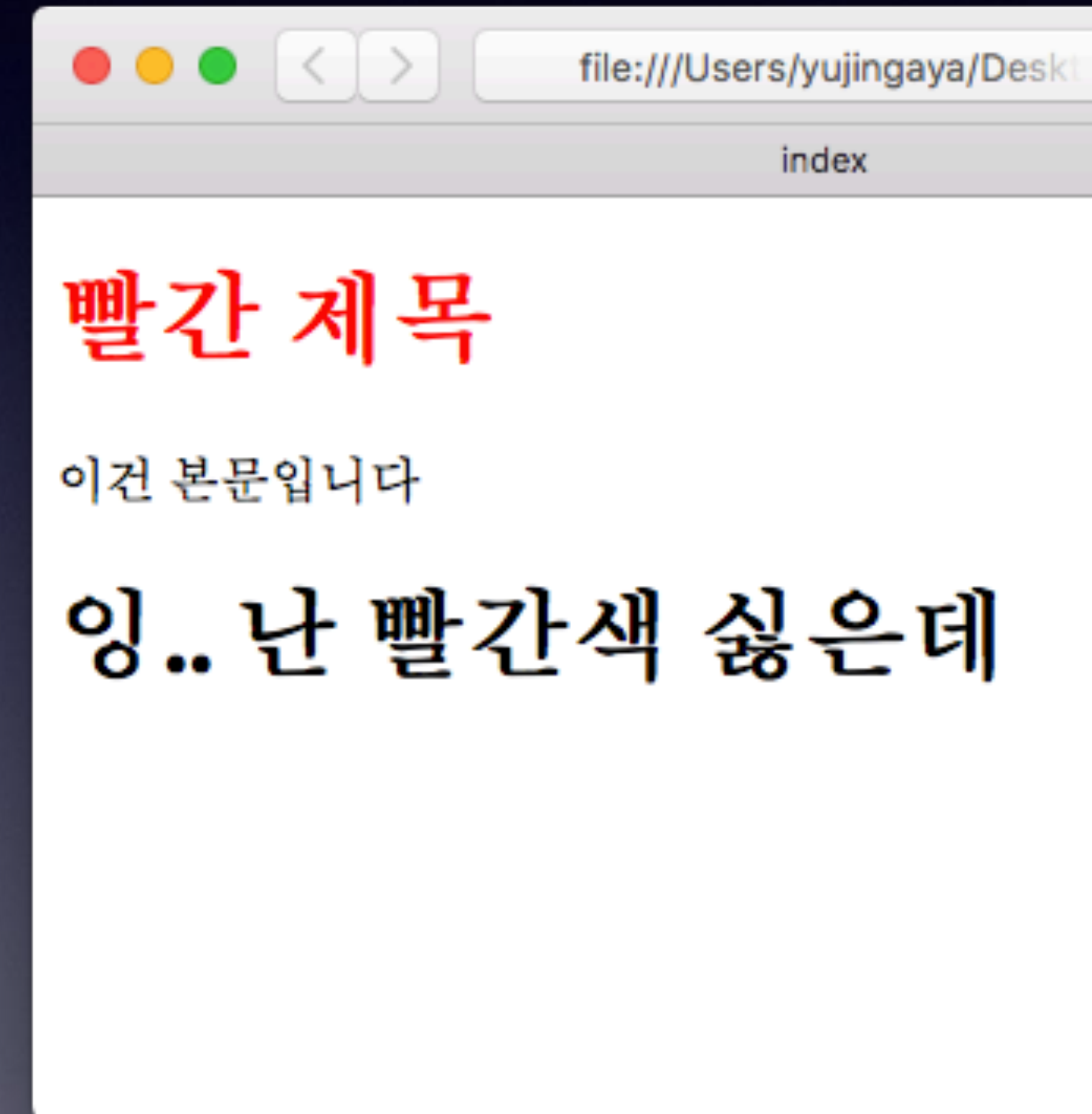
# stylesheet file & link

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="theme.css">
  </head>
  <body>
    <h1 class="red-title">빨간 제목</h1>
    <p>이건 본문입니다</p>
    <h1>잉.. 난 빨간색 싫은데</h1>
  </body>
</html>
```

theme.css

```
h1.red-title {
  color: red;
}
```



# stylesheet file & link

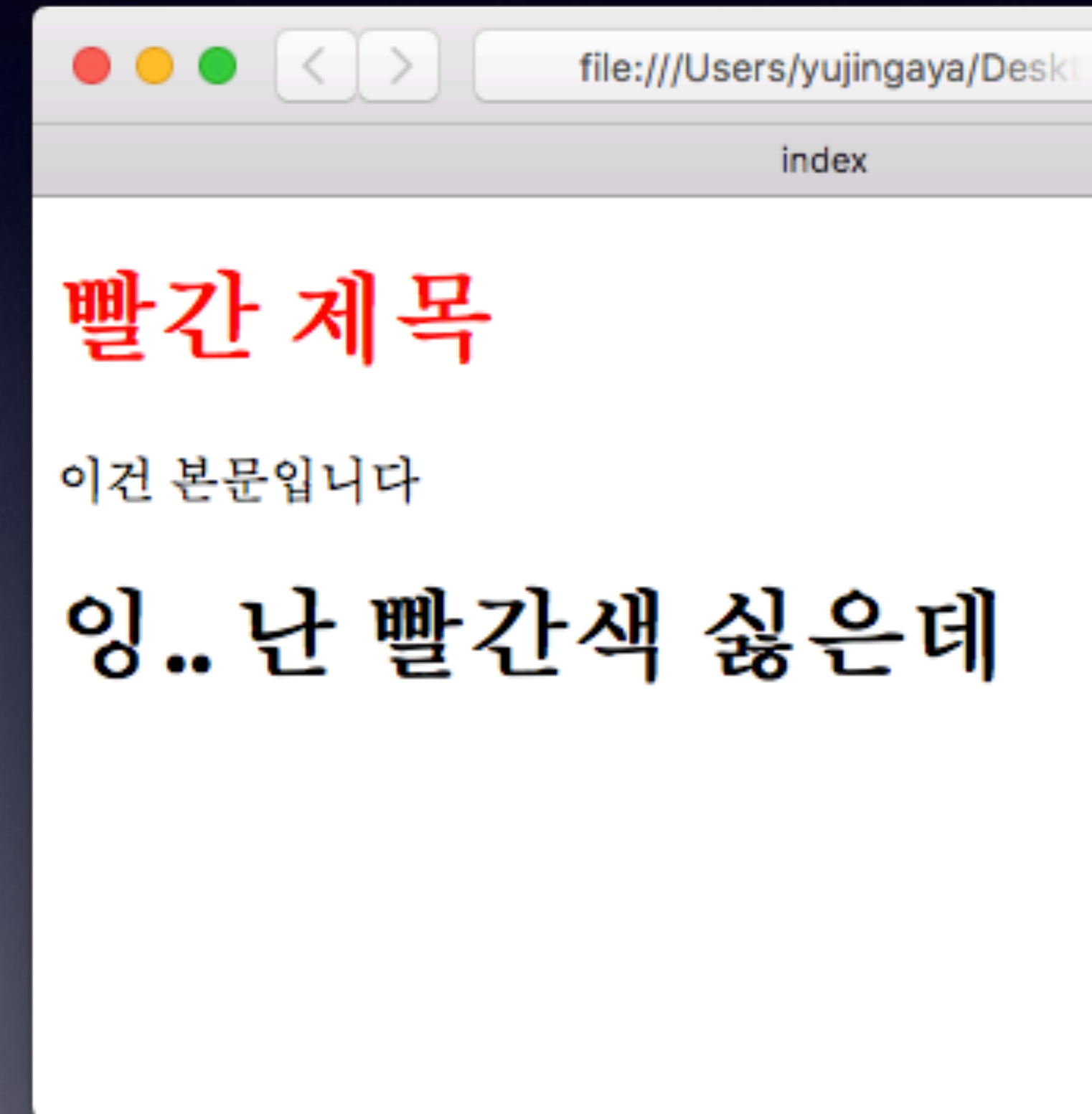
index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="theme.css">
  </head>
  <body>
    <h1 class="red-title">빨간 제목</h1>
    <p>이건 본문입니다</p>
    <h1>잉.. 난 빨간색 싫은데</h1>
  </body>
</html>
```

theme.css

```
h1.red-title {
  color: red;
}
```

another-page.html    yet-another-page.html



# CSS Selector

## Multiple selector

// a 이면서 .link 이면서 .is-red 이면서 .has-shadow ...

```
a.link.is-red.has-shadow { ... }
```

```
<a class="link is-red has-shadow">
```

# CSS Selector

## Descendant selector

// .nav **아래**의 .nav-item

```
#nav .nav-item { ... }
```

```
<nav id="nav">  
  <div class="nav-menu">  
    <a class="nav-item">
```

## Child selector

// .content **바로** 아래의 h2

```
.content > h2 { ... }
```

```
<div class="content">  
  <h2>
```

Adjacent sibling(+), General sibling(~), Attribute([]), ... 등등 다양함!

# CSS 스타일 적용의 우선 순위

`!important`<sub>(흑마법)</sub> > inline > `#id` > `.class` > tag > \* ...

## !important 용법

```
h1 { color: red !important }
```

다른 규칙을 override 하기 위해 사용

~~나는 남의 코드를 읽기 싫다..~~

# CSS 스타일 적용의 우선 순위

`!important`<sub>(흑마법)</sub> > inline > `#id` > `.class` > tag > \* ...

## !important 용법

```
h1 { color: red !important }
```

다른 규칙을 override 하기 위해 사용

~~나는 남의 코드를 읽기 싫다..~~

# CSS 스타일 적용의 우선 순위

`!important`<sub>(흑마법)</sub> > inline > `#id` > `.class` > tag > \* ...

1000의 자리, 100의 자리, 10의 자리, 1의 자리처럼 작동

`#id` 3개 vs. `#id` 2개 => `#id` 3개

`#id` 1개 vs. `.class` 10개 => `#id` 1개



# CSS 스타일 적용의 우선 순위

`!important`<sub>(흑마법)</sub> > inline > `#id` > `.class` > tag > \* ...

1000의 자리, 100의 자리, 10의 자리, 1의 자리처럼 작동

`#id 3개` vs. `#id 2개` => `#id 3개`

`#id 1개` vs. `.class 10개` => `#id 1개`

## Remind

( id, class 여러 개로 선택 범위를 줄일 수 있음 )  
e.g.) `h1.title.is-red.has-shadow { ... }`

# Box Model

Block level element vs. Inline element

<div>, <h1>, <p>, <header>, ...

<span>, <a>, <em>, ...

<h1>

헤딩에서

<p>

본문으로 넘어갈 때는 줄넘김이  
발생합니다

링크는 줄넘김이 발생하지 않습니다

<a>

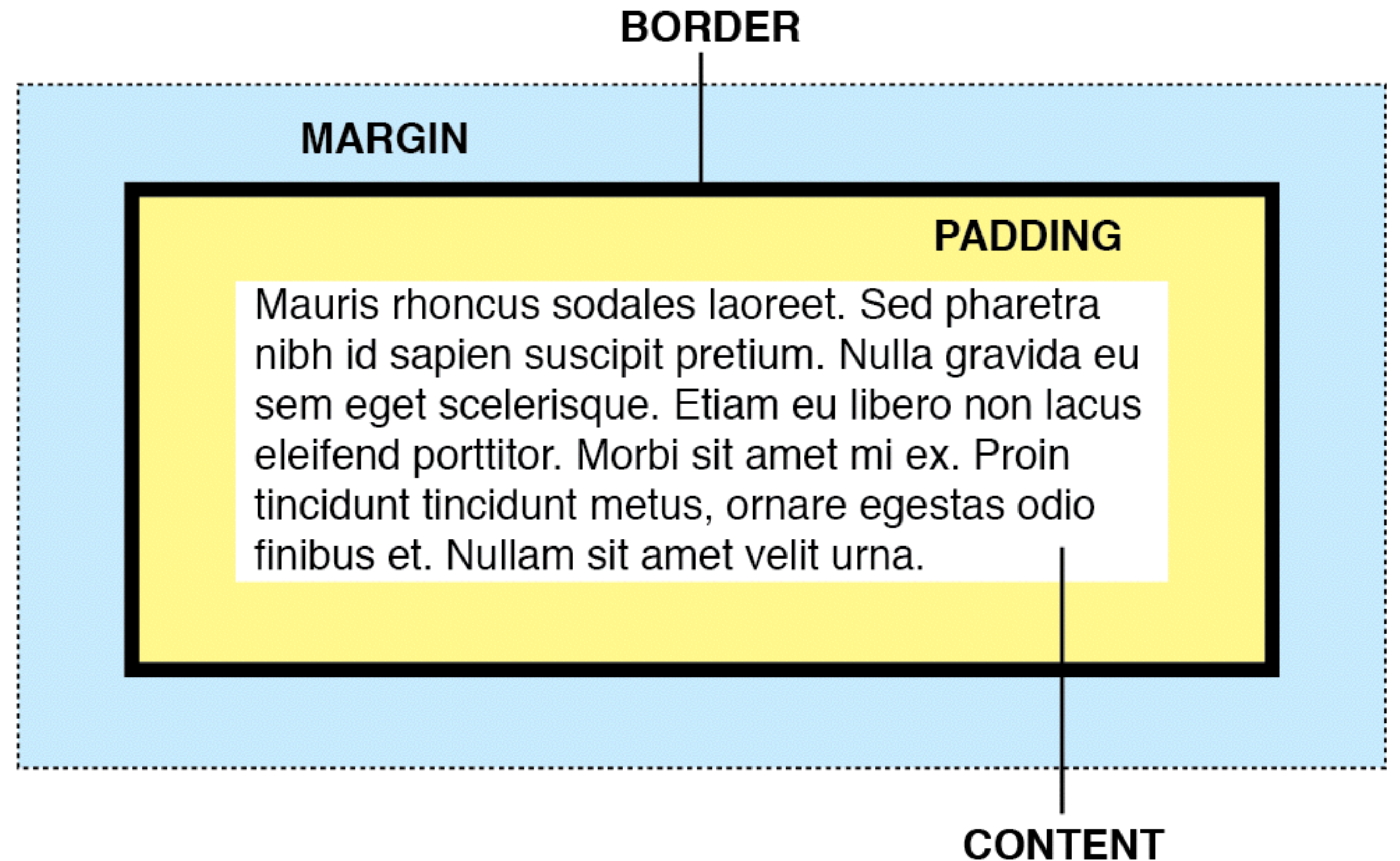
<p>

# Box Model

Margin: 바깥쪽 여백

Padding: 안쪽 여백

Border: 경계선 (두께가 0일 수 있음)



# 이 외에도 수많은 규칙들이 있습니다!

- <http://cssreference.io>: 사전
- <http://caniuse.com>: xx 브라우저에서 작동하나요?
- <http://ko.learnlayout.com>: 튜토리얼 (추천)
- <http://jgthms.com/web-design-in-4-minutes/>: 웹 디자인 시작하기 (추천)
- <http://marksheet.io>: HTML&CSS 가이드
- <https://www.codecademy.com/learn/learn-css>: (숙제)

# 요약

태초에 웹이 있으니.. 연구원들끼리 보고서 돌려보려고 한거였음

근데 그걸로 별 짓을 다 하고있음!!

사실 HTML&CSS 로는 다 못 해요.. ㅎㅎ.. JS 세미나를 기대하세요!

그리고 숙제

- [codeacademy](#) HTML, CSS Pro 제외하고 다 풀어오기
- 추석 이후 첫 세미나까지

질문?