hci+d lab.

Week 07 Extra • 소셜네트워크 데이터마이닝과 분석

HTML & CSS

Joonhwan Lee
human-computer interaction + design lab.

오늘 다룰 내용

- HTML 기초
- CSS 기초
- HTML의 발전, HTML 5

1. HTML 기초

HTML

- HTML 마크업은 HTML 요소(elements) 와 그들의 속성(attributes), 변수(arguments) 등으로 이루어져 있다.
- HTML 마크업은 미리 지정된 태그(tag) 명령어들을 포함하고 있는데, 이 태그는 DTD (Document Type Definition) 에 정의된 바를 따른다.

HTML 의 구성 요소

- 요소 (Elements)
  - HTML 에서 태그 (tag) 로 둘러 싸인 모든 명령어들

```
<p>HTML 을 시작해 보자. </p>

<img src="http://www.joonhwan.com/images/test.jpg">

<div>
  <p>겹겹이 쌓인 태그</p>
</div>
```

HTML 의 구성 요소

- 태그 (Tag)
  - 일종의 요소 (Elements)
  - 시작태그와 종료태그의 두 가지가 있음
  - 일부 태그 중에는 종료 태그가 없는 경우도 있음

```
<p>HTML 을 시작해 보자. </p>
<img src="http://www.joonhwan.com/images/test.jpg">
<div>
  <p>겹겹이 쌓인 태그</p>
</div>
```

HTML 의 구성 요소

- 속성 (Attributes)
  - 요소 (Elements)의 시작 태그 안에서 사용되어 태그의 내용을 좀더 풍부하게

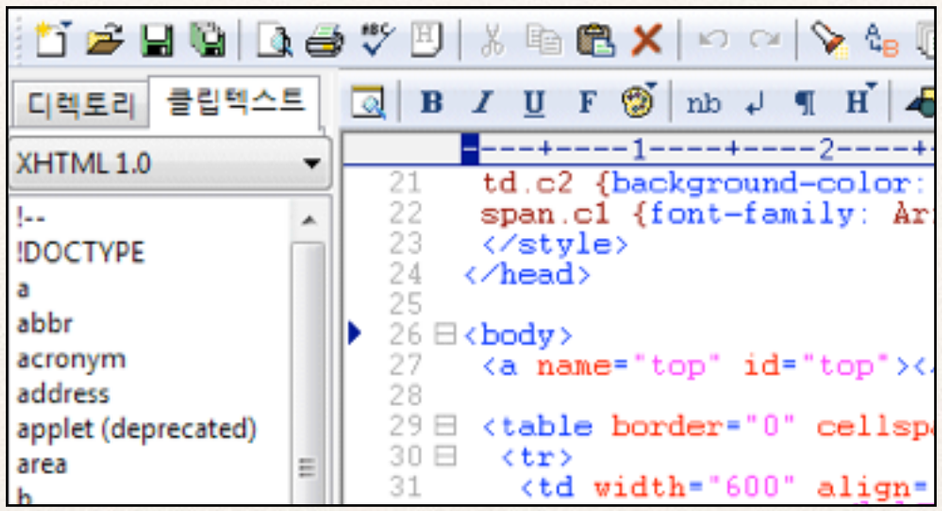
```
<p align=”center”>태그와 함께 사용된 태그의 속성</p>
```

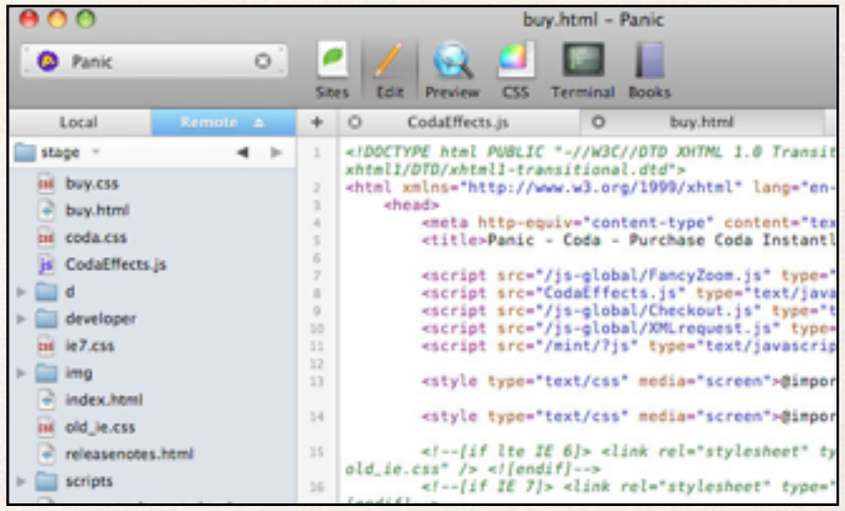
- 변수 (Arguments)
  - 속성과 관련된 값

태그와 함께 사용된 태그의 속성

HTML 의 구성 요소

- HTML 문서 작성
  - 선호하는 아무 텍스트 에디터나 사용 가능
  - HTML 전문 텍스트 에디터
    - 요소, 속성에 대한 DB를 가지고 있어서 문서 작성이 용이
    - 신택스를 칼라코드화 → 문서를 손쉽게 이해할 수 있음
  - 에디트플러스 (PC), Coda (Mac), TextMate (Mac) 등





HTML 의 구성 요소

- HTML 문서 작성
- 드림위버/나모웹에디터
  - drag & drop 으로 HTML 코드 생성 가능
  - 코드의 관리, 유지, 보수가 힘들고 사이즈가 크다
  - 웹 표준화에 맞는 코드 생성이 어렵다

HTML 기초 구조

```
<html>
    <head>
        <title>Hello HTML!</title>
    </head>
    <body>
        My First HTML Page!
    </body>
</html>
```

HTML 기초 구조

```
<html>
    <head>
        <title>Hello HTML!</title>
    </head>
    <body>
        My First HTML Page!
    </body>
</html>
```

문서가 HTML임을 알리는 태그

문서의 속성,
타이틀에 대한 정보

본문

HTML 기초 태그

- `<p> ... </p>`
  - paragraph
- `<br>`
  - 줄바꿈 (line break)
- `<hn> ... </hn>`
  - 표제 (heading)
  - 예. `<h1>타이틀</h1>, <h4>소제목</h4>`

HTML 기초 태그

- `<img>`
  - HTML 문서에 이미지 삽입할 때 사용
  - 속성: width, height, border 등
    - `<img src= “이미지경로” width= “넓이” height= “높이”>`
- `<a> ... </a>`
  - 텍스트, 이미지 등에 링크를 걸 때
  - 속성: target, 변수: _blank, _top 등
  - `<a href= “링크주소” target= “_blank”>링크이름</a>`

HTML 기초 태그

-
  - 글자의 크기, 칼라 등의 속성을 변화 시킬 때 사용
  - 속성: color, size, face 등
  - 변수: red, blue 등 칼라 이름, #123456, #FFFFFF 등과 같은 칼라 값, 폰트 이름
  - <img src= “이미지경로” width= “넓이” height= “높이”>
  - 빨간 색 글자

HTML DOCTYPE

- **HTML 4.01 Strict**
  4.01 표준을 엄격하게 지키는 doctype. <font> 와 같이 deprecated 된 태그는 사용하지 않음. Frameset 지원 X
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

- **HTML 4.01 Transitional**
  4.01 표준을 지원하나 이전 버전의 deprecated 태그 역시 사용가능. Frameset 지원 X
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

HTML DOCTYPE

- **HTML 4.01 Frameset**

  4.01 Transitional 과 동일. Frameset content 를 지원

  ```
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
  ```

- **XHTML 1.0 Strict**

  4.01 표준의 HTML 지원. deprecated 태그 지원 X.

  Frameset 지원 X

  문서는 XML 표준을 따른다.

  ```
  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
  ```

HTML DOCTYPE

- **XHTML 1.0 Transitional**

  4.01 표준의 HTML 지원. deprecated 태그 지원. Frameset 지원 X

  문서는 XML 표준을 따른다.

  ```
  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
  ```

- **XHTML 1.0 Frameset**

  XHTML 1.0 Transitional 과 동일하나 Frameset 지원

  문서는 XML 표준을 따른다.

  ```
  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
  ```

HTML DOCTYPE

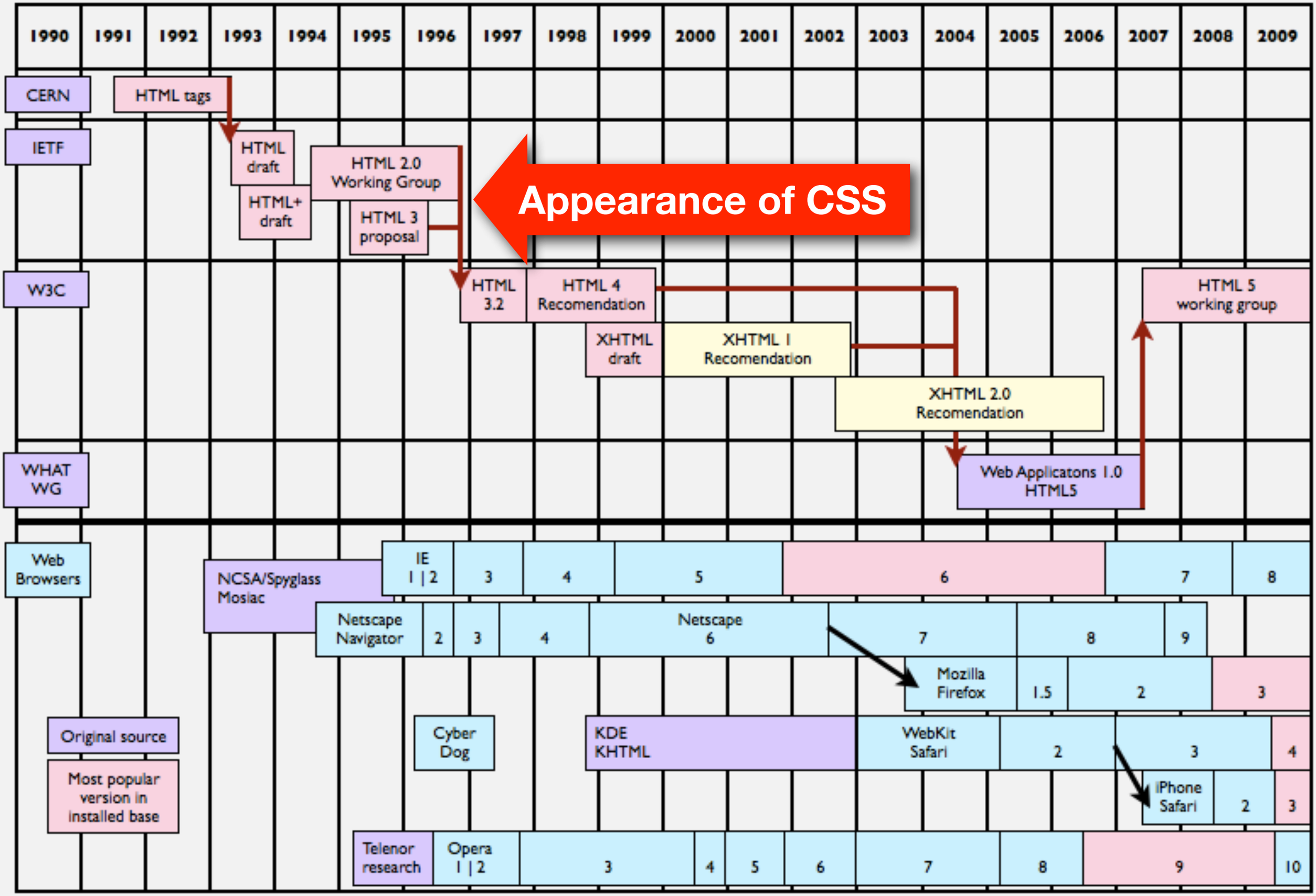
- **XHTML 1.1**
  XHTML 1.0 Strict 와 동일하나 모듈 기능을 제공 (예: ruby support for East-Asian language)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

HTML 기초 태그

- HTML 4.0 : http://www.w3schools.com/html/
- HTML Examples: http://www.w3schools.com/html/html_examples.asp
- HTML Entity Names: http://www.w3schools.com/tags/ref_entities.asp

2. CSS 기초


1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
CERN
HTML tags
IETF
HTML draft
HTML+ draft
HTML 2.0 Working Group
HTML 3 proposal
Appearance of CSS
W3C
HTML 3.2
HTML 4 Recomendation
XHTML draft
XHTML 1 Recomendation
XHTML 2.0 Recomendation
HTML 5 working group
Web Applicatons 1.0 HTML5
WHAT WG
Web Browsers
NCSA/Spyglass Mosiac
IE 1 | 2
3
4
5
6
7
8
Netscape Navigator
2
3
4
Netscape 6
7
8
9
Mozilla Firefox
1.5
2
3
Cyber Dog
KDE KHTML
WebKit Safari
2
3
4
iPhone Safari
2
3
Telenor research
Opera 1 | 2
3
4
5
6
7
8
9
10
Original source
Most popular version in installed base

CSS (Cascading Style Sheets)

- Style Sheets 은 1970년대 SGML 시대 부터 존재.
- HTML 이 성장하면서 문서의 스타일에 대한 관심이 증대
  → HTML 에서 스타일을 분리할 필요성 인식
  → W3C에서 CSS 표준화 작업 시작 (2 were chosen from 9 proposals)
- W3C 내에서 웹기술 표준화 작업이 세 그룹으로 나뉘어져 진행 (1997)
  - HTML Working Group
  - DOM Working Group
  - CSS Working Group

CSS (Cascading Style Sheets)

- CSS 1 (12/1996)
  - Font properties (typeface, emphasis...)
  - Color (text, backgrounds, line...)
  - Text attributes (spacing: word, letter, text line)
  - Alignment (text, image, table, etc..)
  - Margin, border, padding, positioning

CSS (Cascading Style Sheets)

- CSS 2 (5/1998)
  - 현재, W3C 가 지정한 공식 표준 (버전 2.1)
  - CSS 1을 기반으로 레이아웃에 필요한 스타일이 추가로 포함
    - absolute, relate, fixed positioning of elements
    - new font properties (e.g., shadow)
- CSS 3
  - 12/2005 이후 현재 계속 개발 중
  - border style (round, image border), transparency, font embedding 등 추가

CSS (Cascading Style Sheets)

- Browser
  - 1996년 CSS1 완성 이후 발표 된 IE3 가 부분적으로 CSS 지원
  - Macintosh 용 IE5 CSS1 을 완벽하게 지원 → MS에 의해 개발 중단
  - Netscape 역시 Version 5 이후에야 CSS를 제대로 지원
  - IE는 Version 7 이후 부터 CSS 지원
  - 여전히 CSS2 를 완벽하게 지원하는 브라우저는 없음
  - CSS3 는 IE9, Safari 4, FireFox 4 이후 부터 (거의) 지원
    - CSS3 가 아직 완성 단계가 아니라 지원여부도 완벽하지 않음

CSS 로 웹문서 꾸미기

- CSS (Cascading Style Sheet)
  - 웹페이지의 시각적 표현을 제어하기 위한 언어
  - HTML, XHTML → 웹페이지의 “구조”를 지정하는 언어
  - HTML 내에서도 “시각적 표현”이 가능하나 최근에는 HTML과 CSS의 역할이 분리

CSS 로 웹문서 꾸미기

- CSS 의 장점
  - 작은 용량, 이해하기 쉬운 구조 → table 코딩에 비해 보통 50% 정도 코드 절약
    - 샘플 레이아웃 코드 참고 (DivLayout.html vs. TableLayout.html)
- 디자인과 웹구조 (데이터)의 완벽한 분리
  - 하나의 HTML 로 다양한 디자인 가능
  - http://www.csszengarden.com
- 웹표준에 맞는 사이트 제작 (w3c.org 에서 웹표준 정의)

CSS 기초

- CSS 코드는 다음과 같은 서식을 따른다.
  - **`selector { property: value; }`**
  - 예:

```
body { color: black; }
body { color: black; font-size: 12pt; }
body {
    color: black;
    background-color: white;
    font-size: large;
    line-height: 140%;
}
```

CSS 기초

- HTML 에 CSS를 사용하는 방법
  - 외부 스타일 시트 연결

```
<head>
   <link rel="stylesheet" type="text/css"
href="mystyle.css">
</head>
```

CSS 기초

- HTML 에 CSS를 사용하는 방법
  - 내부 스타일 시트
  - 모든 HTML 문서마다 스타일 지정해야해서 비효율적

```
<head>
   <style type="text/css">
      <!--
         body {font-size:9pt;}
      //-->
   </style>
</head>
```

CSS 기초

- HTML 에 CSS를 사용하는 방법
  - HTML 태그 내 스타일 지정 (inline styles)
  - 태그마다 스타일을 지정해야 해서 비효율적
  - 직관적이어서 한 두개 정도 지정해야 할 때는 유용

```
<p style="color:gray;">text color is
gray.</p>
```

CSS 기초

- 선택자(selector): 클래스(class)와 아이디(id)
  - 클래스 (class)
    - . 으로 지정 (예: .red, .italic...)
    - 한 문서에 중복 사용 가능

```
<style type="text/css">
   <!--
      .red {color:red}
   //-->
</style>

<h3 class="red">color of heading3 set to red.</h3>
<p class="red">paragraph is also red.</p>
```

CSS 기초

- 선택자(selector): 클래스(class)와 아이디(id)
  - 아이디 (id)
    - #으로 지정 (예: #navigation, #footer...)
    - 한 문서에 단 한번만 사용 가능

```
<style type="text/css">
   <!--
      #navbar { ... }
      #footer { ... }
   //-->
</style>

<div id=”navbar”> .... </div>
<div id=”footer”> .... </div>
```

CSS 기초

- Text 관련 CSS

속성	속성값	설명
color	red, #FF0000	텍스트 색상
direction	ltr, rtl	텍스트 방향
line-height	150%, 10px	줄 간격
letter-spacing	-0.1px	글자 간격
text-align	left, right, center, justify	텍스트 정렬
text-decoration	none, underline, overline, line-through, blink	텍스트 장식
text-indent	20px	들여쓰기
text-transform	none, capitalize, uppercase, lowercase	대소문자 지정
word-spacing	1px	단어 간격

CSS 기초

- Font 관련 CSS

속성	속성값	설명
font-family	"Lucida Grande", Lucida, Verdana, sans-serif	글자체
font-size	10pt, 10em	글자크기
font-weight	bold, normal	글자두께
font-variant	small-caps, none	Small Caps
font-style	italic, none	글자스타일 (이탤릭)

CSS 기초

- 단위

절대단위	상대단위
pt (포인트, 1pt= 1/72 in)	% (기준이 되는 글꼴에 대한 퍼센트)
in (인치, 1in = 25.4 mm)	em (기준이 되는 글꼴에 대한 문자의 높이)
mm (밀리미터)	
cm(센티미터)	
pc (파이카, 1pc=12pt)	
px (픽셀,1px=모니터의 1도트)	

CSS 기초

✦ 배경관련 CSS

속성	값	설명
background-color	#FFFF80, Ivory	배경색을 지정
background-image	url(bg.gif)	배경이미지를 지정
background-repeat	repeat, repeat-x, repeat-y, no-repeat	배경이미지의 반복 여부를 지정
background-position	top left, top center, top right, center left, center center, center right, bottom left, bottom center, bottom right	배경이미지의 위치를 지정
background-attachment	scroll, fixed	배경이미지의 스크롤 여부를 지정

CSS 기초

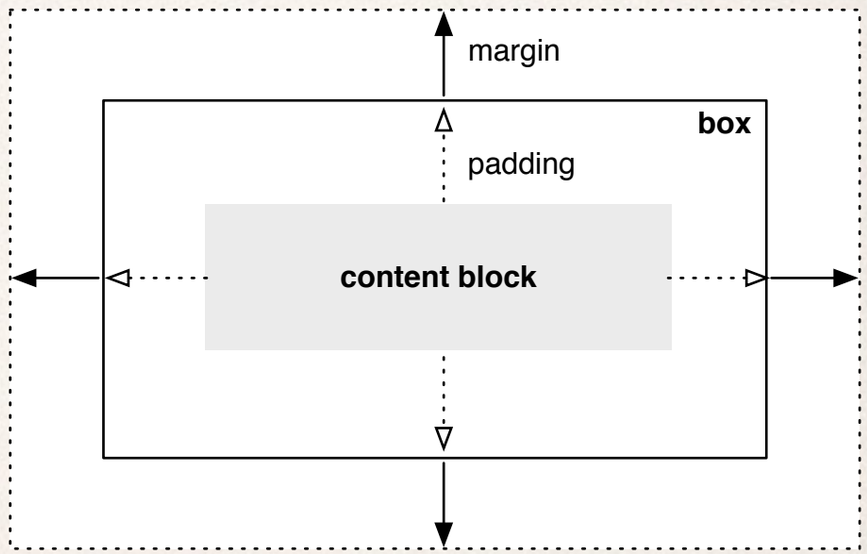
✦ 경계선관련 CSS

속성	값	설명
border-style	none, solid, dotted, hidden, dashed, double, groove, ridge, inset, outset	경계선의 스타일을 지정
border-width	1px, thin, medium, thick	경계선의 두께를 지정
border-color	#000000, silver	경계선의 색을 지정
border-top **border-bottom** **border-left** **border-right** **border**	e.g.) border-top: 2px dotted silver; border-right: 4px double red; border-left: 4px groove green; (style, width, color를 붙여서 사용가능)	박스에서 상/하/좌/우 경계선을 나누어서 지정할 때 사용

CSS 기초

- Box Model:
  Content, Margin, Padding, Border
  - content
  - padding: 콘텐츠와 경계선 (border) 사이의 여백
  - border: 경계선
  - margin: 경계선 밖으로의 여백
  - 예:

```
margin-left: 10px;
padding-top: 5px;
margin: 20px;
```



CSS 기초

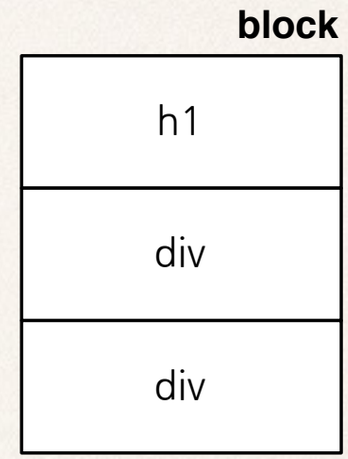
- Sample HTML/CSS
  - TextExample.html / TextStyle.css
  - BackgroundExample.html / BackgroundExample.css
  - BoxExample.html / BoxExample.css

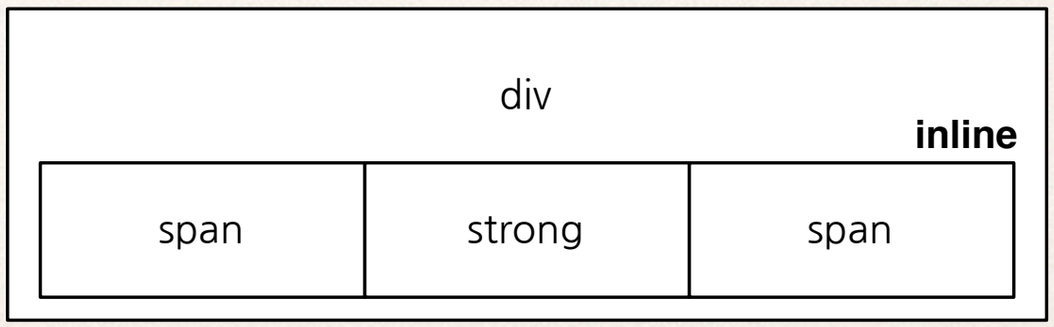
- property 참조: http://www.w3.org/TR/CSS21/

Positioning

- 화면 표시 모델
  - p, h1, div: block element
  - strong, span: inline element
  - display 속성으로 block 혹은 inline 으로 바꿀 수 있다

→ display: none, display: block



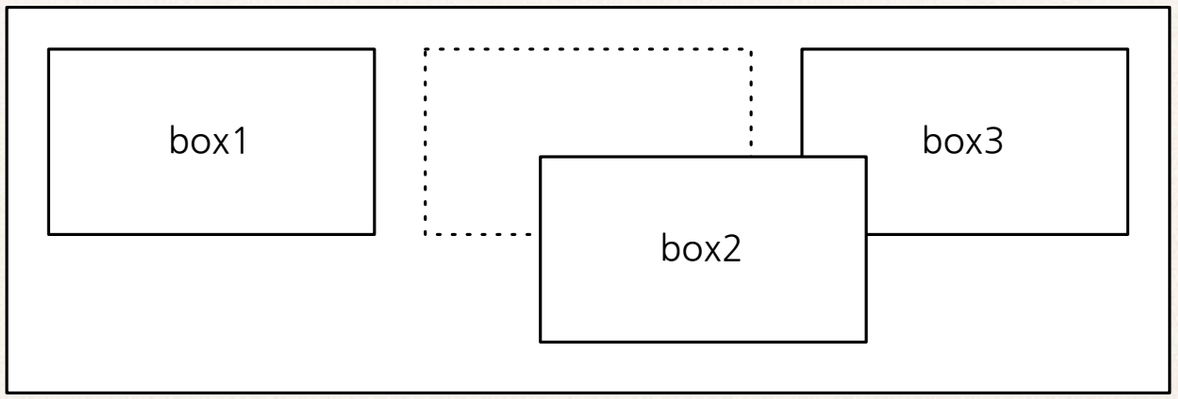


Positioning

- Positioning의 3가지 방식
  - 상대위치(relative): 박스의 시작점이 이전 박스 다음으로 지정.
  - 절대위치(absolute): 브라우저 내에서 (0, 0) 을 기준으로 절대적인 좌표에 박스의 위치를 지정.
  - 플로팅(floating): 박스를 다른 박스 왼쪽이나 오른쪽 끝에 닿도록 이동.

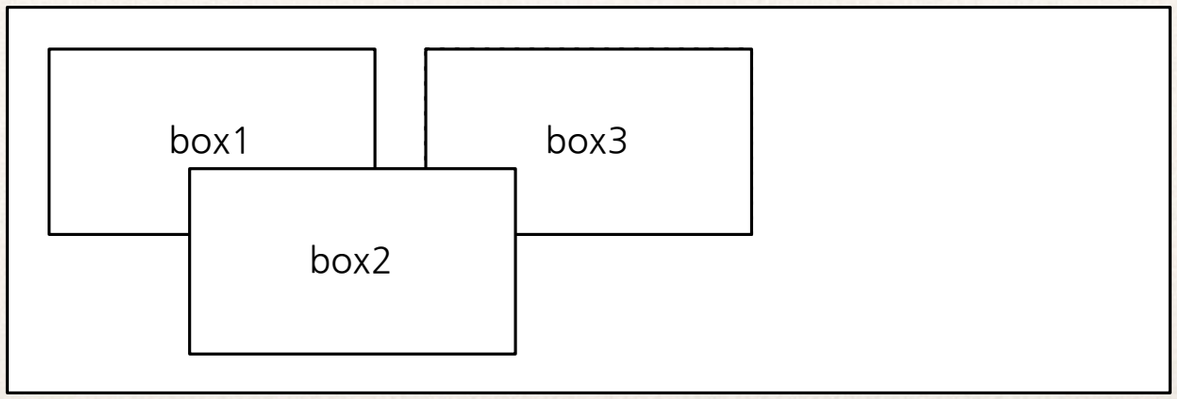
Positioning

- 상대위치 (relative positioning)
  - 엘리먼트의 위치가 상대적으로 결정.
  - 상대위치 상태의 엘리먼트의 위치(position) 을 이동시키면 원래 위치는 그대로 가지고 있는 상태에서 해당 엘리먼트의 시작점만 이동 → 다른 엘리먼트와 중첩된다.
- Lab2/w8-flow-01.html



Positioning

- 절대위치 (absolute positioning)
  - 절대위치로 지정된 엘리먼트는 문서 내에서 절대적인 위치를 가짐
  - 주로 문서의 좌상단이 (0,0) 으로 지정되고 그 위치에서 부터 position 이 결정됨
  - 다른 (상대적으로 지정된) 엘리먼트는 절대적으로 지정된 엘리먼트를 무시하고 레이아웃을 결정
  - Lab2/w8-flow-02.html



Positioning

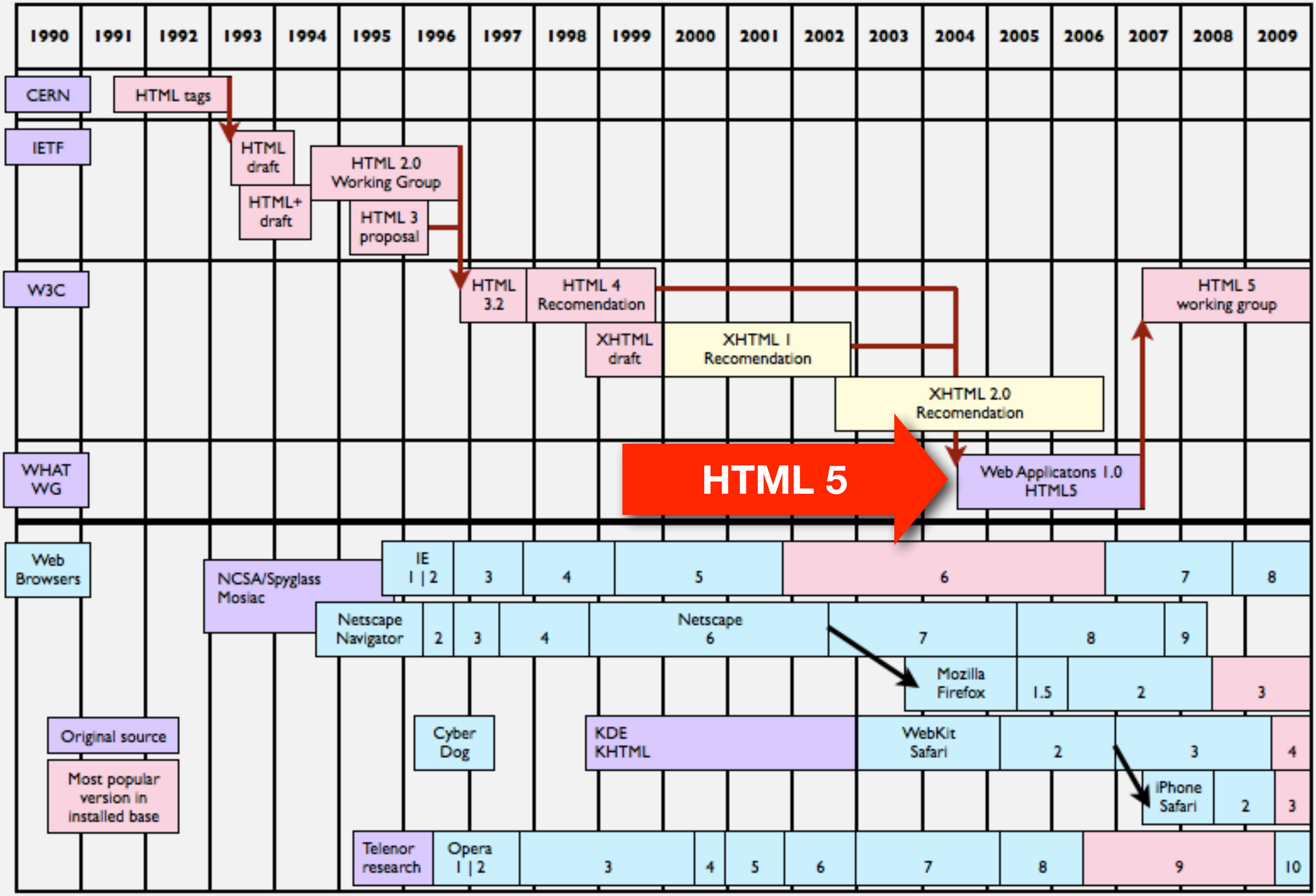
- 플로팅 (floating)
  - 엘리먼트를 플로팅으로 지정하면 해당 엘리먼트가 오른쪽이나 왼쪽 끝면에 닿도록 이동시킬 수 있다.
  - 플로팅된 엘리먼트는 문서의 일반흐름을 타지 않는다 → 다른 엘리먼트들은 플로팅된 엘리먼트가 없는 것처럼 동작.

- 플로팅은 clear 속성으로 해제한다

  `clear: both;`

- Lab2/w8-flow-03.html

3. HTML 5


1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
CERN
HTML tags
IETF
HTML draft
HTML+ draft
HTML 2.0 Working Group
HTML 3 proposal
W3C
HTML 3.2
HTML 4 Recomendation
XHTML draft
XHTML 1 Recomendation
XHTML 2.0 Recomendation
HTML 5 working group
HTML 5
WHAT WG
Web Applicatons 1.0 HTML5
Web Browsers
NCSA/Spyglass Mosiac
IE 1 | 2
3
4
5
6
7
8
Netscape Navigator
2
3
4
Netscape 6
7
8
9
Mozilla Firefox
1.5
2
3
Original source
Most popular version in installed base
Cyber Dog
KDE KHTML
WebKit Safari
2
3
4
iPhone Safari
2
3
Telenor research
Opera 1 | 2
3
4
5
6
7
8
9
10

HTML의 발전

- HTML이 등장한 이후 몇 차례의 버전업
  - 개발자
    - 개발할 수 있는 태그가 많아짐 → 다양한 시각화 가능
    - 웹 어플리케이션으로 할 수 있는 일들이 많아짐
  - 이용자
    - 보다 편리하고 시각적으로 풍부한 웹 사이트
    - 플래시/자바 등의 의존 없이 동영상과 에니메이션 가능
    - 오프라인 웹 어플리케이션 사용 가능
- **도큐멘트가 아니라 웹 어플리케이션을 위한 플랫폼으로 진화**

WHATWG의 탄생

- HTML의 버전업 정지와 XHTML의 추진
  - 1998년 W3C, HTML 표준 업데이트 중단 선언 → XHTML 표준화 작업
  - XHTML 보급의 둔화
    - XHTML 문법을 따르지 않은 XHTML 페이지의 범람 (HTML 4.01 과 혼동)
    - XHTML의 MIME 타입을 바르게 다루지 않는 브라우저가 시장의 절반을 점유 (XHTML 문서도 HTML 문서로 처리)
- Ajax, Web 2.0 의 등장으로 웹 어플리케이션 보급 확대

  → 주로 HTML 플랫폼을 기반으로 개발

  → 개발 중단된 HTML 사양이 웹 어플리케이션 가능성 제한하는 상황으로...

WHATWG 의 탄생

- 보급이 둔화된 XHTML보다 HTML의 업데이트를 재개해야 한다는 의견 대두
  → W3C로 부터 제안 거절 (XHTML 방향성과 병치되는 제안)
- 2004년, Apple, Mozilla, Opera가 WHATWG (Web Hypertext Application Technology Working Group) 발족
  - HTML 진화시키려는 노력 지속 → 결과물을 표준화 단체에 제출하는 것을 목적
    - 널리 보급된 HTML에 기초할 것
    - 브라우저에 실제 구현된 기능을 중시할 것
      (표준화 되지 않은 기술을 표준화로 이끌어내려는 현실적인 방안)

W3C 와 HTML5 표준화

- 2007년, W3C는 워킹그룹을 발족하고 WHATWG와 공동으로 HTML5 제정에 참여하기로 결정
- 2008년 1월, W3C에 의한 HTML5 초안 공개
- 2009년 Google I/O, “구글의 차세대 웹 기술로 지원하겠다”
- 2009년 Apple Steve Jobs, “플래시 지원 않겠다. HTML5는 대응 기술”

→ HTML5가 개발자와 일반인에게 알려지게 된 계기

“HTML5 is not one big thing”

HTML5 ~= HTML + CSS + Native JS APIs Text

HTML5의 새로운 점?

- Rich Internet Application Support
- Semantic Markup
- Better Accessibility
- Better Compatibility

Rich Internet Application

- HTML: 문서 작성을 위한 기술. 어플리케이션 작성하기 위한 기능들이 거의 제공되지 않음
- HTML5: 어플리케이션 작성을 위한 플랫폼 → 해당 API의 제공
  - 동영상, 음성 재생 (video/audio 요소)
  - 2D 그래픽 처리 (canvas 요소)
  - 오프라인에서 작동되는 어플리케이션 기능 (어플리케이션 캐시, e.g. Google Gears)
  - 도메인 간 통신 구현 (Cross Document Messaging 등)

Rich Internet Application

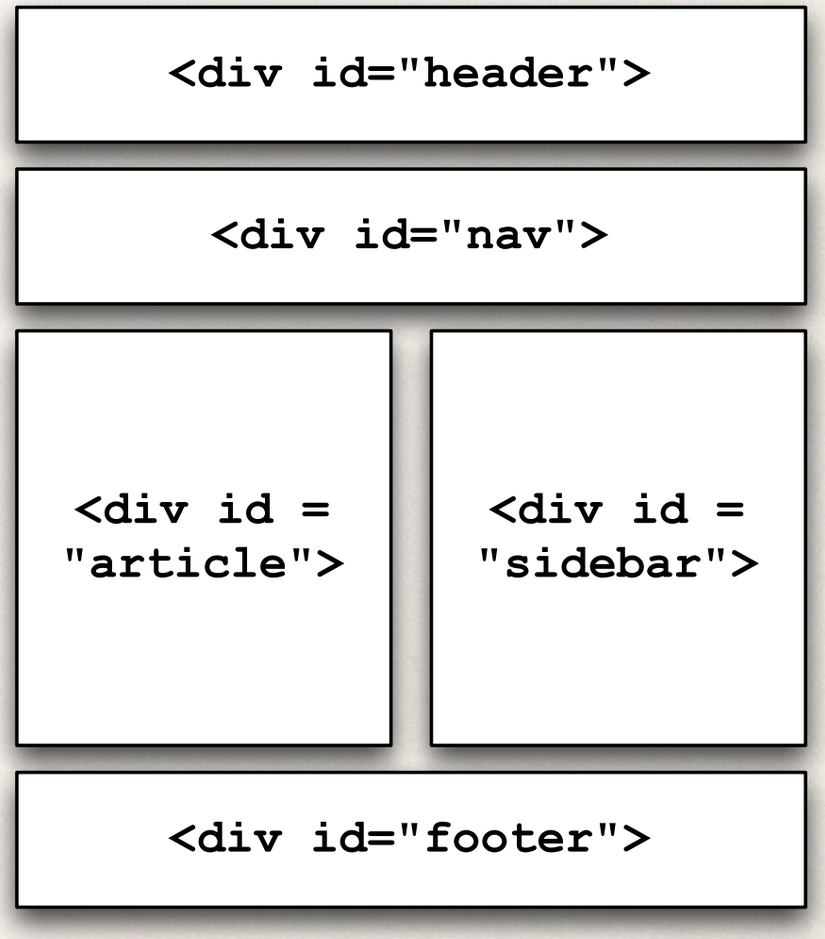
- 클라이언트 측에 데이터 저장 (Web Storage, Web SQL Database)
- 백그라운드 프로세스 (Web Worker)
- 서버로부터 데이터 푸시 혹은 서버와의 쌍방향 통신 (Web Sockets)
- 로컬파일의 내용 읽어들임 (File API)
- 기존의 HTML+JavaScript 로는 불가능했던 기술
- 주로 Java 나 FLEX 에 의존했던 것을 HTML로 구현

Semantic Markup

- HTML5에는 문서 구조의 의미나 문서 내에 삽입된 데이터의 의미 등을 명확히 하기 위한 마크업이 다수 추가
  - 예: <head> <footer> <section> 등의 마크업

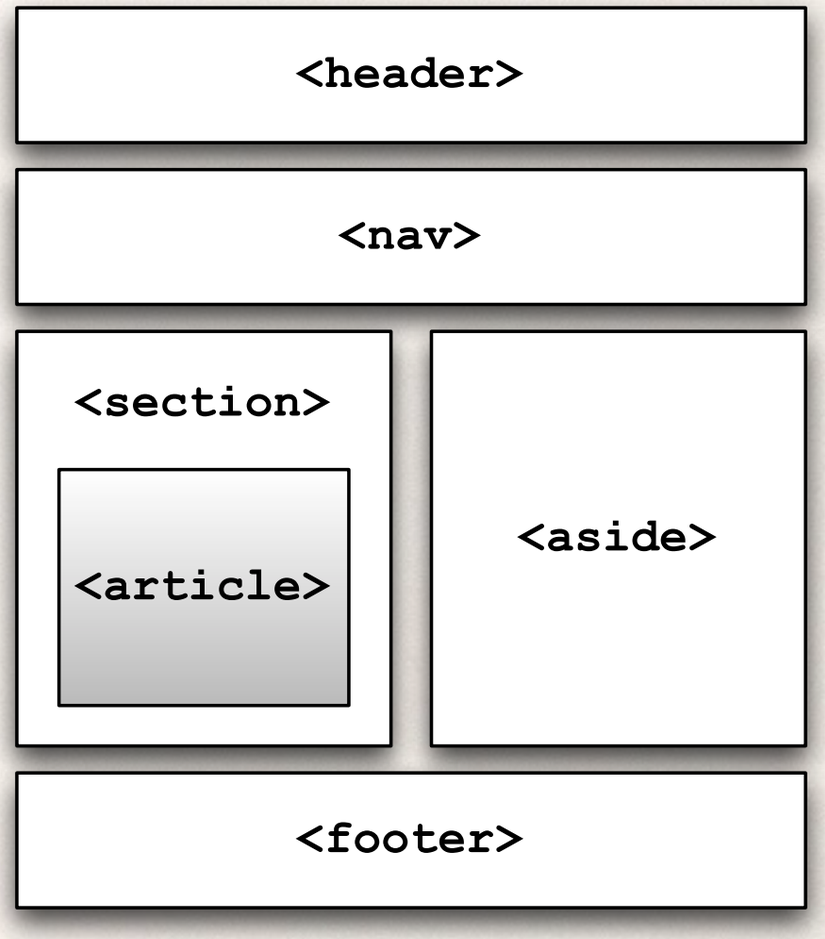
Semantic Markup

```
<html>
<body>
  <div id=”header”>
  </div>
  <div id=”nav”>
  </div>
  <div id=”article”>
  </div>
  <div id=”sidebar”>
  </div>
  <div id=”footer”>
  </div>
</body>
</html>
```



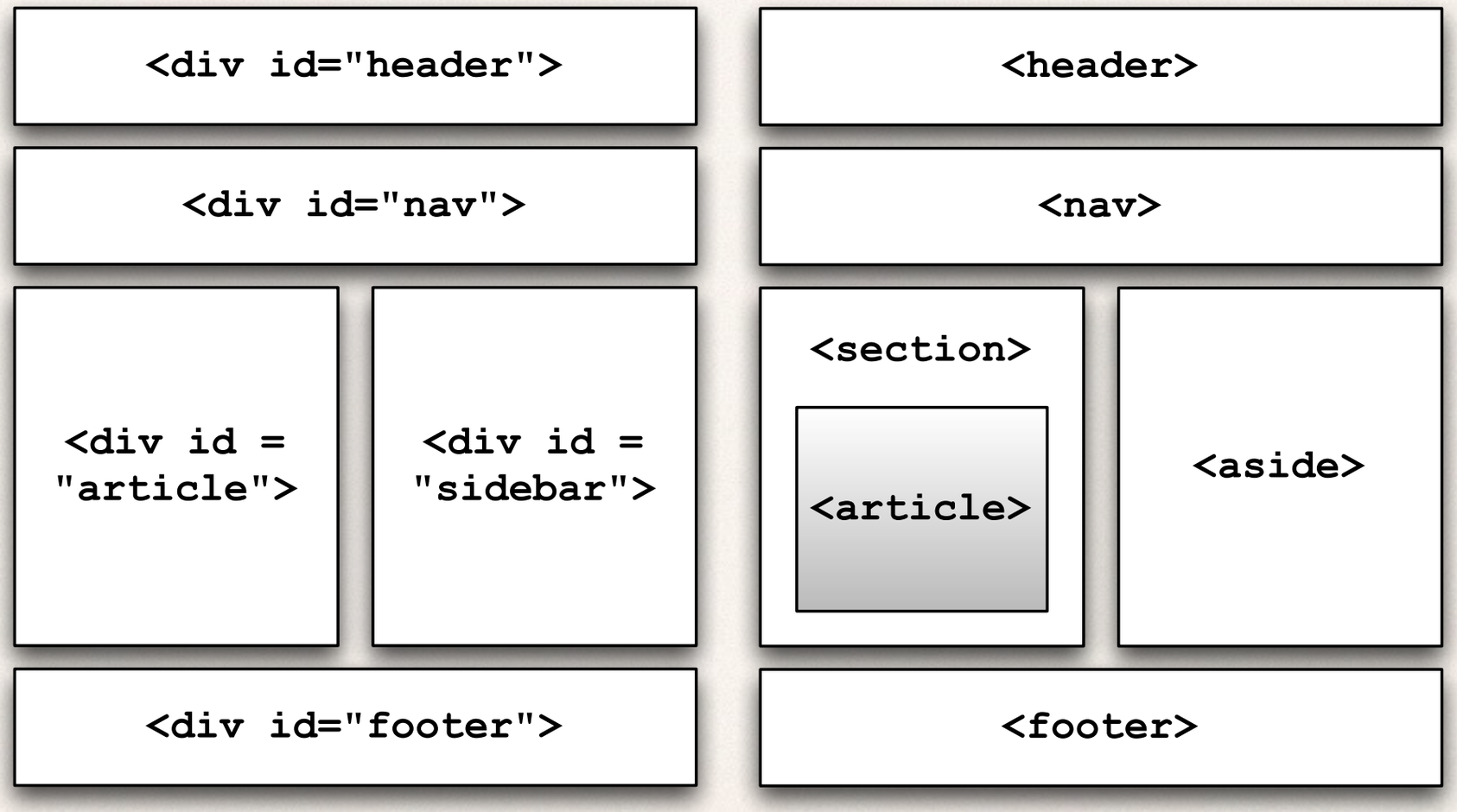
Semantic Markup

```
<html>
<body>
  <header>
  </header>
  <nav>
  </nav>
  <section>
    <article>
    </article>
  </section>
  <footer>
  </footer>
</body>
</html>
```



Semantic Markup

Ordinary HTML markup vs. semantic markup



Semantic Markup

- 기존의 HTML은 문서의 구성을 표현하기 위해서 class 나 id 를 사용
- HTML5 에서는 <head>, <footer>, <section> 등의 요소를 사용하여 문서의 구성을 표현
  - 코딩 시의 가독성 향상 (예: 가 어디서 끝나는지 확인할 필요가 없다)
  - 프로그램이 자동으로 문서의 구조 파악
    - 검색엔진 등이 자동으로 HTML을 파싱하는 기술의 정확도 향상

Better Accessibility

- Accessibility (접근성)
  - 장애가 있는 사람들에게 생활을 둘러싼 여러가지 사물을 이용할 수 있도록 돕는 개념
  - 문서나 어플리케이션 사용의 편의성 (컴퓨터의 경우)
    - 예: 시각 장애자를 위한 문서/인터페이스 읽어주기 기능 (스크린 리더)
- 외국의 경우 법안으로 접근성의 개념을 정의하고 따르도록 권유
  - 예: 미국 (Section 508. http://www.section508.gov/)
- 웹 컨텐츠 → 접근성이 매우 중요하나 기존의 기술은 접근성을 고려하지 않음
  - 예: Navigation Bar 의 구성 시, 버튼, 텍스트 등이 디자이너 마음대로 생성 됨 → 스크린 리더가 작동될 수 없음

Better Accessibility

- HTML5의 접근성
  - 시맨틴 요소 추가 (<head>, <footer>, <section>)
  - 보다 다양한 입력 폼의 제공 (숫자, 문자 입력 키보드 제한 → 모바일 환경에 적합)
- WAI-ARIA (WAI Accessible Rich Internet Application) 사양 포함
  - 웹 컨텐츠의 접근성 향상을 목표로 한 W3C의 사양 중 하나
  - 임의의 요소에 요소의 역할, 상태 등을 나타내기 위한 속성을 부여할 수 있음
    → 스크린 리더 등을 통해 의미를 전달할 수 있다

Better Compatibility

- HTML의 발전과정에서 발생한 문제: 호환성
  - 브라우저 마다 HTML 을 해석하고 렌더링하는 차이가 존재 (최근에는 거의 사라짐)
  - 호환성의 중요성은 버전이 올라갈 때마다 고민해야할 문제

- 버전 간 호환성
- 브라우저 간 호환성

Better Compatibility

- 버전 간 호환성
  - 새로운 버전을 지원하는 브라우저는 이전 버전의 HTML 역시 문제없이 표시하여야

    → HTML5는 많은 요소와 속성이 삭제 됨

    → 이전 버전의 요소와 속성도 새로운 브라우저에서 표시
  - 옛날 버전의 브라우저에서도 HTML5가 문제없이 표시되어야

    → 새롭게 도입된 요소와 속성이 옛날 브라우저에서도 어느 정도 표시되어야 함

    예: <progress value=”80” max=”100”>80%</progress>

    새로운 브라우저 → 프로그레스 바

    옛날 브라우저 → 80% 만 표시

Better Compatibility

- 브라우저 간 호환성
  - 표준화 원칙

    “표준을 따르기만 한다면 어떤 브라우저에서도 똑같이 작동한다”

    → 브라우저의 표준화 준수
  - HTML5의 추가 원칙

    “이미 사용되고 있는 내용에서 사양을 추출한다”

    → IE, FireFox, Safari 등은 표준화와 별도로 자체적인 기능을 탑재

    예: IE의 ruby, Safari의 Canvas

Questions?