

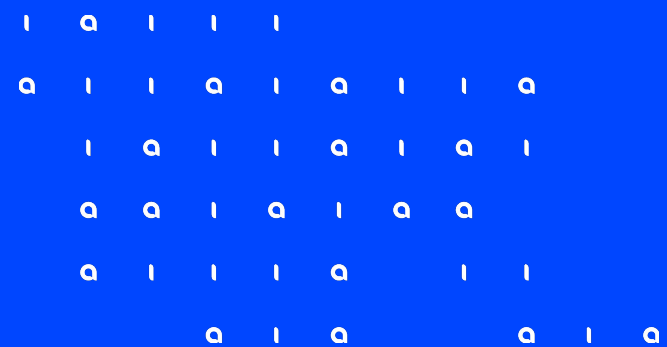
K-디지털 트레이닝
디지털 선도기업
아카데미



[신한투자증권] 프로 디지털 아카데미

클라우드 기반 금융 ICT개발자 양성과정

실무형 웹페이지 실습



목차



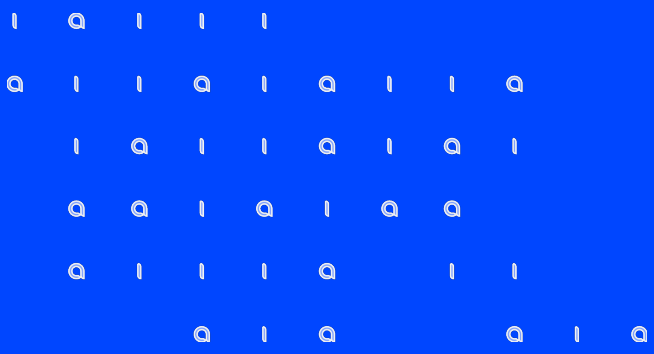
01	—	목차	p
02	—	목차	p
03	—	목차	p
04	—	목차	p
05	—	목차	p
06	—	목차	p
07	—	목차	p
08	—	목차	p
09	—	목차	p
10	—	목차	p

※ 목차만 작성하시면 됩니다.
페이지수는 인쇄소에서 제본전에 수정할 예정입니다.

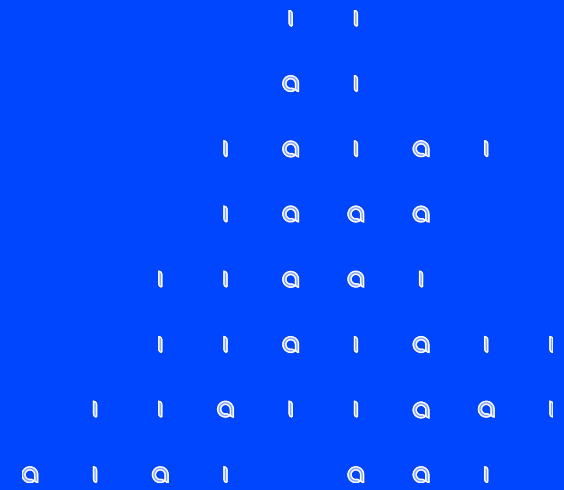
교과목 교육과정

구분

교과목	세부 주제	학습목표	학습내용	학습방법	학습시간
실무형 웹페이지 실습	HTML과 CSS	<ol style="list-style-type: none"> 1. 전반적인 웹의 동작 방식을 파악할 수 있다. 2. HTML과 CSS를 정의할 수 있다. 3. HTML의 다양한 태그들을 사용할 수 있다. 4. 원하는 형태의 CSS 정의할 수 있다. 	<ol style="list-style-type: none"> 1. 웹의 동작방식 2. HTML과 CSS에 대한 정의 3. 대표적인 HTML 태그 4. 대표적인 CSS 태그 	이론	4
	헛갈리기 쉬운 CSS	<ol style="list-style-type: none"> 1. Margin과 Padding의 차이를 알기. 2. 크기를 표현하는 다양한 단위 3. Position 속성 4. Flex 속성과 하위 속성 	<ol style="list-style-type: none"> 1. 헛갈리기 쉬운 다양한 CSS 속성을 짚고 넘어가기 	이론 및 실습	4
	반응형 웹과 미디어쿼리 (부트스트랩)	<ol style="list-style-type: none"> 1. 반응형 웹이 무엇인지 안다. 2. Media query에 대해 이해한다. 3. Breakpoint를 이해한다. 4. Grid System을 사용할 수 있다. 	<ol style="list-style-type: none"> 1. 미디어 쿼리 작성하기 2. 부트스트랩 사용하기 3. 반응형 웹 구축하기 	이론 및 실습	4
	실전 웹페이지 작성하기	<ol style="list-style-type: none"> 1. 스스로 웹사이트 구축을 위한 HTML/CSS를 사용할 수 있다. 2. 필요한 리소스를 온라인상에서 찾아 사용할 수 있다. 		실습	26
총계					4



1. HTML과 CSS



학습목표

- ▶ 전반적인 웹의 동작 방식을 파악할 수 있다.
- ▶ HTML과 CSS를 정의할 수 있다.
- ▶ HTML의 다양한 태그들을 사용할 수 있다.
- ▶ 원하는 형태의 CSS를 정의할 수 있다.

인터넷과 웹

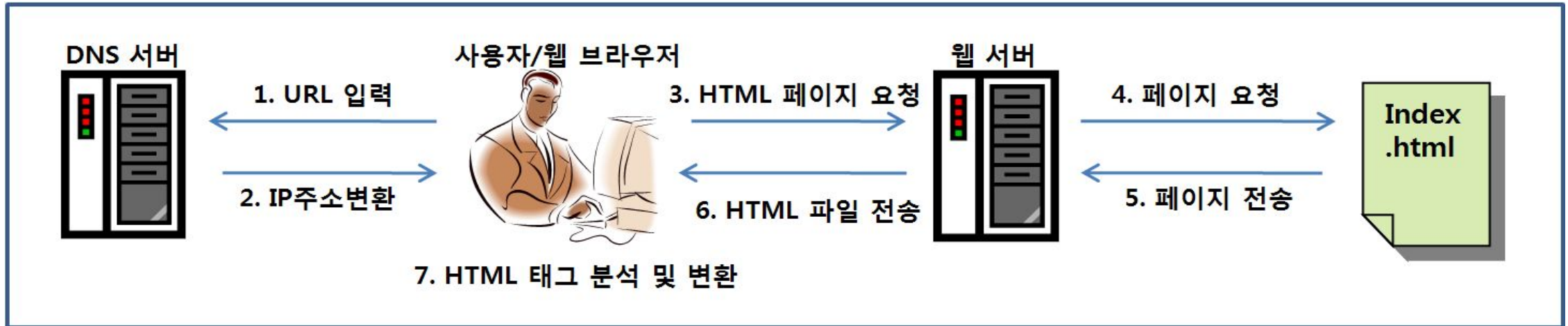
인터넷이란

- 웹의 개념이 나오기 전부터 만들어진 컴퓨터 연결 네트워크(1969년)
- 컴퓨터마다 고유한 주소(IP 주소, 113.198.80.208)를 부여하여 컴퓨터 구분
- 인터넷을 활용하는 응용 서비스 사례
 - 전자우편(e-mail)
 - 뉴스(news)
 - 파일 전송(ftp)
 - 채팅(Internet Relay Chat)
 - 메신저(Messenger)
 - P2P(Peer to Peer)
 - 스트리밍 서비스(Streaming Service)
 - 인터넷 전화기(Internet Phone)
 - 월드 와이드 웹(World Wide Web)

World Wide Web(WWW - 웹)

- **월드 와이드 웹, 웹(WWW)**
 - 인터넷을 활용하는 응용 서비스 중의 하나
 - 웹 서버와 웹 브라우저로 구성되는 **문서 (정보) 전달 및 공유 서비스**

웹의 동작 방식



웹의 구성요소 3가지

URL

(문서의 위치)

HTML

(문서)

HTTP

(규칙)

웹의 구성요소 3가지

URL – 웹 페이지의 주소



- 프로토콜 : HTTP, https, file, ftp, ssh 등
- 서버주소 : 웹 페이지를 가진 컴퓨터의 인터넷 주소, IP 주소 혹은 도메인 주소
- TCP/IP 포트 번호 : 서버가 브라우저로부터 접속을 기다리는 포트 번호.

default 값은 프로토콜마다 다르며, HTTP의 경우 80, ssh는 22

- 경로명 : 웹 서버 내 웹 페이지 파일의 폴더 경로
- 파일이름 : 웹 페이지의 HTML 파일 이름

웹의 구성요소 3가지

HTML (Hyper Text Markup Language)

하이퍼텍스트 마크업 언어(HyperText Markup Language, **HTML**)는 **XML**의 일종으로서 웹 페이지를 위한 마크업 언어이다.

HTML은 제목, 단락, 목록 등과 같은 본문을 위한 구조적 의미를 나타내는 것뿐만 아니라 링크, 인용과 그 밖의 항목으로 구조적 문서를 만들 수 있는 방법을 제공한다.

그리고 이미지와 객체를 내장하고 대화형 양식(**Form**)을 생성하는 데 사용될 수 있다.

HTML은 웹 페이지 콘텐츠 안의 꺾쇠 괄호<>에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성한다.

HTML은 웹 브라우저와 같은 HTML 처리 장치의 행동에 영향을 주는 자바스크립트와 문서의 외관과 배치(스타일)를 정의하는 CSS 같은 스크립트를 포함하거나 불러올 수 있다.

HTML은 웹 페이지를 위한 마크업 언어

⇒ 프로그래밍 언어는 아님.

(프로그래밍 언어는 연산이 가능해야함)

웹의 구성요소 3가지

HTML (Hyper Text Markup Language)

HyperText Markup Language의 약자로 월드와이드웹(World Wide Web)에서 사용하는 **마크업(Markup)** 언어

□ **웹페이지 구조 표현**을 위한 언어

마크업(Markup) 언어?

- 어떠한 문서 안의 **정보가 어떻게 구조화** 되었는가를 지정하는 언어
- 문서에 포함된 문장, 표, 그림 등과 같은 콘텐츠를 의미하는 것이 아니라, 이들의 **배치와 계층**을 나타내는 정보를 의미

웹의 구성요소 3가지

HTTP (HyperText Transfer Protocol)

HyperText를 전송하기 위한 통신 규칙



1~5 사이의 과정 : HTTP 세션

웹의 구성요소 3가지

URL

(문서의 위치)

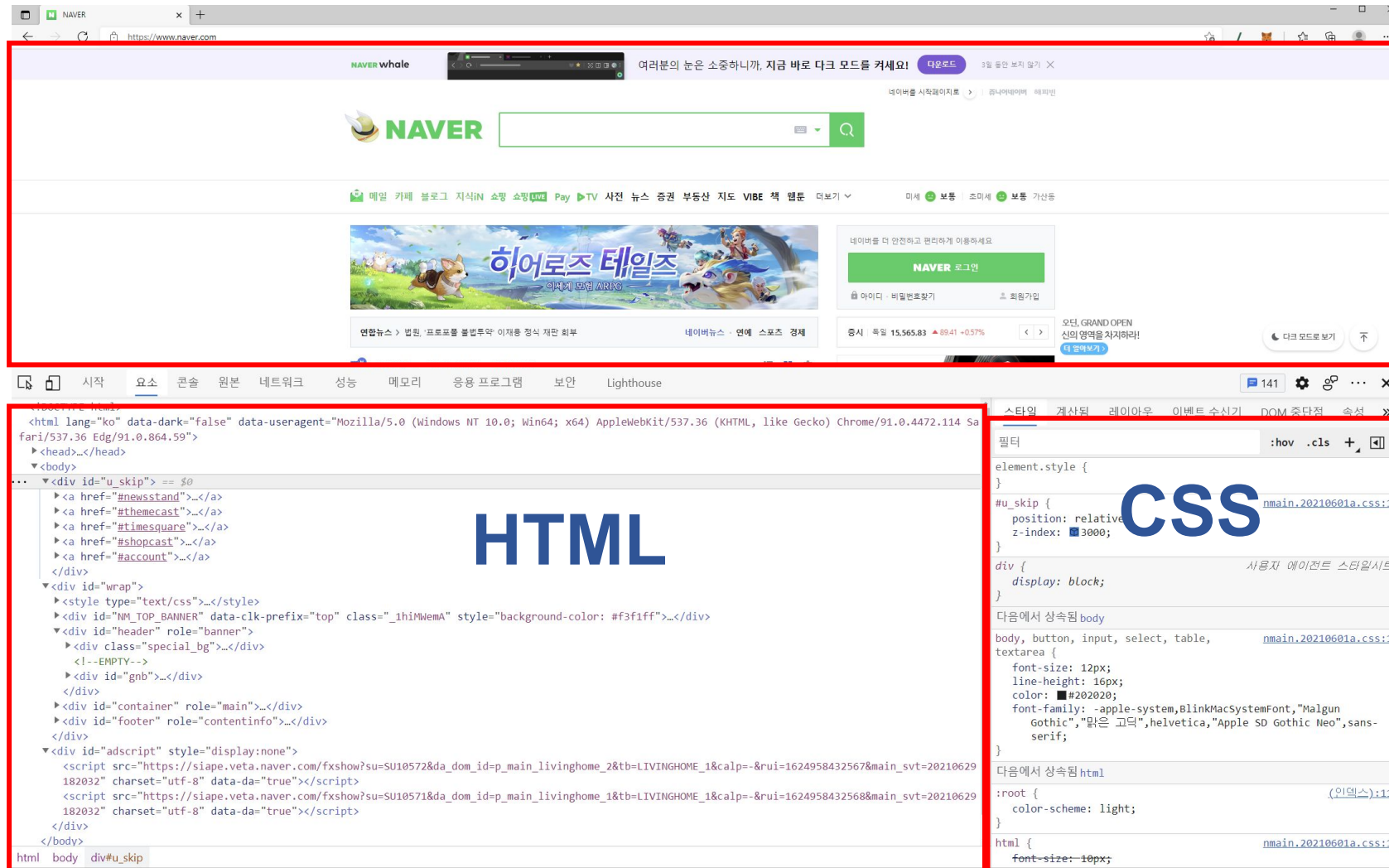
HTML

(문서)

HTTP

(규칙)

HTML과 CSS



HTML과 CSS

HTML 문서의 기본 형태

HEAD

- 문서에 대한 **메타 데이터**
- 문서 제목, 문자 집합, 스타일, 스크립트 및 기타 메타 정보를 정의

BODY

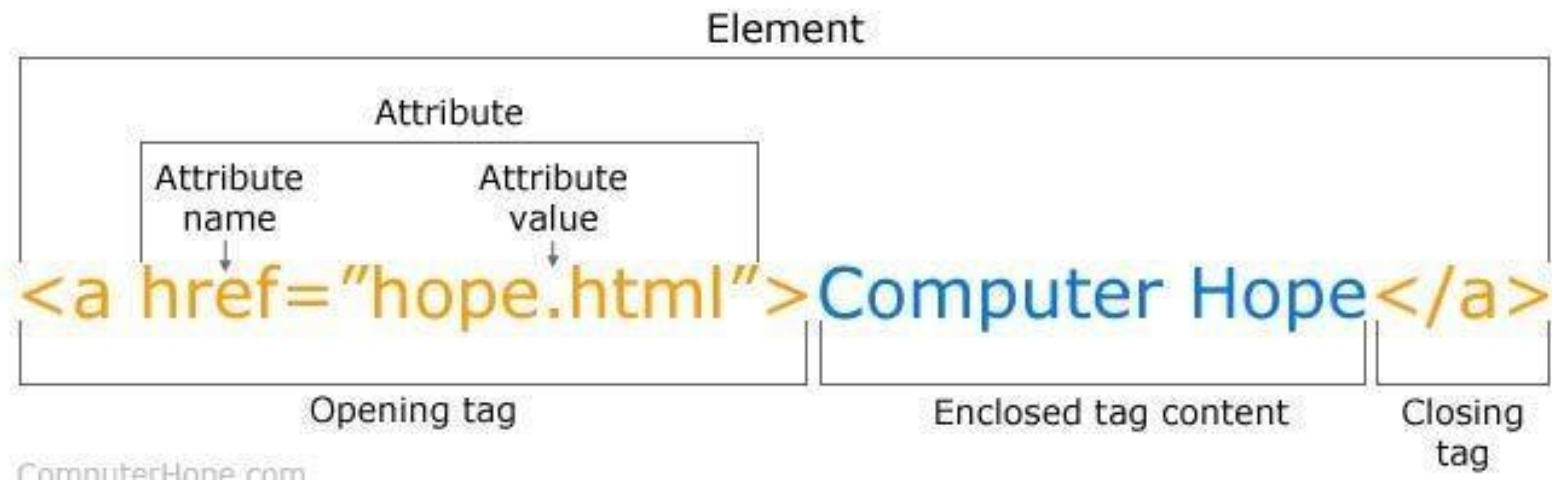
- `<body>` 태그는 문서의 몸통을 나타내는 태그입니다.
- `<head>` 태그와 대조적이며, 브라우저 화면에 보이는 것들이 주로 들어감. (주요 내용)

```
<!DOCTYPE html>
<!-- 주석문 -->
<html>
  <head>
    문서의 제목, CSS, 자바스크립트, 메타데이터
    정의
  </head>
  <body>
    문서의 본문 텍스트, 이미지, 동영상, 테이블,
    자바스크립트 코드 등
  </body>
</html>
```

HTML과 CSS

HTML 태그의 구성

- 여는 태그가 있으면 닫는 태그가 있다.
- 태그 하나로 이루어져 있는 경우도 있다.(열고 닫음이 한번에) `<hr/>` , `
` , ``
- 태그에는 속성을 부여할 수 있다.



HTML과 CSS

다양한 HTML 태그

<https://www.w3schools.com/tags/default.asp>

정말 많은 종류의 태그가 있습니다.

전부 외울 필요는 없습니다.

주요한 몇몇 태그만 아시면 바로 시작할 수 있습니다.

프로그래밍 공부는 우선 완성하고, 보완하시면서
나아가시는 것이 좋습니다.

□ 완벽주의를 내려 놓으십시오.

HTML과 CSS

주요한 HTML 태그(1)

<!-- -->	주석 태그	주석을 남기기 위한 태그
head	헤드	웹 페이지의 메타데이터를 담기 위한 태그
body	바디	본문을 담기 위한 태그
a	앵커 태그	anchor태그로서 페이지 이동을 담당(주요 속성:href)
div	Division 태그	기본이 되는 태그로서 구역을 나눔
h숫자	헤더	제목들을 나타내는 태그로서 숫자와 함께 사용. (<h1>, <h2>...)
p	문장	문장을 나타내는 태그
meta	메타	
title	제목	문서의 제목(탭에 표시되는 항목)

HTML과 CSS

주요한 HTML 태그(2)

form	폼 태그	사용자로부터 입력을 받기 위한 Form 태그(input을 묶어 서버로 전송)
input	인풋 태그	사용자로부터 입력을 받는 태그.(주요 속성: type)
img	이미지	이미지를 그리기 위한 태그 (주요 속성: src)
link	링크	주로 CSS 를 참조하기 위한 태그(주요 속성: href)
script	스크립트	주로 Javascript 를 참조하기 위한 태그(주요 속성: src)
ul	Unordered List	순서 없는 리스트
ol	Oredered List	순서 있는 리스트
li	List Item	ul혹은 이안에서 사용되며 ListItem
br	개행	
hr	수평선	

HTML과 CSS

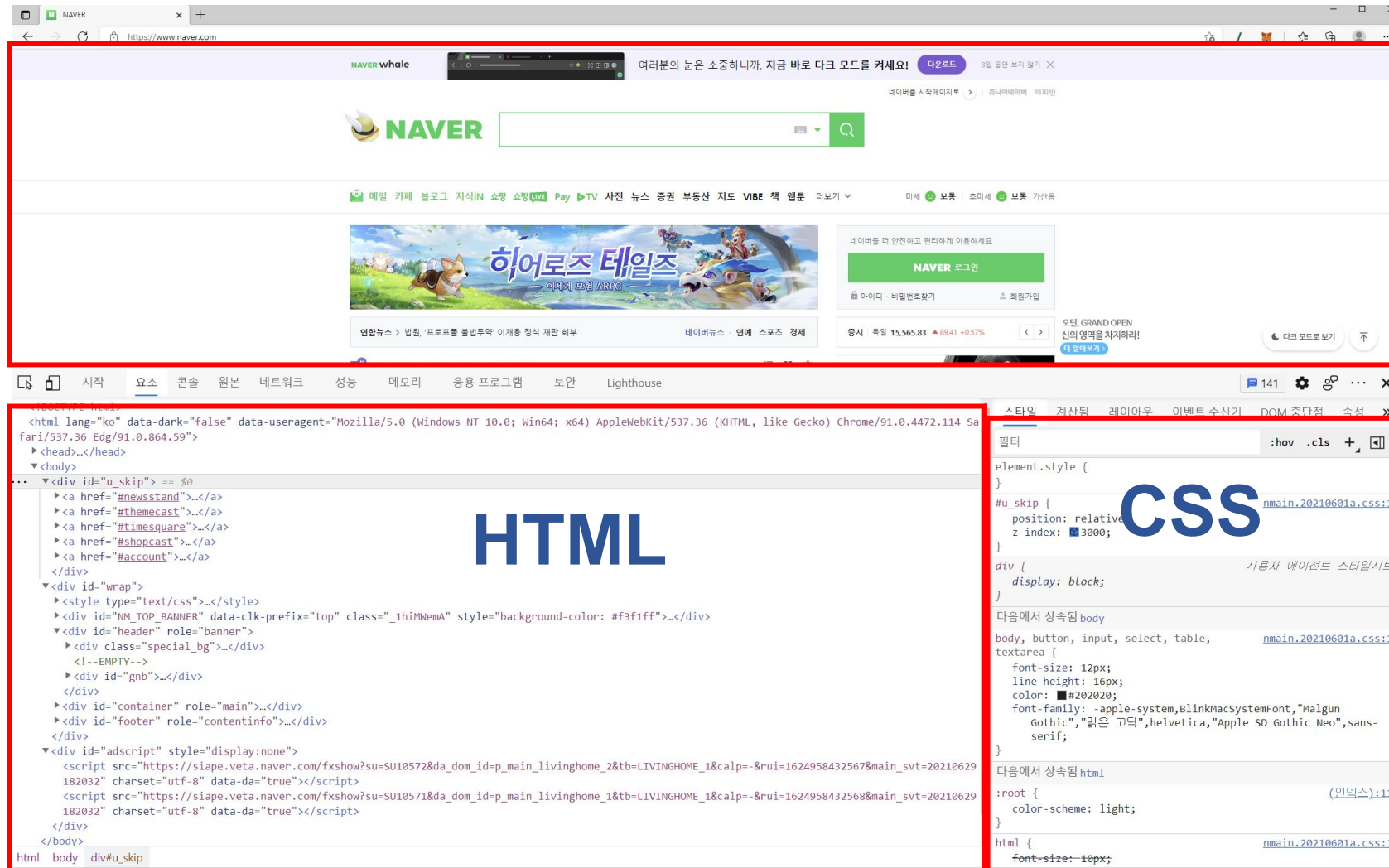
HTML의 Semantic Tag

시멘틱 태그(Semantic Tag)는 의미를 가지는 태그이다.

검색엔진이 해당 태그들을 읽고 의미를 해석해서, 관련 내용을 잘 추출할 수 있게 한다.
(웹 접근성을 위한 태그) - 기본적으로 **div**와 같다.

article	본문	일반적으로 내용을 저장
section	섹션	섹션을 구분지음
aside	사이드바	sidebar를 위함
nav	navigation	이동할 수 있는 탭이 위치
header	헤더	브랜드 로고 등이 위치
footer	Footer	하단에 위치시키는 Footer

HTML과 CSS



HTML과 CSS

CSS (Cascading Style Sheets)

CSS는 Markup 언어가 **실제 표시되는 방법을 기술**하는 언어로, XML에서 사용할 수 있다.
(HTML은 XML의 Subset이다!)

W3C의 표준이며, 레이아웃과 스타일을 정의할 때의 자유도가 높다.

마크업 언어(ex. HTML)가 웹사이트의 몸체를 담당한다면 **CSS는 옷과 액세서리 같은 꾸미는 역할을 담당**한다고 할 수 있다.

즉, HTML 구조는 그대로 두고 CSS 파일만 변경해도 전혀 다른 웹사이트처럼 꾸밀 수 있다.

CSS는 Markup언어가 실제 표시되는 방법을 정의!

HTML과 CSS

HTML

몸체(구조)

CSS

옷(꾸밈)

HTML과 CSS

다양한 CSS 속성

정말 많은 종류의 속성이 있습니다.

전부 외울 필요는 없습니다.

중요한 것 몇 개는 인지하시고, 나머지는 벤치마킹 혹은 찾아가면서 적용해보세요.

Reference

- W3(css 속성 정의) (<https://www.w3.org/>)
- W3Schools(<https://www.w3schools.com/>)
- MDN(<https://developer.mozilla.org/en-US/docs/Web/CSS>)

HTML과 CSS

주요한 CSS 속성

margin (top right bottom left) 영역 밖 여백

padding (top right bottom left) 영역 안 여백

width (min, max) 넓이

height (min, max) 높이

background 배경

float Float

display block, inline-block, inline, flex, none

overflow auto, scroll, hidden, visible

position absolute vs relative

border (color size style radius) 테두리

font (size, weight, family) 글꼴 관련

color 색상

vertical-align (display: inline-block 일 때)

text-align 이미지나, 텍스트 가운데 정렬

box-sizing content-box : 콘텐츠 기준
border-box : 테두리를 기준

핵심요

- 웹의 세가지: URL, HTTP, HTML
- HTML은 문서의 내용 및 골격
- CSS는 문서의 외향

연습문제

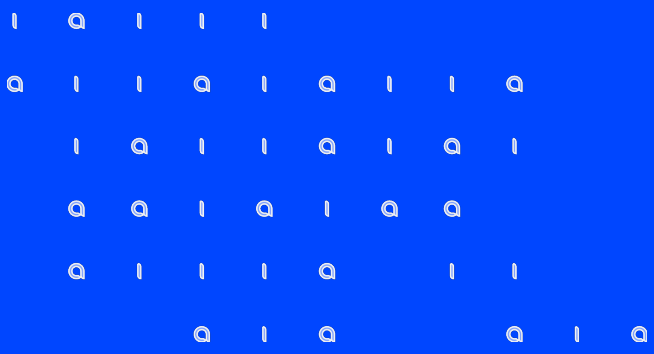
!

1) 인터넷은 웹이다. (O/X)

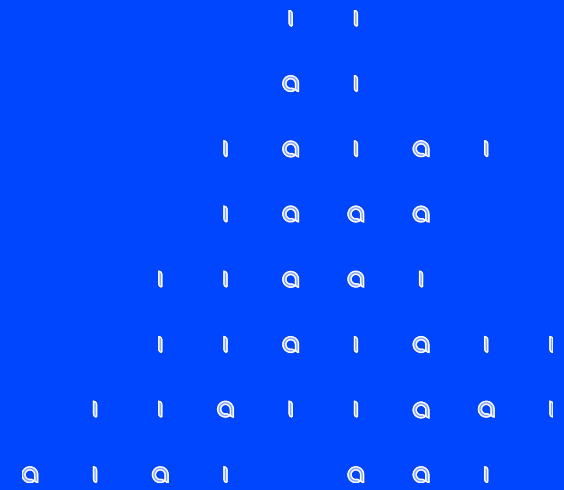
1) 구름 이미지를 **HTML**에 삽입하고, 구름 이미지를 클릭하면 **NAVER**로 이동하도록 하이퍼 링크를 만드세요

2) (심화) **NAVER**로 이동하는 것을 새 탭으로 열리도록 하세요.
=> (배운적은 없으나, 구글링 혹은 **W3C**를 확인하여 해결하세요.)





2. 헛갈리기 쉬운 CSS

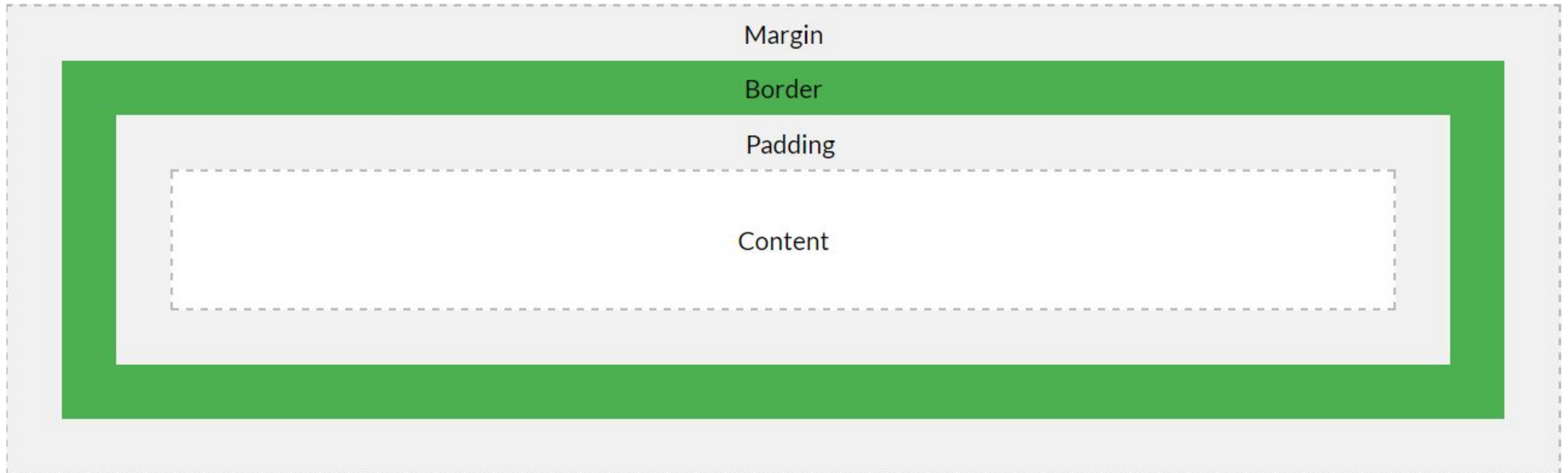


학습목

표

- ▶ Margin과 Padding의 차이를 알기.
- ▶ Flex 속성과 하위 속성
- ▶ display와 float 속성
- ▶ Position 속성
- ▶ Size에 대한 다양한 단위

1. Margin vs padding



- Content** - 컨텐츠 영역
- Padding** - border 안쪽 여백
- Border** - 테두리
- Margin** - border 바깥 여백

2. 크기를 표현하는 다양한 단위

[fontsize]

px vs em vs rem

px: 절대적인 크기 (15px)

em: 상위요소에 대비하여 상대적인 크기 (1.5em: 상위태그의 font-size 10px이면, 15px)

rem: 문서전체(HTML태그)에 대비하여 상대적인 크기 (1.5rem: HTML의 font-size 10px이면, 15px)

2. 크기를 표현하는 다양한 단위

[width 또는 height]

px vs vw vs vh vs %

px: 절대적인 크기 (100px)

vw: viewport에서 몇%를 차지할 것이냐?(100vw: 현 스크린 너비 전체)

vh: viewport에서 height를 얼마나 차지할 것이냐?(100vh: 현 스크린 높이 전체)

%: 상위 태그에서 width혹은 height를 얼마나 차지할 것이냐?(100%: 상위태그를 꽉 채우겠다.)

2. 크기를 표현하는 다양한 단위

[fontsize]

px vs em vs rem

[width 또는 height]

px vs vw vs vh vs %

em과 rem도 width에 사용될 수 있지만, font-size 속성에 기반한다는 것을 알고 있으면 됨!
외우려 하지 말고 필요할 때 사용하다 보면 외워짐!

3. 색상표현

컬러 키워드: red, violet 등

16진법 (헥사코드) #32123F

RGB () / RGBA()<투명도 추가>

4. Position (요소의 위치를 정하기)

`position: static | absolute | fixed | relative | sticky | initial | inherit;`

static	Normal-flow 에 따라 배치되며 offset 값이 적용되지 않는다. (기본값)
absolute	부모 요소의 위치를 기준으로 offset 에 따라 배치된다. 부모가 position 값(static 제외)을 가지면 offset 값의 시작점이 된다. * 부모의 <i>position</i> 값이 static 인 경우 조상의 <i>position</i> 값이 static 이 아닐 때까지 거슬러 올라가 기준으로 삼습니다.
fixed	뷰포트(브라우저의 창)를 기준으로 offset 에 따라 배치된다. 즉, 화면 스크롤에 관계없이 항상 화면의 정해진 위치에 정보가 나타난다. 부모의 위치에 영향을 받지 않는다.
relative	자신이 원래 있어야 할 위치를 기준으로 offset 에 따라 배치된다. 부모의 position 속성에 영향을 받지 않는다.

4. Position (요소의 위치를 정하기)

offset!

```
1 top: 50%;  
2 left: 10px;  
3 bottom: -10px;  
4 right: auto;
```

z-index

: 무엇이 먼저 보일지 표시 - 숫자가 클수록 먼저 보임

5. 비교적 최근에 나온 flex

비교적 최근에 나온 flex

스크린 사이즈가 다양화 된다.

⇒ 반응형 웹에 대한 수요가 늘어난다.

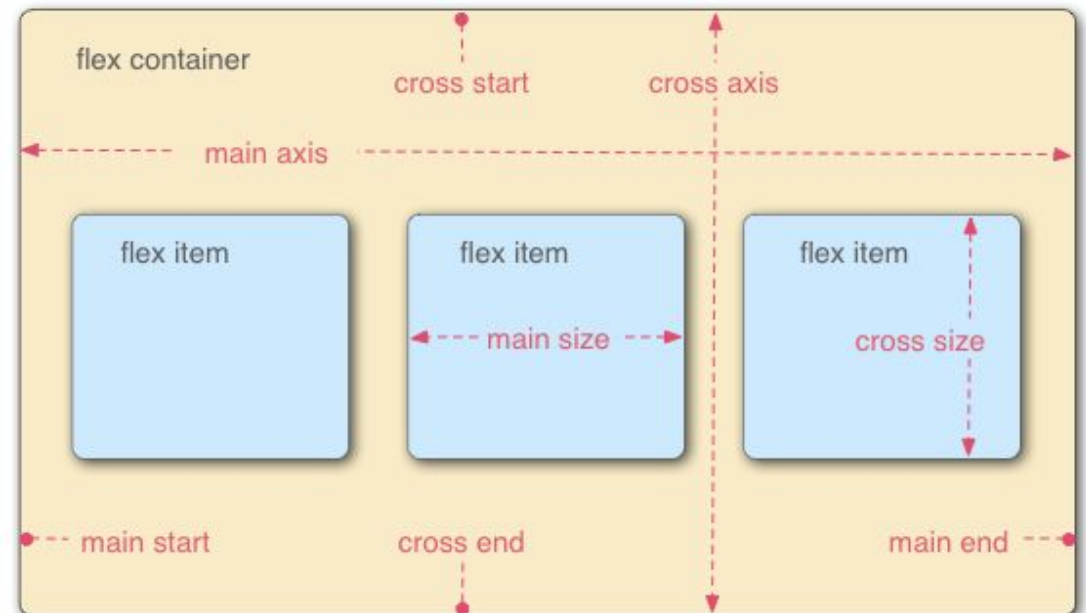
⇒ 이를 위한 **CSS**속성에 대한 요구가 늘어남
(기존엔 %를 활용하여 반응형 구축)

⇒ **display:flex** 속성이 나옴!

최근 반응형 웹은 거의 **flex**속성을 사용.

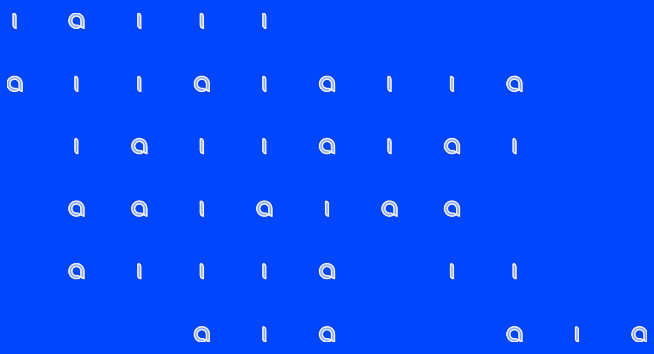
⇒ flex속성 학습은 다음 링크 참조

<https://flexboxfroggy.com/>

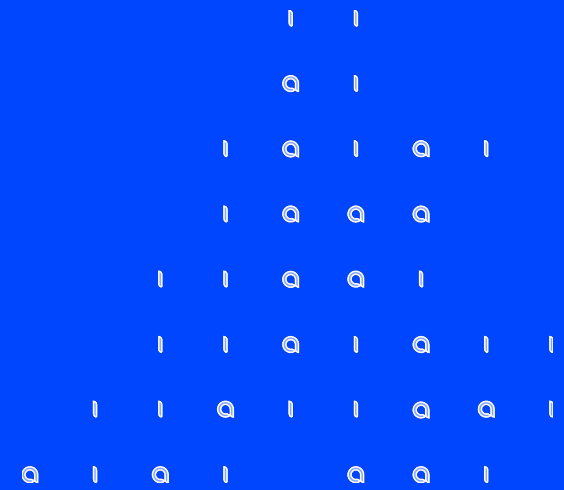


핵심요

- Margin은 영역 밖 여백, Padding은 영역 안 여백
- Px은 절대적인 크기
- em: 상위 태그의 font-size에 대비한 크기, rem: root(html)태그의 font-size에 대비한 크기
- vw/vh 는 각각 viewport(사용 device)의 사이즈에 대비한 크기, %는 상위 요소의 크기에 대비한 크기
- Position은 어떻게 보일까에 대한 속성
relative, static은 다른 element에 대비한 상대적인 속성
absolute는 다른 element에 영향 받지 않는 절대적인 속성
fixed는 화면상에서 항상 해당위치에 고정
- Flex 속성은 element들의 배치에 관한 속성, flex-direction에 따라 primary axis와 secondary axis가 바뀜.
그에 따라 요소들을 나란히 배치할 수 있음.



3. 반응형 웹과 미디어쿼리



학습목

표

- ▶ 다양한 스크린 사이즈와 반응형 웹
- ▶ Media Query란
- ▶ 부트스트랩 사용하기

교과목 교육과정

구분

교과목	세부 주제	학습목표	학습내용	학습방법	학습시간
실무형 웹페이지 실습	HTML과 CSS	<ol style="list-style-type: none"> 1. 전반적인 웹의 동작 방식을 파악할 수 있다. 2. HTML과 CSS를 정의할 수 있다. 3. HTML의 다양한 태그들을 사용할 수 있다. 4. 원하는 형태의 CSS 정의할 수 있다. 	<ol style="list-style-type: none"> 1. 웹의 동작방식 2. HTML과 CSS에 대한 정의 3. 대표적인 HTML 태그 4. 대표적인 CSS 태그 	이론	4
	헛갈리기 쉬운 CSS	<ol style="list-style-type: none"> 1. Margin과 Padding의 차이를 알기. 2. 크기를 표현하는 다양한 단위 3. Position 속성 4. Flex 속성과 하위 속성 	<ol style="list-style-type: none"> 1. 헛갈리기 쉬운 다양한 CSS 속성을 짚고 넘어가기 	이론 및 실습	4
	반응형 웹과 미디어쿼리 (부트스트랩)	<ol style="list-style-type: none"> 1. 반응형 웹이 무엇인지 안다. 2. Media query에 대해 이해한다. 3. Breakpoint를 이해한다. 4. Grid System을 사용할 수 있다. 	<ol style="list-style-type: none"> 1. 미디어 쿼리 작성하기 2. 부트스트랩 사용하기 3. 반응형 웹 구축하기 	이론 및 실습	4
	실전 웹페이지 작성하기	<ol style="list-style-type: none"> 1. 스스로 웹사이트 구축을 위한 HTML/CSS를 사용할 수 있다. 2. 필요한 리소스를 온라인상에서 찾아 사용할 수 있다. 		실습	26
총계					4

1. 다양한 스크린 사이즈와 반응형 웹

스마트폰이 나오면서..

현대사회는 다양한 **Device**를 통해 웹을 접할 수 있다.

- 스마트폰 같은 작은 사이즈의 모바일기기
- tablet pc부터
- 다양한 컴퓨터 모니터 화면
- 집에 있는 TV

□ 기존에 개발된 스크린 사이즈(컴퓨터 모니터)로는 다양한 **Device**에 대응을 하지 못한다.
(화면이 깨지는 등, 사용자에게 좋은 **UX**를 주지 못한다.)

어떻게 해결할 수 있는지 생각해보자.

(엔지니어링의 시작은 아이디어)

1. 다양한 스크린 사이즈와 반응형 웹

다양한 screen size에 대응하기

[방법1]

- 모바일용 웹페이지, **table**용 웹페이지, **Desktop** 웹페이지를 각각 따로 구성한다.
- 사용자로 하여금 사용자의 디바이스에 따라 다른 페이지를 보여준다.

1. 다양한 스크린 사이즈와 반응형 웹

다양한 screen size에 대응하기

[방법1]

- 모바일용 웹페이지, **table**용 웹페이지, **Desktop** 웹페이지를 각각 따로 구성한다.
- 사용자로 하여금 사용자의 디바이스에 따라 다른 페이지를 보여준다.

대응은 되나, 작업해야 할 것들이 늘어난다.
□비용이 증가한다.

1. 다양한 스크린 사이즈와 반응형 웹

다양한 screen size에 대응하기

[방법1]

- 모바일용 웹페이지, **table**용 웹페이지, **Desktop** 웹페이지를 각각 따로 구성한다.
- 사용자로 하여금 사용자의 디바이스에 따라 다른 페이지를 보여준다.

대응은 되나, 작업해야 할 것들이 늘어난다.
□비용이 증가한다.

[방법2]

- 하나의 페이지를 구성하되, **Screen Size**에 따라 다르게 **Styling**하자.

비용이 줄어든다.

Styling을 담당하는 언어는?

1. 다양한 스크린 사이즈와 반응형 웹

다양한 screen size에 대응하기

[방법1]

- 모바일용 웹페이지, **table**용 웹페이지, **Desktop** 웹페이지를 각각 따로 구성한다.
- 사용자로 하여금 사용자의 디바이스에 따라 다른 페이지를 보여준다.

대응은 되나, 작업해야 할 것들이 늘어난다.
□비용이 증가한다.

[방법2]

- 하나의 페이지를 구성하되, **Screen Size**에 따라 다르게 **Styling**하자.

비용이 줄어든다.

Styling을 담당하는 언어는? **CSS**

CSS 문법에 대한 확장을 요구
media query에 탄생

1. 다양한 스크린 사이즈와 반응형 웹

다양한 screen size에 대응하기

[방법1]

- 모바일용 웹페이지, **table**용 웹페이지, **Desktop** 웹페이지를 각각 따로 구성한다.
- 사용자로 하여금 사용자의 디바이스에 따라 다른 페이지를 보여준다.

적응형 웹
(naver 등)

user-agent에 따라 subdomain을 사용하거나 경로를 다르게 routing

[방법2]

- 하나의 페이지를 구성하되, **Screen Size**에 따라 다르게 **Styling**하자.

반응형 웹
(대부분의 웹)

1. 다양한 스크린 사이즈와 반응형 웹

반응형 웹이란

반응형 웹 디자인(responsive web design, RWD)이란 하나의 웹사이트에서 **PC**, 스마트폰, 태블릿 **PC** 등 접속하는 디스플레이의 종류에 따라 화면의 크기가 자동으로 변하도록 만든 웹페이지 접근 기법을 말한다.

웹사이트를 **PC용**과 **모바일용**으로 각각 **별개로** 제작하지 않고, 하나의 **공용 웹사이트**를 만들어 다양한 디바이스에 대응할 수 있다.

PC용 URL과 **모바일용 URL**이 동일하기 때문에 검색 포털 등 광고를 통한 사용자 접속을 효율적으로 관리할 수 있다. 또한 웹 페이지 내용을 수정할 경우, 하나의 페이지만 수정하면 **PC**와 **모바일** 등 다양한 디바이스에서 동일하게 반영된다. 반대로는 디바이스별로 별도의 웹사이트를 제작하는 적응형 웹 (adaptive web)이다.

1. 다양한 스크린 사이즈와 반응형 웹

반응형 웹이란

반응형 웹 디자인(responsive web design, RWD)이란 하나의 웹사이트에서 **PC**, 스마트폰, 태블릿 **PC** 등 접속하는 디스플레이의 종류에 따라 화면의 크기가 자동으로 변하도록 만든 웹페이지 접근 기법을 말한다.

웹사이트를 **PC**용과 모바일용으로 각각 **별개로** 제작하지 **않고**, 하나의 **공용** 웹사이트를 만들어 다양한 디바이스에 대응할 수 있다.

PC용 URL과 모바일용 **URL**이 동일하기 때문에 검색 포털 등 광고를 통한 사용자 접속을 효율적으로 관리할 수 있다. 또한 웹 페이지 내용을 수정할 경우, 하나의 페이지만 수정하면 **PC**와 모바일 등 다양한 디바이스에서 **동일하게** 반영된다. 반대로는 디바이스별로 별도의 웹사이트를 제작하는 적응형 웹 (adaptive web)이다.

2. media query

Media query: device에 따라 다른 CSS 적용

Media Query 기본 구조

```
@media media-type and (media-feature-rule) {  
  스타일  
}
```

Media Type(장치)

- all : 모든 장치
- print : 인쇄 결과물 및 출력 미리보기 화면에 표시 중인 문서
- screen : 스크린 화면
- speech: 음성 합성장치 (잘 사용 X)

2. media query

다양한 조건(1) – 너비 기준

```
@media screen (max-width: 800px) {  
  .small-tomato {  
    background-color: tomato;  
  }  
}
```

```
@media screen (min-width: 800px) {  
  .large-tomato {  
    color: tomato;  
  }  
}
```

다양한 조건(2) - 방향성 기준

```
@media screen (orientation: landscape) {  
  /*가로 방향 - 가로가 더 긴 것  
   (pc 모니터 포함) */  
  .small-tomato {  
    background-color: tomato;  
  }  
}
```

```
@media screen (orientation: portrait) {  
  /*세로 방향 - 세로가 더 긴 것 */  
  .small-tomato {  
    background-color: tomato;  
  }  
}
```

2. media query

다양한 조건(3) – 논리 연산자 지원

```
@media screen (max-width: 800px) and (min-width: 574px) {  
  .small-tomato {  
    background-color: tomato;  
  }  
}
```

```
@media not (screen (max-width: 800px) or (min-width: 574px)) {  
  // not 과 or 은 잘 안쓰게 됩니다. 있다고만 알아주세요  
  .small-tomato {  
    background-color: tomato;  
  }  
}
```

2. media query

연습문제

div를 하나 만들고, background-color를 각각 조건에 맞게 변경되도록 하시오.

- width가 376px보다 작으면 black
- width가 376px보다 크고 600px보다 작으면 blue
- width가 600px보다 크고 960px보다 작으면 tomato
- width가 960px보다 크고 1280px보다 작으면 yellow
- width가 1280px 보다 크면 gold 로 하세요.

3. 부트스트랩 사용하기

CSS 라이브러리

모두 다 외우고 사용하기도 힘들고
디자인을 어디서부터 시작할까 막막하다.

□ CSS 라이브러리가 존재

3. 부트스트랩 사용하기

CSS 스타일 규칙 + HTML 태그와 클래스

아름다운 웹 사이트를 쉽게 만들 수 있게 도와주는 유명한 **CSS** 라이브러리 존재

몇몇은 자바스크립트 위젯을 포함,
개발자들은 대부분 **CSS** 라이브러리와 함께 다른 **JS** 라이브러리를 사용.

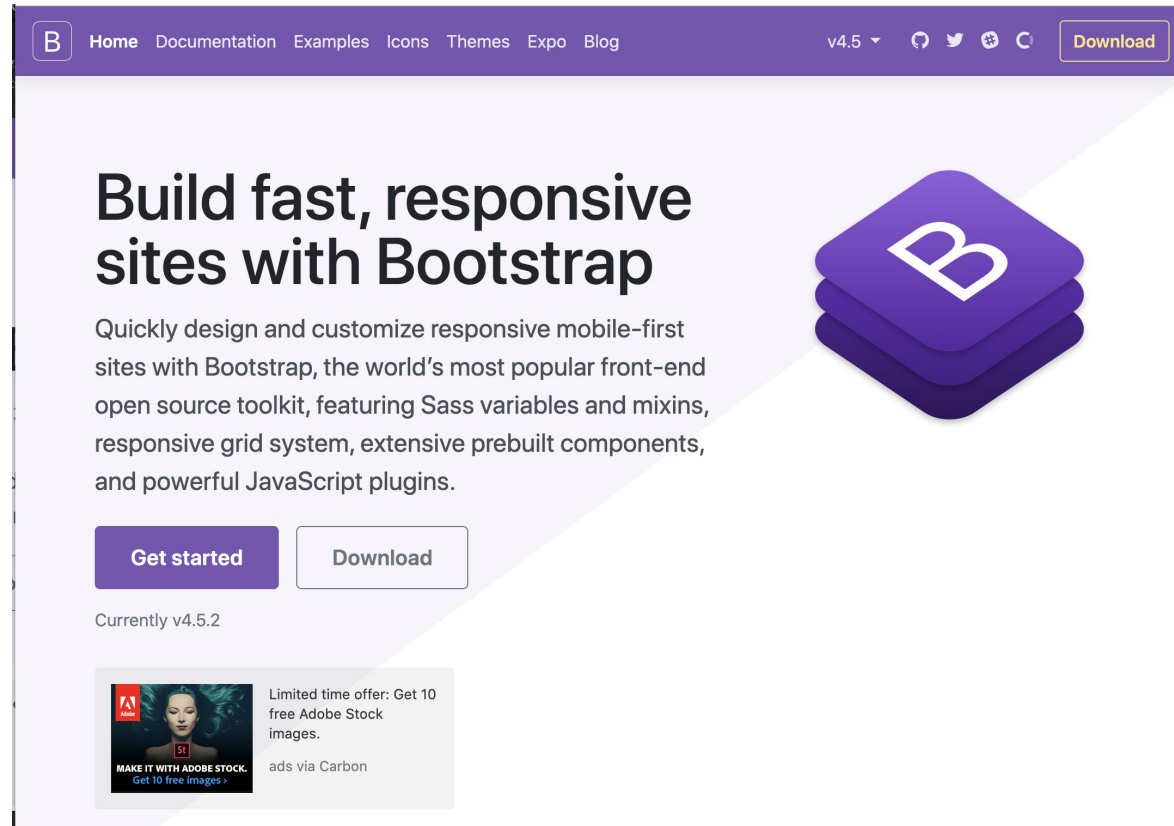
부트스트랩 - <https://getbootstrap.com/>

파운데이션 - <https://get.foundation/>

Semantic-UI - <https://semantic-ui.com/>

Pure CSS - <https://purecss.io/>

3. 부트스트랩 사용하기



[부트스트랩]

<https://getbootstrap.com/>

3. 부트스트랩 사용하기

BreakPoint와 그리드시스템

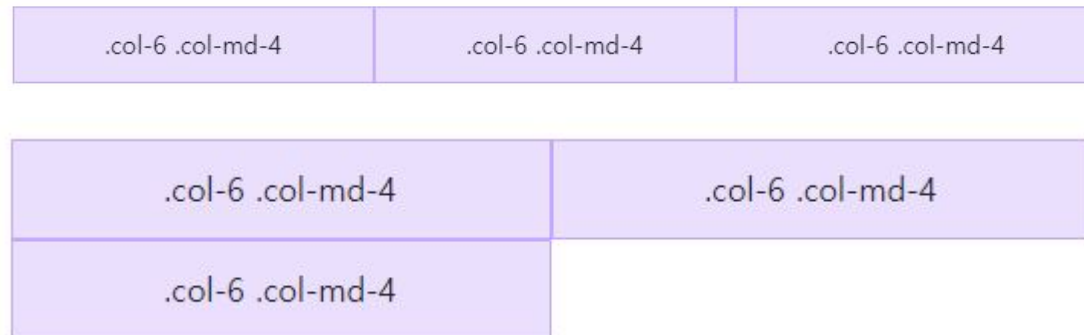
부트스트랩에서 다양한 컴포넌트들은 잘 가져다 사용하시면 됩니다.
CSS 라이브러리마다 디자인의 형태가 다 다르기 때문입니다.

하지만 대부분의 CSS라이브러리(UI Kit)에서 공통적인 것이 하나 있습니다.
레이아웃을 위한 **그리드 시스템 (Grid System)** 과 이를 위한 **BreakPoint**입니다.

3. 부트스트랩 사용하기

Grid System

```
<div class="container text-center">
  <div class="row">
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>
</div>
```

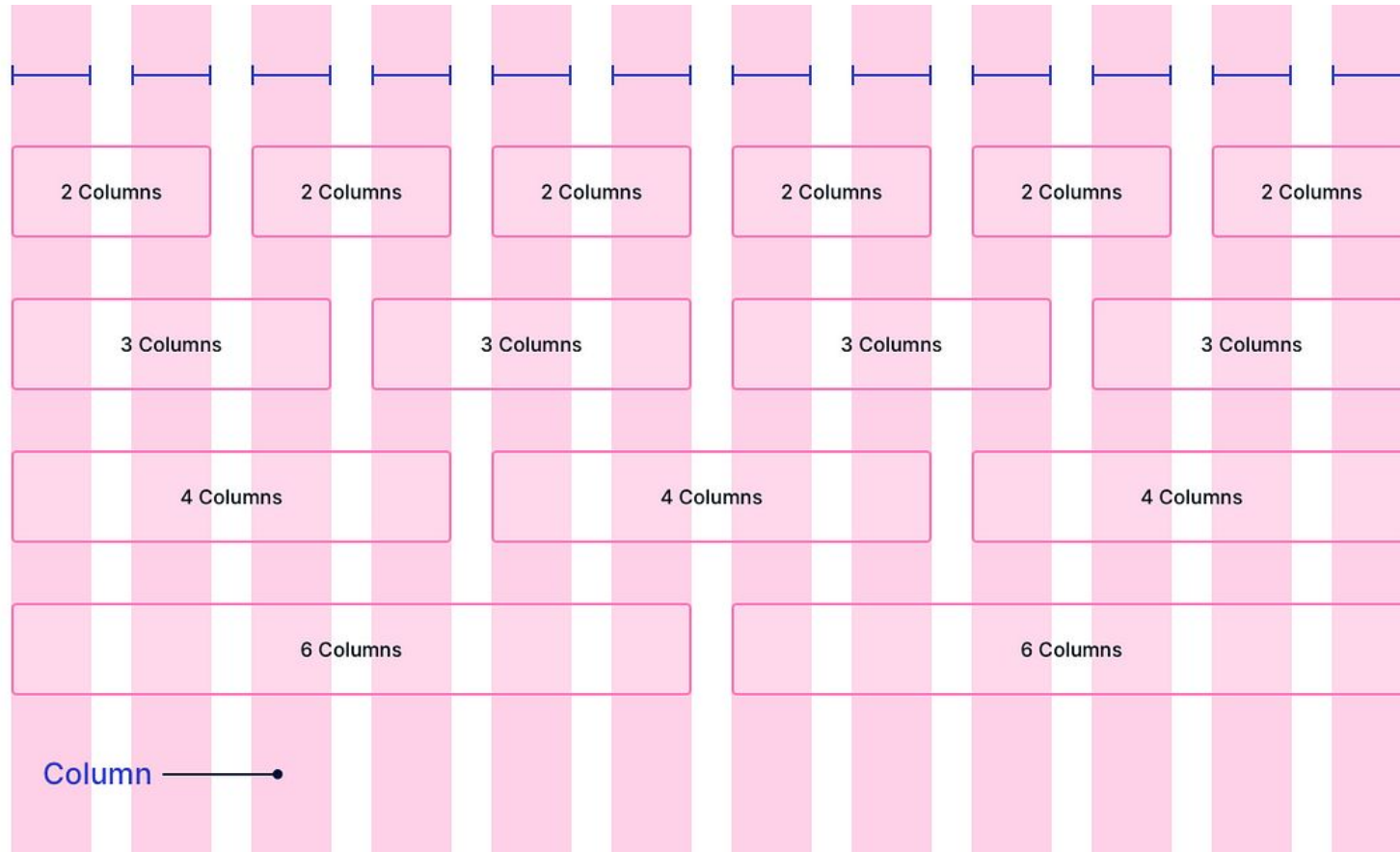


BreakPoint

Breakpoint	Class infix	Dimensions
Extra small	<i>None</i>	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

3. 부트스트랩 사용하기

Grid System은 전체를 12등분



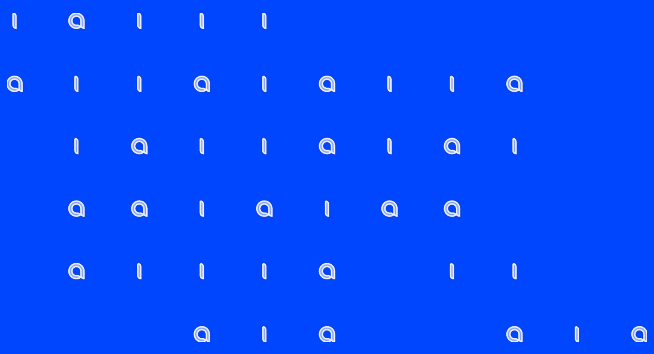
3. 부트스트랩 사용하기

Grid System은 전체를 12등분

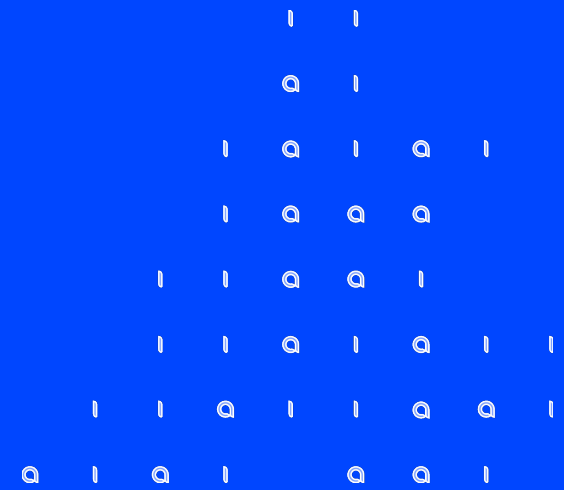
	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container max-width	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
# of columns	12					
Gutter width	1.5rem (.75rem on left and right)					
Custom gutters	Yes					
Nestable	Yes					
Column ordering	Yes					

핵심요

- 반응형 웹은 하나의 HTML/CSS로 다양한 Screen에 대응하는 웹
- CSS의 MediaQuery는 사이즈의 max와 min 값을 적용함으로써 반응형 웹을 구현
- Grid System은 너비를 12등분



4. 실전 웹페이지 작성하기



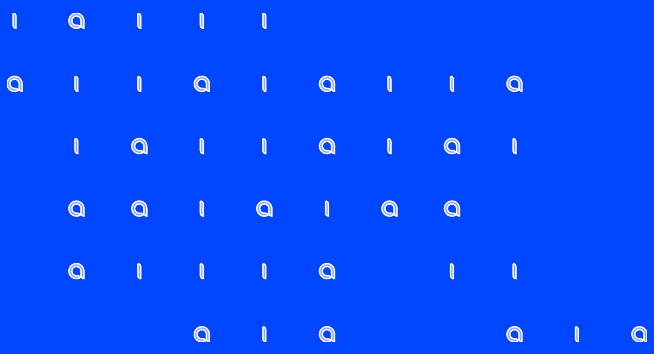
실전 웹페이지 작성하기

부트스트랩 등 **CSS**라이브러리를 이용하여, 다양한 웹페이지를 작성합니다.

주제는 자유롭게 선정하되 아래 페이지는 반드시 포함되어야 합니다.

예) **OTT, Commerce** 등

1. 홈화면
2. 회원가입
3. 로그인
4. 리스트
5. 상세페이지
6. 댓글 창
7. 워드클라우드



THANK YOU.

