

=

# Do it!

## HTML5 + CSS3 웹 표준의 정석 (2017년판)

(요약본 d0.7)



## 목차

0	훈련 준비	10
0.1	웹표준1(HTML5+CSS3) 학습 진도표	10
1	HTML 기본 다지기	11
1.1	HTML과의 첫 만남	11
(1)	HTML이란 무엇일까?	11
(2)	HTML 소스 엿보기	11
(3)	웹 표준이란 무엇일까?	11
(4)	왜 HTML5와 CSS3를 공부해야 할까?	11
1.2	웹 브라우저와 웹 편집기	12
(1)	웹 브라우저 종류와 HTML5 지원 정도	12
[실습]	크롬 설치하고 기본 웹 브라우저로 정하기	12
(2)	웹 편집기의 종류	13
(3)	통합개발환경(IDE) - 이클립스 소개	13
[실습]	이클립스 설치하고 살펴보기	13
1.3	HTML 기본 문서 구조	14
(1)	태그, 이건 꼭 알아두세요!	14
[실습]	간단한 HTML 문서 만들기	14
(2)	HTML 문서 기본 구조 살펴보기	15
(3)	<!doctype> - 문서 유형을 지정하는 선언문	15
(4)	<html> 태그 - 웹 문서 시작을 알리는 태그	15
(5)	<head> 태그 - 브라우저에게 정보를 주는 태그	15
(6)	<body> 태그 - 실제 브라우저에 표시될 내용	16
(7)	웹 문서에서 특수 문자 및 특수 기호 사용하기	16
1.4	웹 문서 만들고 업로드하기	16
(1)	호스팅 서버 준비하기	16
[실습]	닷홈에 무료 호스팅 서비스 신청하기	16
(2)	FTP 프로그램 설치하기	16
[실습]	FTP 프로그램 FileZilla 설치하고 서버에 접속하기	17
[실습]	내가 만든 문서 웹 서버에 올리고 확인하기	17
2	텍스트 관련 태그들	18
2.1	텍스트를 덩어리로 묶어 주는 태그	18
(1)	<h> 태그 - 제목 표시하기	18
(2)	<p> 태그 - 단락 만들기	18
(3)	  태그 - 줄 바꾸기	18
(4)	<hr> 태그 - 분위기 전환을 위한 수평 줄 넣기	18
(5)	<blockquote> 태그 - 인용문 넣기	18
(6)	<pre> 태그 - 입력하는 그대로 화면에 표시하기	19
2.2	텍스트를 한 줄로 표시하는 태그	19
(1)	<strong> 태그, <b> 태그 - 굵게 표시하기	19
(2)	<em> 태그, <i> 태그 - 이탤릭체로 표시하기	19
(3)	<q> 태그 - 인용 내용 표시하기	20
(4)	<mark> 태그 - 형광펜 효과 내기	20
(5)	<span> 태그 - 줄바꿈 없이 영역 묶기	20
(6)	<ruby> 태그 - 동아시아 글자 표시하기	21
(7)	기타 텍스트 관련 태그들	21
2.3	목록을 만드는 태그들	21
(1)	<ul> 태그, <li> 태그 - 순서 없는 목록 만들기	21
(2)	<ol> 태그, <li> 태그 - 순서 목록 만들기	22
(3)	<dl>, <dt>, <dd> 태그 - 설명 목록 만들기	22
[실습]	온라인 프로필 - 텍스트 입력하기	23

2.4 표를 만드는 태그	23
(1) <table>, <tr>, <td>, <th> 태그 - 기본적인 표 만들기	23
(2) colspan, rowspan 속성 - 행 또는 열 합치기	25
(2) <caption> 태그, <figcaption> 태그 - 표에 제목 붙이기	25
(3) <thead>, <tbody>, <tfoot> 태그 - 표 구조 정의하기	27
(4) <col>, <colgroup> 태그 - 여러 열 묶어 스타일 지정하기	28
[실습] 온라인 프로필 - 표 추가하기	29
[과제] 연습문제	29
3 이미지와 하이퍼링크	31
3.1. 이미지	31
(1) 웹에서 사용하는 이미지 형식	31
(2) <img> 태그 - 이미지 삽입하기	31
(3) alt 속성 - 이미지를 설명해 주는 대체 텍스트	31
(4) width, height 속성 - 이미지 크기 조정하기	31
(5) <figure>, <figcaption> 태그 - 이미지에 설명 글 붙이기	32
[실습] 온라인 프로필 - 이미지 삽입하기	32
3.2. 링크 만들기	32
(1) <a> 태그, href 속성 - 링크 만들기	32
(2) target 속성 - 새 탭에서 링크 열기	33
(3) 한 페이지 안에서 점프하는 앵커 만들기	33
[실습] 온라인 프로필 - 링크 추가하기	34
[실습] 인터넷에 내 온라인 프로필 올리기	34
3.3. SVG 이미지	34
(1) SVG 파일 형식이란?	34
(2) SVG 이미지 삽입하기	35
(3) SVG 파일을 지원하지 않는 브라우저를 위해	35
[과제] 연습문제	36
4 폼과 관련된 태그들	37
4.1 폼 만들기	37
(1) 웹에서 자주 만나는 폼	37
(2) <form> 태그 - 폼 만들기	37
(3) <label> 태그 - 폼 요소에 레이블 붙이기	37
(4) <fieldset>, <legend> 태그 - 폼 요소 그룹으로 묶기	38
4.2 사용자가 정보를 입력하는 <input> 태그	39
(1) <input> 태그 살펴보기	39
(2) <input> 태그 - 입력 항목 만들기	39
(3) type="hidden" - 히든 필드 만들기	39
(4) type="text" - 텍스트 필드 만들기	39
(5) type="password" - 비밀번호 입력란 만들기	39
(6) type="search", type="url", type="email", type="tel" - 분화된 테스트 필드	40
(7) type="number" - 숫자 입력하기	41
(8) type="range" - 슬라이드 막대로 숫자 지정하기	41
(9) type="radio", type="checkbox" - 라디오 버튼과 체크박스 넣기	42
(10) type="color" - 색상 선택 상자 표시하기	43
(11) type="date", type="month", type="week" - 날짜 표시하기	44
(12) type="time", type="datetime", type="datetime-local" - 시간 지정하기	44
(13) type="submit", type="reset" - 서버 전송, 리셋 버튼 넣기	45
(14) type="image" - 이미지 버튼 넣기	45
(15) type="button" - 버튼 넣기	46
(16) type="file" - 파일 첨부하기	46
4.3 <input> 태그의 다양한 속성	46
(1) autofocus 속성 - 입력 커서 표시하기	46

(2) placeholder 속성 - 힌트 표시하기.....	46
(3) readonly 속성 - 읽기 전용 필드 만들기.....	46
(4) required 속성 - 필수 필드 지정하기.....	46
(5) min, max, step 속성 .....	46
(6) size, minlength, maxLength 속성 - 길이, 최소 길이, 최대 길이 속성 .....	48
[실습] 상품 주문서 작성하기 .....	49
4.4 여러 데이터 나열에서 보여주기.....	49
(1) <select>, <optgroup>, <option> 태그 - 드롭다운 목록 만들기.....	49
(2) <datalist>, <option> 태그.....	50
(3) <textarea> 태그 - 여러 줄 입력하는 텍스트 영역 만들기.....	52
4.5 기타 다양한 폼 요소들 .....	53
(1) <button> 태그 - 버튼 넣기.....	53
(2) <output> 태그 - 결과 출력하기.....	54
(3) <progress> 태그 - 진행 상태 보여주기.....	54
(4) <meter> 태그 - 값이 차지하는 크기 표시하기.....	55
[과제] 연습문제 .....	56
5 CSS 기초 .....	57
5.1 스타일과 스타일 시트 .....	57
(1) 왜 스타일을 사용할까? .....	57
(2) 스타일의 형식 .....	57
(3) 스타일을 표기하는 방법 .....	57
(4) 스타일 주석 .....	57
(5) 스타일과 스타일 시트 .....	57
5.2 주요 선택자 .....	59
(1) 전체 선택자 - 모든 요소에 스타일 적용하기.....	59
(2) 태그 선택자 - 특정 태그에 스타일 적용하기 .....	59
(3) 클래스 선택자 - 특정 부분에 스타일 적용하기.....	59
(4) id 선택자 - 특정 부분에 스타일 적용하기.....	60
(5) 그룹 선택자 - 둘 이상 요소에 같은 스타일 적용하기.....	60
5.3 캐스캐이딩 스타일 시트(CSS).....	60
(1) 캐스캐이딩(Cascading)의 의미.....	60
(2) 스타일 우선순위 .....	61
(3) 스타일 상속 .....	61
5.4 CSS3와 CSS 모듈 .....	62
(1) CSS3란? .....	62
(2) CSS3와 브라우저 접두사(prefix).....	62
6 텍스트 관련 스타일 .....	64
6.1 글꼴 관련 스타일 .....	64
(1) font-family 속성 - 글꼴 지정하기.....	64
(2) @font-face 속성 - 웹 폰트 사용하기.....	64
(3) font-size 속성 - 글자 크기 조절하기.....	65
(4) font-weight 속성 - 글자 굵기 지정하기.....	66
(5) font-variant 속성 - 작은 대문자로 표시하기.....	66
(6) font-style 속성 - 글자 스타일 지정하기.....	66
(7) font 속성 - 글꼴 속성을 한꺼번에 묶어서 표현하기.....	67
6.2 텍스트 스타일 .....	68
(1) color 속성 - 글자 색 지정하기.....	68
(2) text-decoration 속성 - 텍스트에 줄 표시하기/없애기 .....	68
(3) text-transform 속성 - 텍스트 대·소문자 변환하기.....	69
(4) text-shadow 속성 - 텍스트에 그림자 효과 추가하기.....	69
(5) white-space 속성 .....	70
(6) letter-spacing과 word-spacing 속성 - 텍스트 간격 조절하기.....	70

6.3 문단 스타일 .....	71
(1) direction 속성 .....	71
(2) text-align 속성 - 텍스트 정렬하기 .....	71
(3) text-justify 속성 - 정렬 시 공백 조절하기 .....	72
(4) text-indent 속성 - 텍스트 들여쓰기 .....	72
(5) line-height 속성 - 줄간격 조절하기 .....	72
(6) text-overflow 속성 - 넘치는 텍스트 표기하기 .....	73
[실습] 상품 소개 페이지 - 텍스트 스타일 활용하기 .....	73
6.4 목록과 링크 스타일 .....	73
(1) list-style-type 속성 - 목록의 불릿과 번호 스타일 지정하기 .....	73
(2) list-style-image 속성 - 불릿 대신 이미지 넣기 .....	74
(3) list-style-position 속성 - 목록에 들여쓰기 효과 내기 .....	74
(4) list-style 속성 - 리스트 속성 한꺼번에 표시하기 .....	75
[과제] 연습문제 .....	75
7 색상과 배경을 위한 스타일 .....	76
7.1 웹에서 색상 표현하기 .....	76
(1) 16진수 표기법 .....	76
(2) rgb와 rgba 표기법 .....	76
(3) hsl과 hsla 표기법 .....	76
(4) 색상 이름 표기법 .....	76
(5) 색상 추출 사이트 이용하기 .....	77
7.2 배경색과 배경 이미지 .....	77
(1) background-color 속성 - 배경색 지정하기 .....	77
(2) background-clip 속성 - 배경 적용 범위 조절하기 .....	78
(3) background-image 속성 - 웹 요소에 이미지 넣기 .....	79
(4) background-repeat 속성 - 배경 이미지 반복하기 .....	79
(5) background-size 속성 - 배경 이미지 크기 조절하기 .....	79
(6) background-position 속성 - 배경 이미지 위치 조절하기 .....	80
(7) background-origin 속성 - 배경 이미지 위치 기준 조절하기 .....	81
(8) background-attachment 속성 - 배경 이미지 고정하기 .....	82
(9) background 속성 - 속성 하나로 사용하기 .....	83
7.3 그라데이션 효과로 배경 꾸미기 .....	83
(1) 그라데이션 브라우저 접두사 .....	83
(2) 선형 그라데이션 .....	84
(3) 원형 그라데이션 .....	85
(4) 그라데이션 반복 .....	86
[과제] 연습문제 .....	87
8. 레이아웃을 위한 스타일 .....	88
8.1. CSS와 박스 모델 .....	88
(1) 블록 레벨 요소와 인라인 레벨 요소 .....	88
(2) 박스 모델(box model) - 박스 형태의 콘텐츠 .....	88
(3) width, height 속성 - 콘텐츠 영역의 크기 .....	89
(4) display 속성 - 화면 배치 방법 결정하기 .....	90
8.2. 테두리 관련 속성들 .....	92
(1) border-style 속성 - 테두리선 스타일 지정하기 .....	92
(2) border-width 속성 - 테두리 두께 지정하기 .....	93
(3) border-color 속성 - 테두리 색상 지정하기 .....	93
(4) border 속성 - 테두리 스타일 묶어서 지정하기 .....	93
(5) border-radius 속성 - 박스 모서리 둥글게 만들기 .....	94
(6) box-shadow 속성 - 선택한 요소에 그림자 효과 내기 .....	95
8.3. 여백을 조절하는 속성들 .....	95
(1) margin 속성 - 요소 주변 여백 설정하기 .....	95

(2) padding 속성 - 콘텐츠 영역과 테두리 사이 여백 설정하기 .....	96
[실습] 박스 모델로 콘텐츠 정리하기 .....	97
[과제] 연습문제 .....	97
9. CSS 포지셔닝 .....	98
9.1. 여러 요소를 원하는 형태로 배치하기 .....	98
(1) CSS 포지셔닝이란? .....	98
(2) box-sizing 속성 - 박스 너비 기준 정하기 .....	98
(3) float 속성 - 왼쪽이나 오른쪽으로 배치하기 .....	98
(4) clear 속성 - float 속성 해제하기 .....	99
[실습] 2단 레이아웃 만들기 1 & 2 .....	100
(5) position 속성 - 배치 방법 지정하기 .....	100
(6) visibility 속성 - 요소를 보이게 하거나 보이지 않게 하기 .....	101
(7) z-index 속성 - 요소 쌓는 순서 정하기 .....	101
9.2. 다단으로 편집하기 .....	102
(1) column-width - 단의 너비 고정하고 다단 구성하기 .....	102
(2) column-count 속성 - 단의 개수 고정하고 다단 구성하기 .....	103
(3) column-gap 속성 - 단과 단 사이 여백 지정하기 .....	104
(4) column-rule 속성 - 구분선의 색상, 스타일, 너비 지정하기 .....	104
(5) break-after 속성 - 다단 위치 지정하기 .....	104
(6) column-span 속성 - 여러 단을 하나로 합치기 .....	106
9.3. 표 스타일 .....	106
(1) caption-side 속성 - 표 제목 위치 정하기 .....	106
(2) border 속성 - 표 테두리 스타일 지정하기 .....	106
(3) border-collapse 속성 - 테두리 통합, 분리하기 .....	107
(4) border-spacing 속성 - 인접한 셀 테두리 사이 거리 지정하기 .....	108
(5) empty-cells 속성 - 빈 셀의 표시 여부 지정하기 .....	108
(6) width, height 속성 - 표 너비와 높이 지정하기 .....	108
(7) table-layout 속성 - 콘텐츠에 맞게 셀 너비 지정하기 .....	108
(8) text-align 속성 - 셀 안에서 수형 정렬하기 .....	108
(9) vertical-align 속성 - 셀 안에서 수직 정렬하기 .....	108
[과제] 연습문제 .....	109
10. HTML5와 시멘틱 태그 .....	110
10.1. HTML4 문서 vs HTML5 문서 .....	110
(1) HTML4 문서 .....	110
(2) 시멘틱 태그가 사용된 HTML5 문서 .....	110
(3) 왜 시멘틱 태그로 레이아웃을 만들어야 할까? .....	111
10.2. 문서 구조를 위한 HTML5 시멘틱 태그들 .....	111
(1) <header> 태그 - 머리말(제목) 지정하기 .....	111
(2) <nav> 태그 - 문서를 연결하는 내비게이션 링크 .....	111
(3) <section> 태그 - 주제별 콘텐츠 영역 나타내기 .....	111
(4) <article> 태그 - 실제 콘텐츠 내용 넣기 .....	111
(5) <aside> 태그 - 본문 이외의 내용 표시하기 .....	111
(6) <iframe> 태그 - 외부 문서 삽입하기 .....	112
(7) <footer> 태그 - 제작 정보와 저작권 정보 표시하기 .....	112
(8) <address> 태그 - 사이트 제작자 정보, 연락처 정보 나타내기 .....	112
(9) <div> 태그는 언제 사용할까? .....	112
10.3. IE8 이하 버전에서는 어떻게 하나요? .....	112
(1) 새로운 시멘틱 태그 지원 상황 .....	112
(2) IE8 이하에서 시멘틱 태그를 사용하려면? .....	112
11. HTML5와 멀티미디어 .....	114
11.1. 웹과 멀티미디어 .....	114
(1) 플러그인 프로그램 .....	114

(2) <object> 태그와 <embed> 태그 - 플러그인 사용하기 .....	114
(3) 멀티미디어의 웹 표준화 .....	114
(4) HTML5 비디오 변환하기 .....	114
11.2. 오디오 & 비디오 재생하기 .....	114
(1) <audio> 태그 - 오디오 파일 삽입하기 .....	115
(2) <video> 태그 - 비디오 파일 삽입하기 .....	115
(3) <source> 태그 - 여러 미디어 파일 한꺼번에 지정하기 .....	115
(4) 이전 브라우저에서는 어떻게 해야 할까? .....	116
(5) <audio> 태그와 <video> 태그의 속성 .....	116
[실습] 웹 문서에 멀티미디어 파일 넣기 .....	116
(6) <track> 태그 - 비디오 화면에 자막 추가하기 .....	116
[실습] 동영상으로 홈페이지 대문 만들기 .....	116
12. 다재다능한 CSS3 선택자 .....	116
12.1. 연결 선택자(combination selector) .....	116
(1) 하위 선택자 - 지정한 모든 하위 요소에 스타일 적용하기 .....	116
(2) 자식 선택자 - 자식 요소에만 스타일 적용하기 .....	116
(3) 인접 형제 선택자 - 가장 가까운 형제 요소에 스타일 적용하기 .....	117
(4) 형제 선택자 - 형제 요소에 스타일 적용하기 .....	117
12.2. 속성 선택자 .....	117
(1) [속성] 선택자 - 지정한 속성에 스타일 적용하기 .....	117
(2) [속성 = 값] 선택자 - 특정 값을 갖는 속성에 스타일 적용하기 .....	117
(3) [속성 ~ 값] 선택자 - 여러 값 중 특정 값이 포함된 속성에 스타일 적용하기 .....	117
(4) [속성 != 값] 선택자 - 특정 값이 포함된 속성에 스타일 적용하기 .....	118
(5) [속성 ^= 값] 선택자 - 특정 값으로 시작하는 속성에 스타일 적용하기 .....	118
(6) [속성 \$= 값] 선택자 - 특정 값으로 끝나는 속성에 스타일 적용하기 .....	118
(7) [속성 *= 값] 선택자 - 값의 일부가 일치하는 속성에 스타일 적용하기 .....	118
12.3. 가상 클래스와 가상 요소 .....	119
(1) 사용자 동작에 반응하는 가상 클래스 .....	119
(2) UI 요소 상태에 따른 가상 클래스 .....	119
(3) 구조 가상 클래스 .....	120
(4) 그 외 가상 클래스 .....	121
(5) 가상 요소 .....	121
[과제] 연습문제 .....	121
13. CSS3와 애니메이션 .....	122
13.1. 변형 .....	122
(1) 2차원 변형과 3차원 변형 .....	122
(2) transform과 변형 함수 .....	122
(3) translate 변형 함수 - 요소 이동시키기 .....	122
(4) scale 변형 함수 - 요소 확대/축소하기 .....	122
(5) rotate 변형 함수 - 요소 회전하기 .....	122
(6) skew 변형 함수 - 요소를 비틀어 왜곡하기 .....	122
13.2. 변형과 관련된 속성들 .....	122
(1) transform-origin 속성 - 변형 기준점 설정하기 .....	122
(2) perspective, perspective-origin 속성 - 원근감 표현하기 .....	123
(3) transform-style 속성 - 3D 변형 적용하기 .....	123
(4) backface-visibility 속성 - 요소의 뒷면 표시하기 .....	123
13.3. 트랜지션 .....	123
(1) 트랜지션이란? .....	123
(2) transition-property 속성 - 트랜지션을 적용할 속성 지정하기 .....	123
(3) transition-duration 속성 - 트랜지션 진행 시간 지정하기 .....	124
(4) transition-timing-function 속성 - 트랜지션 속도 곡선 지정하기 .....	124
(5) transition-delay 속성 - 지연 시간 설정하기 .....	125

(6) transition 속성 - 트랜지션 속성 한꺼번에 표기하기 .....	125
13.4. 애니메이션 .....	125
(1) CSS와 애니메이션 .....	125
(2) @keyframes 속성 - 애니메이션 지점 설정하기 .....	126
(3) animation-name 속성 - 애니메이션 이름 지정하기 .....	126
(4) animation-duration 속성 - 애니메이션 실행 시간 설정하기 .....	126
(5) animation-direction 속성 - 애니메이션 방향 지정하기 .....	126
(6) animation-iteration-count 속성 - 반복 횟수 지정하기 .....	126
(7) animation-timing-function 속성 - 애니메이션 속도 곡선 지정하기 .....	126
(8) animation 속성 - 애니메이션 관련 속성 한꺼번에 표기하기 .....	126
[실습] 상품 페이지에 가격 떠오르게 만들기 .....	128
[과제] 연습문제 .....	128
14. 반응형 웹이란? .....	130
14.1. 모바일 기기와 웹 디자인 .....	130
(1) 반응형 웹 디자인 .....	130
(2) 반응형 웹의 장단점 .....	130
(3) 모바일 기기를 위한 기본 다지기, 뷰포트 .....	130
(4) 뷰포트 지정하기 .....	130
[실습] 크롬의 디바이스 모드 활용하기 .....	131
14.2. 가변 그리드 레이아웃 .....	131
(1) 고정 그리드와 가변 그리드 .....	131
(2) 가변 그리드 레이아웃 만들기 .....	131
14.3. 가변 레이아웃과 가변 요소 .....	132
(1) 가변 글꼴 .....	132
(2) 가변 이미지 .....	133
(3) 가변 비디오 .....	134
[실습] 가변 그리드 레이아웃 구성하기 .....	134
14.4. 미디어 쿼리 .....	134
(1) 미디어 쿼리란? .....	134
(2) 미디어 쿼리 구문 .....	134
(3) 미디어 쿼리의 조건 .....	135
14.5. 미디어 쿼리를 사용해 사이트 구성하기 .....	137
(1) 외부 CSS 파일 연결하기 .....	137
(2) 웹 문서에서 직접 정의하기 .....	138
[실습] 모바일용 문서 작성하기 .....	138
[실습] 태블릿용 미디어 쿼리 작성하기 .....	138
[실습] PC용 미디어 쿼리 작성하기 .....	138
14.6. 플렉서블 박스 레이아웃 .....	138
(1) 플렉서블 박스 레이아웃과 새로운 용어 .....	138
(2) 플렉서블 박스 레이아웃 기본 속성 .....	139
(3) 플렉스 항목 배치를 위한 속성들 .....	144
[실습] 플렉스 박스를 이용해 사이트 구성하기 .....	145
15. 반응형 웹 사이트 만들기 .....	147
15.1. 레이아웃 구조 만들기 .....	147
(1) 레이아웃 구상 .....	147
(2) 미디어 쿼리 중단점 만들기 .....	147
15.2. 각 기기용 레이아웃 만들기 .....	147
[실습] 문서 기본 구조 만들기 .....	147
[실습] 모바일용 레이아웃 구조 만들기 .....	147
[실습] 웹 폰트 사용 및 메뉴 만들기 .....	147
[실습] 태블릿용 및 PC용 레이아웃 만들기 .....	147





## 0 훈련 준비

### 0.1 웹표준1(HTML5+CSS3) 학습 진도표

학습 목표	학습 내용	학습 소요시간
처음 시작하는 HTML5	1. HTML 기본 다지기	
	2. 텍스트 관련 태그들	
	3. 이미지와 하이퍼링크	
	4. 폼 관련 태그들	
CSS3로 완성하는 웹 표준	5. CSS 기초	
	6. 텍스트 관련 스타일	
	7. 색상과 배경을 위한 스타일	
	8. 레이아웃을 위한 스타일	
HTML, CSS 한 걸음 더 나아가기	9. CSS 포지셔닝	
	10. HTML5와 시맨틱 태그	
	11. HTML5와 멀티미디어	
	12. 다재다능한 CSS3 선택자	
반응형 웹사이트 만들기	13. CSS3와 애니메이션	
	14. 반응형 웹이란	
	15. 반응형 웹 사이트 만들기	
총계	기초부터 반응형 웹까지 학습 달성	

# 1 HTML 기본 다지기

## 1.1 HTML과의 첫 만남

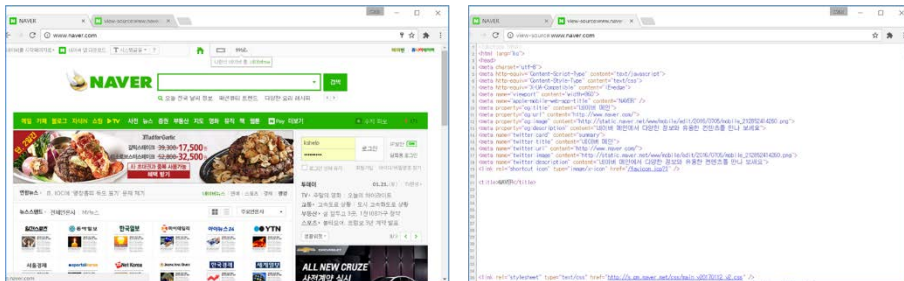
### (1) HTML이란 무엇일까?

- 컴퓨터에서 사용하는 모든 파일에는 각각 고유의 형식이 있다.
- 웹에서는 웹에 맞는 형식인 \*.html(또는 \*.htm)로 문서를 저장해야 한다.
- 텍스트뿐만 아니라 이미지, 링크 등 여러 요소들을 다루고 표시할 수 있어야 한다.
- 웹에서 자유롭게 오갈 수 있는 웹 문서를 만드는 언어가 HTML



### (2) HTML 소스 엿보기

실제로 우리가 자주 들어가는 웹 사이트도 사실 HTML과 CSS, Javascript로 이루어져 있다.



### (3) 웹 표준이란 무엇일까?

- 웹 사이트를 만들 때 지켜야 하는 약속들을 정리한 것
- 웹 표준을 지켜 사이트를 제작하면 장소나 브라우저와 상관없이 쉽게 웹 사이트를 볼 수 있다.
- 웹 표준으로 문서 하나를 만들면 어떤 기기에서나 볼 수 있기 때문에 웹 개발자와 디자이너의 시간 절약
- HTML5로 문서를 만드는 것 = 웹 표준을 지킨 문서를 만드는 것

### (4) 왜 HTML5와 CSS3를 공부해야 할까?

- 최신 웹 표준에 맞는 웹 사이트 제작
  - HTML5와 CSS3를 사용하면 사용자가 접속한 기기에 따라 사이트 레이아웃을 다양하게 바꿀 수 있다.
- 앱 화면을 디자인하기 위한 기초
  - HTML5의 API를 사용해 '웹앱'을 만들 수도 있고, 앱 화면은 HTML5와 CSS3를 사용해 디자인
- 인터랙티브한 사이트 제작

- 플러그인을 제작할 줄 몰라도 HTML5과 함께 CSS3를 이용하면 인터랙티브한 사이트 제작 가능
- 사이트와 블로그 수정이 쉽다
  - HTML5를 공부하면 사이트나 블로그 소스를 이해할 수 있고 사이트를 원하는 모습으로 바꿀 수 있다.

## 1.2 웹 브라우저와 웹 편집기

### (1) 웹 브라우저 종류와 HTML5 지원 정도

브라우저 이름	제조사	특징
크롬 	구글	빠른 업데이트를 통해 HTML5 표준에 가장 잘 대응하고 있어 최근 가장 많이 사용되는 브라우저입니다.
파이어폭스 	모질라	일반 사용자들보다 개발자들이 자주 사용하는 웹 브라우저로 개발 도구와 부가 기능들이 뛰어납니다.
인터넷 익스플로러 	마이크로소프트	HTML5에 대한 지원은 취약하지만 아직까지 인터넷 익스플로러(IE)를 기준으로 하는 사이트가 많기 때문에 국내에서 많이 사용되고 있습니다. 윈도우 7 서비스팩1부터 지원하기 시작한 IE11을 마지막으로 더 이상 개발되지 않고 있습니다. ▶ 윈도우 10에서는 '보조 프로그램' 안에 'IE11'이 포함되어 있습니다.
엣지 	마이크로소프트	마이크로소프트에서 인터넷 익스플로러 대신 HTML5를 지원하기 위해 새로 개발해 윈도우 10에서 기본으로 사용하는 웹 브라우저입니다.

▶ 이외에도 애플의 '사파리'나 오페라 재단의 '오페라' 브라우저도 있습니다.

사용 중인 웹 브라우저가 HTML5를 얼마나 지원하는지 확인하려면 <http://html5test.com>

### [실습] 크롬 설치하고 기본 웹 브라우저로 정하기

p24 ~ p25

## (2) 웹 편집기의 종류

편집기 종류	특징
텍스트 편집기	<p>적절한 프로그램은 아니지만 HTML 태그에 익숙하다면 윈도우의 메모장이나 매킨토시의 텍스트 편집기를 이용해서도 웹 문서를 작성할 수 있습니다. (물론 한글이나 워드에서도 웹 문서를 작성할 수 있죠) 사용자가 태그를 일일이 직접 입력해야 하기 때문에 HTML 태그와 속성을 알고 있어야 하고 입력할 때 오류가 발생하기 쉽습니다.</p>  메모장
웹 편집기	<p>사용할 것 같은 편집기로 웹을 위한 전용 편집기입니다. 태그와 속성, 일반 텍스트 등을 서로 다른 색으로 표시해주므로 소스를 읽기 쉽고 편집기에 따라 태그나 속성의 일부만 입력하더라도 전체를 완성해주기 때문에 오타로 인한 문제를 줄일 수 있습니다. 윈도우용 편집기로는 노트패드++(무료)나 에디트 플러스(유료) 등이 있고 맥용 편집기로는 텍스트메이트(Textmate)(무료)나 코다(Coda)(유료) 등이 있으며 드림위버(Dreamweaver)(유료)나 비주얼 스튜디오 코드(Visual Studio Code)(무료) 같은 편집기는 윈도우와 맥에서 모두 사용 가능합니다.</p> <p>▶ 이 책에서는 비주얼 스튜디오 코드를 이용해 소스 코드를 작성할 것입니다.</p> <div>  노트패드++              에디트 플러스              텍스트 메이트         </div> <div>  코다              드림위버              비주얼 스튜디오 코드         </div>
편집기 종류	특징
통합개발환경(IDE)	<p>웹을 개발할 때 대부분 단순히 HTML만 사용하는 것이 아니라 자바스크립트나 ASP.NET, PHP, node.js 등 여러 언어들을 함께 사용합니다. 웹 IDE는 웹 편집기의 편리함과 함께 사용자에게 필요한 개발 환경을 한꺼번에 구축하고 사용할 수 있게 해줍니다. 웹 IDE 프로그램에는 비주얼 스튜디오 2015(Visual Studio 2015)(무료, 유료), 웹스톰(Webstorm)(무료), 서브라임 텍스트(Sublime Text)(무료, 유료) 등이 있습니다.</p> <div>  비주얼 스튜디오 2015              웹스톰              서브라임 텍스트         </div>
웹 기반 코드 편집기	<p>웹 기반 코드 편집기는 웹 상에서 소스 코드를 입력하고 결과를 바로 확인할 수 있는 편집기입니다. 웹 코딩을 위한 시스템 환경을 만들거나 별도의 편집기 프로그램을 이용하지 않고 브라우저 화면에 코드를 입력하고 결과를 볼 수 있다는 장점이 있으며 팀 작업을 하면서 여러 팀원들과 소스를 공유하는 데도 유용하고 초보자의 코딩 연습용으로도 적합합니다. Codepen(<a href="http://codepen.io/">http://codepen.io/</a>)이나 JSFiddle(<a href="https://jsfiddle.net/">https://jsfiddle.net/</a>), Liveweave(<a href="http://liveweave.com/">http://liveweave.com/</a>), Plunker(<a href="http://plnkr.co/">http://plnkr.co/</a>) 등이 있습니다.</p>

## (3) 통합개발환경(IDE) - 이클립스 소개

- 2003년 IBM에서 개발
- 자바 통합 개발 환경(IDE: Integrated Development Environments) 제공
- 프로젝트 생성 기능 제공
- 자동 코드 완성 기능 제공
- 디버깅 기능 제공
- 이클립스 연합(Eclipse Foundation) 설립 - 지속적 버전업과 배포
- 다양한 개발 환경을 구축할 수 있도록 플러그인(Plug-In) 설치 가능
- 안드로이드 개발 환경
- 스프링(Spring) 개발 환경
- C, C++ 개발 환경

### [실습] 이클립스 설치하고 살펴보기

#### 1. 자바 개발 도구(JDK) 설치

- JRE = JVM + 표준 클래스 라이브러리

- JDK = JRE + 개발에 필요한 도구
- 패스(path)설정
- 내컴퓨터 우클릭/속성/고급시스템설정/고급탭/환경 변수/시스템 변수/새로만들기/  
변수이름 JAVA\_HOME  
변수값 C:\Program Files\Java\jdk버전
- path 편집/변수값 뒤에 (세미콜론);%JAVA\_HOME%\bin

## 2. Apache Tomcat 설치

- 다운로드 사이트: <http://tomcat.apache.org/>

## 3. 이클립스 설치

- 이클립스는 자바 언어로 개발된 툴 - JDK 필요
- 다운로드 사이트: <https://www.eclipse.org/downloads/>
  - Eclipse IDE for Java Developers 버전: 순수 자바 학습용
  - Eclipse IDE for Java EE Developers 버전 (수업에서 사용하는 버전): 웹 애플리케이션 등의 Enterprise (Network) 환경에서 실행
  - CPU 사양에 맞게 다운로드
- Preferences 설정
  - 에디터의 인코딩 설정: Eclipse > Windows > Preferences > General/Workspace > Text file encoding > "Other: UTF-8" 선택
  - HTML 파일 인코딩 설정: Eclipse > Windows > Preferences > Web > HTML Files > "Encoding: ISO 10646/Unicode(UTF-8)" 선택
  - 웹 브라우저 변경: Window > Preferences > General > Web Browser > Use external web browser & Chrome 선택

## 1.3 HTML 기본 문서 구조

### (1) 태그, 이건 꼭 알아두세요!

- 태그는 소문자로 쓴다  
예) 
- 여는 태그와 닫는 태그를 정확히 입력한다 (닫는 태그가 없는 태그도 있다)
- 적당하게 들여쓴다
- 태그는 속성과 함께 사용된다  
예) 
- 인코딩 방식은 utf-8로 한다  
예) <meta charset="utf-8">

### [실습] 간단한 HTML 문서 만들기

p37 ~ p41

## (2) HTML 문서 기본 구조 살펴보기



① <!doctype html> - 현재 문서가 HTML5 언어로 작성된 웹 문서라는 뜻입니다.

② <html> ~ </html> - 웹 문서의 시작과 끝을 나타내는 태그입니다. 웹 브라우저가 <html> 태그를 만나면 </html>까지의 소스를 읽어 HTML 문법에 맞추어 브라우저에 표시합니다.

③ <head> ~ </head> - 웹 브라우저가 웹 문서를 해석하기 위해 필요한 정보들을 입력하는 부분입니다. 여기

에 있는 정보는 실제 문서 내용이 아니기 때문에 문서 제목만 브라우저 창에 표시되고 나머지는 웹 브라우저 화면에 표시되지 않습니다. 스타일 및 스크립트 등이 포함되기도 합니다.

④ <body> ~ </body> - 실제로 웹 브라우저 화면에 나타날 내용입니다. 앞으로 우리가 공부하게 될 HTML 태그들은 대부분 <body> 태그 안에 들어갑니다.

## (3) <!doctype> - 문서 유형을 지정하는 선언문

- <!doctype> - 웹 브라우저에게 '이제부터 처리할 문서는 HTML 문서이고 어떤 유형을 사용했으니 그 버전에 맞는 방법으로 해석하라.' 라고 알려주는 것
- 기본형: <!DOCTYPE html> 또는 <!doctype html>

## (4) <html> 태그 - 웹 문서 시작을 알리는 태그

- 실제 문서 정보와 내용이 시작되고 끝나는 것을 표시하는 태그
- lang이라는 속성을 사용해 문서에서 사용할 언어 지정 가능. 예) <html lang="ko">
- <html> 태그와 </html> 태그 사이에는 문서 정보를 지정하는 <head> 부분과 실제 화면에 보이는 문서 내용을 입력하는 <body> 태그가 포함된다.

## (5) <head> 태그 - 브라우저에게 정보를 주는 태그

- 웹 브라우저 화면에는 보이지 않지만, 웹 브라우저가 알아두어야 할 정보들 입력
- <title> 태그: 브라우저의 제목 표시줄에 표시되는 내용.  
예) <title> 문서 제목 </title>
- <meta> 태그: 문자 인코딩 방법 및 문서의 키워드와 요약 정보를 지정  
예) <meta charset="utf-8">
- <link> 태그: 문서에서 사용할 외부 파일들 링크  
예) <link rel="stylesheet" href="mail.css">

#### (6) <body> 태그 - 실제 브라우저에 표시될 내용

- 실제 브라우저에 표시될 내용 입력
- 대부분의 태그가 <body> 태그와 </body> 태그 사이에서 사용한다.

#### (7) 웹 문서에서 특수 문자 및 특수 기호 사용하기

- 키보드에서 한글 자음을 누른 후 <한자>키를 누르면 특수문자가 표시된다. 한글 각 자음마다 서로 다른 특수 문자가 표시된다

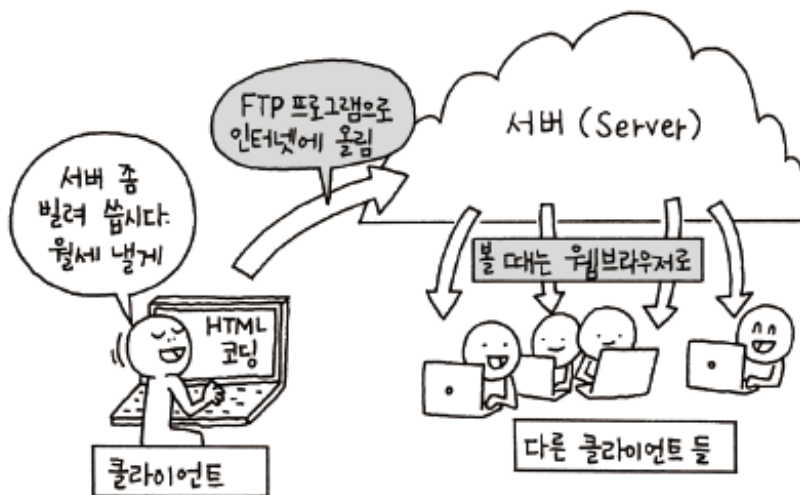
1	#	●	▼	▶	◆	■	±	♫	⊗
2	&	◎	→	▶	■	⊗	±	⊗	⊗
3	*	◇	←	△	●	♫	±	⊗	⊗
4	@	◆	↑	♠	●	♫	±	⊗	⊗
5	ㅑ	□	↓	♥	■	♫	±	⊗	⊗
6	ㅓ	■	↔	♥	■	♫	±	⊗	⊗
7	ㅕ	△	=	♣	■	♫	±	⊗	⊗
8	ㅗ	▲	◁	♣	■	♫	±	⊗	⊗
9	ㅛ	▽	◁	♣	■	♫	±	⊗	⊗

'ㅁ'과 [한자]를 눌렀을 때 사용할 수 있는 특수 문자

### 1.4 웹 문서 만들고 업로드하기

#### (1) 호스팅 서버 준비하기

- HTML로 만든 웹사이트를 다른 사람들이 볼 수 있도록 하려면 → 웹 문서와 사용한 파일들을 서버 컴퓨터로 업로드해야 한다.
- 웹 서버를 직접 구입하고 관리하기 어렵기 때문에 매달 혹은 몇 년마다 일정 금액을 내고 사용하는 호스팅 서비스를 이용한다.



#### [실습] 닷홈에 무료 호스팅 서비스 신청하기

p50 ~ p52

#### (2) FTP 프로그램 설치하기

- FTP 프로그램 : 사용자 컴퓨터에서 작성한 웹 문서와 각종 파일을 서버로 업로드(upload)하거나 서버에서 다운로드(download)할 수 있도록 해주는 프로그램



**[실습] FTP 프로그램 FileZilla 설치하고 서버에 접속하기**  
p53 ~ p55

**[실습] 내가 만든 문서 웹 서버에 올리고 확인하기**  
p56 ~ p57

## 2 텍스트 관련 태그들

### 2.1 텍스트를 덩어리로 묶어 주는 태그

#### (1) <h> 태그 - 제목 표시하기

- 각 웹 콘텐츠 영역에서 제목을 표시할 때 사용하는 태그
- <h1>, <h2>, <h3>, <h4>, <h5>, <h6> 태그처럼 여섯 단계로 나누어져 있고, 숫자가 클수록 그 크기가 작아진다.
- 기본형: <h> 텍스트 </h>
- <h> 태그는 문서 안에서 여러 번 사용할 수 있으며, 같은 <h> 태그라도 스타일을 사용해서 서로 다른 형태로 표시할 수 있다.
- <p> 태그를 사용한 후 글자 크기를 크게 하거나 진하게 하는 식으로 스타일을 적용해 제목 텍스트를 만들기도 하는데, 이것은 웹 표준에 맞지 않는다.

#### (2) <p> 태그 - 단락 만들기

- 텍스트 단락을 만든다.
- <p> 태그로 표시하는 텍스트 앞뒤에서 줄바꿈이 일어난다.
- 기본형: <p> 내용 </p>

#### (3) <br> 태그 - 줄 바꾸기

- <br> 태그가 삽입한 위치에서 줄이 바뀐다. 닫는 태그는 없다.

#### (4) <hr> 태그 - 분위기 전환을 위한 수평 줄 넣기

- 수평 줄을 삽입할 때 사용한다.
- 기본형: <hr>

#### (5) <blockquote> 태그 - 인용문 넣기

- 다른 블로그나 사이트에서 인용할 경우 인용 내용 표시
- cite 속성을 이용해 인용 사이트 주소를 표시할 수 있다.
- 기본형: <blockquote> 인용 내용 </blockquote>

[02/text-1.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="utf-8">
  <title>제주 이색 여행지</title>
</head>
<body>
  <h1>제주 이색 여행지</h1>
  <h2>야외 텐트를 닮은 건축물 "테쉬폰"</h2>
  <p>아일랜드 출신 임피제 신부가 1954년 제주에 오면서 목장 숙소로 짓기 시작한 후 사료공장, 성당으로 활용됐습니다.</p>
  <p>제주에서 점차 다른 지방으로 보급됐지만 현재 제주에만 건축물이 남아있는데, <br>
  국내 근현대 건축사의 한 페이지를 보여주는 가치를 지닌다고 전문가들은 평가합니다.</p>
  <hr>
  <blockquote>
    성이시돌목장은 제주특별자치도 제주시 한림읍 금악리에 있는 목장이다. 특히 이시돌목장은 제주 지역 최초의 전기업목장(❧❧❧❧)으로 1961년 11월 말 제주시 한림읍 금악리에 세워 양돈 사업을 실시하였으며 면양을 사육하였던 것으로 알려져 있다. 이시돌목장의 특색있는 건축양식으로 테쉬폰도 유명하다.
    (출처:향토문화전자대전)
  </blockquote>
</body>
</html>
```

## (6) <pre> 태그 - 입력하는 그대로 화면에 표시하기

- <pre> 태그를 사용할 경우 소스에 표시한 공백이 브라우저에 그대로 표시된다.
- 주로 프로그램 소스를 표시할 때 사용한다.
- 기본형: <pre> 내용 </pre>

[02/text-2.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="utf-8">
  <title>로컬 스토리지(Local Storage)를 저장하는 함수</title>
</head>
<body>
  <h3>로컬 스토리지(Local Storage)를 저장하는 함수 : </h3>
  <pre>
function savetheLocal(){
  var second = document.getElementById("second");
  var thevalue = second.value;
  localStorage.setItem(1, thevalue);
  gettheLocal();
}
  </pre>
</body>
</html>
```

### [참조] 특수문자 사용법

- 공백(스페이스 한 칸): &nbsp;
- 부등호(< >): &lt; &gt;
- 앰퍼샌드(&): &amp;
- 쌍따옴표("): &quot;

```
aaa&nbsp;&nbsp;&nbsp;bbb <!-- aaa bbb -->
A&amp;B <!-- A&B -->
<h1>제목이 &lt;여기에&gt; 표시됩니다</h1> <!-- 제목이 <여기에> 표시됩니다 -->
<a href="http://www.google.co.kr/">구글 &quot;검색&quot; 홈페이지</a> <!-- 구글 "검색" 홈페이지 -->
```

## 2.2 텍스트를 한 줄로 표시하는 태그

### (1) <strong> 태그, <b> 태그 - 굵게 표시하기

- 눈으로 보기에는 차이가 느껴지지 않는 두 태그를 구분하는 이유는 화면 낭독기에서의 기능 때문이다.
- 기본형: <strong> 굵게 강조할 텍스트 </strong>  
          <b> 굵게 표시할 텍스트 </b>

### (2) <em> 태그, <i> 태그 - 이탤릭체로 표시하기

- 중요한 내용을 담고 있는 것이 <em> 태그이고 단순히 이탤릭체로 표시하는 것이 <i> 태그이다.
- 기본형: <em> 이탤릭체로 강조할 텍스트 </em>  
          <i> 이탤릭체로 표시할 텍스트 </i>

[02/text-3.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
```

```

<meta charset="utf-8">
<title>제주 이색 여행지</title>
<style>
  p {
    font-size: 16px; /* 글자 크기 */
    line-height: 25px; /* 줄 간격 */
  }
</style>
</head>
<body>
  <h2>제주 이색 여행지 - 이중섭 거리</h2>
  <p><strong>주말</strong>마다 <b>'서귀포문화예술디자인시장'</b>이 열립니다.</p>
  <p><em>'아트마켓'</em>이라고도 부르는데, <i>문화예술체험</i>이나 <i>공연관람</i>을 할 수도 있고 <br>
  작가들이 직접 만든 창작예술품 등을 판매하기도 합니다.</p>
</body>
</html>

```

### (3) <q> 태그 - 인용 내용 표시하기

- 인라인 레벨 태그이기 때문에 줄바꿈 없이 다른 내용과 함께 한 줄로 표시되고 인용 내용에 따옴표를 붙여 표시한다. 반면 <blockquote> 태그는 블록 레벨 태그이기 때문에 인용 내용이 줄이 바뀌어 나타나고 다른 내용과 구별되도록 안으로 들여 써진다.
- 기본형: <q> 인용 내용 </q>

[02/text-4.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="utf-8">
  <title>웹 접근성(Web Accessibility)</title>
</head>
<body>
  <h1>웹 접근성</h1>
  <p>웹의 창시자인 팀 버너스 리 (Tim Berners-Lee)의 <q
  cite="http://www.w3.org/standards/webdesign/accessibility">웹의 힘은 보편성에 있다. 장애에 구애없이 모든
  사람이 접근할 수 있는 것이 필수적인 요소이다.</q>라는 말로 웹 접근성을 설명한다. </p>
</body>
</html>

```

### (4) <mark> 태그 - 형광펜 효과 내기

- 선택한 부분에 형광펜을 그어놓은 듯한 효과
- 배경색이 노란색으로 표시되는 것이 기본
- 기본형: <mark> 내용 </mark>

### (5) <span> 태그 - 줄바꿈 없이 영역 묶기

- 텍스트 단락 안에서 줄바꿈 없이 일부 텍스트만 묶어서 스타일을 적용하려고 할 때 주로 사용
- 기본형: <span> 내용 </span>

[02/text-5.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="utf-8">
  <title>제주 이색 여행지</title>
</head>
<body>

```

```

<h2>야외 텐트를 닮은 건축물 <mark>"테쉬폰"</mark></h2>
<p>아일랜드 출신 임피제 신부가 1954년 제주에 오면서 목장 숙소로 짓기 시작한 후 사료공장, 성당으로
활용됐습니다. 제주에서 점차 다른 지방으로 보급됐지만 현재 제주에만 건축물이 남아있으며, <span
style="color:blue;">국내 근현대 건축사의 한 페이지를 보여주는 가치를 지닌다</span>고 전문가들은
평가합니다.</p>
</body>
</html>

```

## (6) <ruby> 태그 - 동아시아 글자 표시하기

- 동아시아 국가들의 글자들에 주석을 함께 표기하기 위한 용도로 사용된다.
- 기본형: <ruby> 내용 <rt> 주석 </rt> </ruby>

[02/text-6.html]

```

<!doctype html>
<html lang="ko">
  <head>
    <meta charset="utf-8">
    <title>Ruby</title>
  </head>
  <body>
    <p>루비(Ruby)는 1995년, 일본의 프로그래머인 마츠모토 유키hiro
    (<ruby>궂궂궂궂<rt>まつもとゆきひろ</rt></ruby>)가 만든 프로그래밍 언어입니다.</p>
  </body>
</html>

```

## (7) 기타 텍스트 관련 태그들

- <code>: 파일 이름이나 컴퓨터 프로그램 등 컴퓨터가 인식할 수 있는 소스를 표시하는 태그
- <s>: 문서에서 특정 텍스트를 제거한다는 의미로 취소선을 그리는 태그
- <small>: 부가 정보처럼 작게 표시해도되는 텍스트
- <u>: 밑줄

## 2.3 목록을 만드는 태그들

### (1) <ul> 태그, <li> 태그 - 순서 없는 목록 만들기

- 각 항목 앞에 작은 원이나 사각형 같은 불릿(bullet)이 붙는다.
- 기본형: <ul>

```

    <li> 내용 </li>
    <li> 내용 </li>
    ...
</ul>

```

[02/ul.html]

```

<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="utf-8">
    <title>제주 관광 안내</title>
  </head>
  <body>
    <h1>관광 안내 전화 </h1>
    <p>한국관광공사에서는 전국의 관광안내소와 공동으로 여러분의 여행편의를 위해 관광안내전화 1330
    서비스를 연중무휴 실시하고 있습니다.</p>
    <p>1330에는 해당 지역의 지도와 관광 가이드북, 관광안내소를 대신할 수 있을 정도의 다양한 정보가
    있습니다. 원하는 관광지는 물론이며 숙박, 교통, 음식점 등의 자세한 정보를 한국어를 비롯한 영어, 중국어,

```

```

일어의 3개 국어로도 이용 가능합니다.</p>
<ul>
    <li>일반 전화 : (국번없이) 1330</li>
    <li>휴대 전화 : 064-1330</li>
</ul>
</body>
</html>

```

## (2) <ol> 태그, <li> 태그 - 순서 목록 만들기

- 항목을 나열하되 순서가 필요한 목록을 만들때에 사용한다.
- type 속성: 1, a, A, i, I
- start 속성: 순서 목록은 기본적으로 1부터 시작하는데 start 속성을 이용하면 중간 번호부터 시작할 수 있다.
- reserved 속성: 항목을 역순으로 표시한다.

```

[02/ol.html]

<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="utf-8">
    <title>1박 2일 가족 여행 코스</title>
</head>
<body>
    <h1>1박 2일 가족 여행 코스</h1>
    <ul>
        <li>1일차
            <ol type="a">
                <li>해녀박물관</li>
                <li>낙시체험</li>
            </ol>
        </li>
        <li>2일차
            <ol type="a" start="3">
                <li>용눈이오름</li>
                <li>만장굴</li>
                <li>카약체험</li>
            </ol start>
        </li>
    </ul>
</body>
</html>

```

## (3) <dl>, <dt>, <dd> 태그 - 설명 목록 만들기

- 사전 구성처럼 ‘제목’과 ‘설명’이 한 쌍인 설명 목록을 만든다.
- 기본형: <dl>
  - <dt> 제목 </dt>
  - <dd> 설명 </dd>
 </dl>

```

[02/dl.html]

<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="utf-8">
    <title>제주 올레</title>
    <style>

```

```

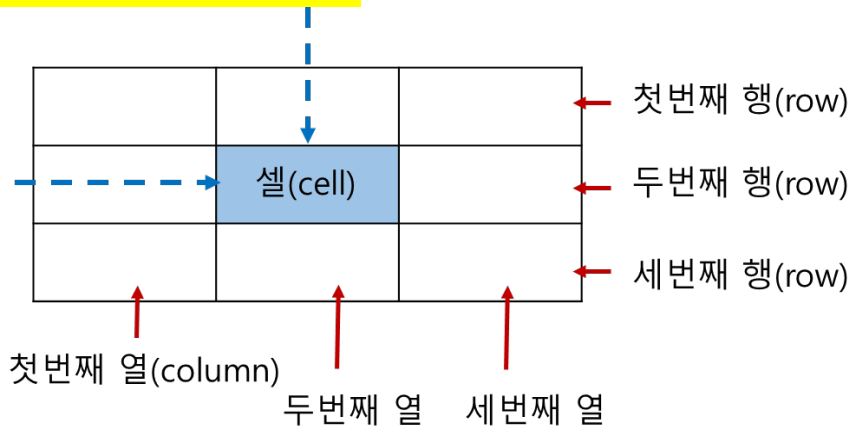
        dl {
            line-height:25px; /* 줄 간격 */
        }
    </style>
</head>
<body>
    <h1>제주 올레 </h1>
    <dl>
        <dt>올레 1코스</dt>
        <dd>코스 : 시흥 초등학교 옆 - 광치기 해변</dd>
        <dd>거리 : 14.6km(4~5시간)</dd>
        <dd>난이도 : 중</dd>
        <dt>올레 2코스</dt>
        <dd>코스 : 광치기 해변 - 온평 포구</dd>
        <dd>거리 : 14.5km(4~5시간)</dd>
        <dd>난이도 : 중</dd>
    </dl>
</body>
</html>

```

## [실습] 온라인 프로필 - 텍스트 입력하기

p75 ~ p78

### 2.4 표를 만드는 태그



#### (1) <table>, <tr>, <td>, <th> 태그 - 기본적인 표 만들기

- 기본형: <table>
 

```

            <tr>
                <td>내용 </td>
                <td>내용 </td>
                ...
            </tr>
            ...
        </table>

```
- <th> 태그: <td> 태그 대신 사용하면 간단히 제목 셀(굵게 표시하고 셀의 중앙에 배치)을 만들 수 있다.
- colspan, rowspan 속성: 행 또는 열 합치기

[02/table-1.html]

```

<!DOCTYPE html>
<html lang="ko">

```

```

<head>
<meta charset="utf-8">
<title>표 만들기</title>
<style>
td {
padding: 5px; /* 셀 테두리와 내용 사이의 간격(패딩) */
}
</style>
</head>
<body>
<table border="1">
<tr>
<th>제목 셀</th>
<td>1행 2열</td>
<td>1행 3열</td>
</tr>
<tr>
<th>제목 셀</th>
<td>2행 2열</td>
<td>2행 3열</td>
</tr>
</table>

</body>
</html>

```

[02/table-2.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>표 만들기</title>
<style>
th {
padding: 15px; /* 셀 테두리와 내용 사이의 간격(패딩) */
}
</style>
</head>
<body>
<table border="1">
<colgroup>
<col style="width: 100px;">
<col style="width: 300px;">
<col style="width: 100px;">
<col style="width: 300px;">
</colgroup>
<tr>
<th>이름</th>
<td></td>
<th>연락처</th>
<td></td>
</tr>
<tr>
<th>주소</th>
<td colspan="3"></td>
</tr>
<tr>
<th>자기소개</th>
<td colspan="3"></td>
</tr>
</table>

</body>
</html>

```



## (2) colspan, rowspan 속성 - 행 또는 열 합치기

### (2) <caption> 태그, <figcaption> 태그 - 표에 제목 붙이기

- <caption> 태그는 <table> 태그 바로 다음에 사용한다. 제목이 위쪽 중앙에 표시된다.
- <figcaption> 태그를 <table> 태그보다 앞에 사용하면 표 위에 제목이 표시되고 </table> 태그 다음에 추가하면 표 아래에 제목이 표시된다.
- aria-describedby 속성: 표에 대한 설명 제공하기

[02/table-3.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>표 만들기</title>
<style>
td, th {
    padding: 10px; /* 셀 테두리와 내용 사이의 간격(패딩) */
}
</style>
</head>
<body>
    <table border="1">
        <caption>
            <strong>Modern Browser</strong>
            <p>국내에서 자주 사용하는 모던 브라우저</p>
        </caption>
        <tr>
            <th>브라우저</th>
            <th>제조업체</th>
            <th>다운로드</th>
        </tr>
        <tr>
            <th>크롬(Chrome)</th>
            <td>Google</td>
            <td>https://www.google.com/chrome/</td>
        </tr>
        <tr>
            <th>파이어폭스(Firfox)</th>
            <td>Mozilla</td>
            <td>https://www.mozilla.org/ko/firefox/</td>
        </tr>
        <tr>
            <th>엣지(Edge)</th>
            <td>Microsoft</td>
            <td>https://www.microsoft.com/ko-kr/windows/microsoft-edge</td>
        </tr>
    </table>
</body>
</html>
```

[02/table-4.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>표 만들기</title>
```

```

<style>
td, th {
    padding: 10px; /* 셀 테두리와 내용 사이의 간격(패딩) */
}
</style>
</head>
<body>
    <figure>
        <figcaption>
            <p>
                국내에서 자주 사용하는 <b>모던 브라우저</b>
            </p>
        </figcaption>
        <table border="1">
            <tr>
                <th>브라우저</th>
                <th>제조회사</th>
                <th>다운로드</th>
            </tr>
            <tr>
                <th>크롬(Chrome)</th>
                <td>Google</td>
                <td>https://www.google.com/chrome/</td>
            </tr>
            <tr>
                <th>파이어폭스(Firfox)</th>
                <td>Mozilla</td>
                <td>https://www.mozilla.org/ko/firefox/</td>
            </tr>
            <tr>
                <th>엣지(Edge)</th>
                <td>Microsoft</td>
                <td>https://www.microsoft.com/ko-kr/windows/microsoft-edge</td>
            </tr>
        </table>
    </figure>
</body>
</html>

```

[02/table-5.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>표 만들기</title>
<style>
td, th {
    padding: 10px; /* 셀 테두리와 내용 사이의 간격(패딩) */
}
</style>
</head>
<body>
    <p id="summary">다음 표는 HTML5를 지원하는 모던(Modern Browser)을 정리한 것입니다. 최신
        버전일수록 HTML5를 좀더 많이 지원하기 때문에 최신 버전을 다운로드하는 것이
        좋습니다.</p>
    <table border="1" aria-describedby="summary">
        <caption>Modern Browser</caption>
        <tr>
            <th>브라우저</th>
            <th>제조회사</th>
            <th>다운로드</th>
        </tr>
        <tr>

```

```

        <th>크롬(Chrome)</th>
        <td>Google</td>
        <td>https://www.google.com/chrome/</td>
    </tr>
    <tr>
        <th>파이어폭스(Firfox)</th>
        <td>Mozilla</td>
        <td>https://www.mozilla.org/ko/firefox/</td>
    </tr>
    <tr>
        <th>엣지(Edge)</th>
        <td>Microsoft</td>
        <td>https://www.microsoft.com/ko-kr/windows/microsoft-edge</td>
    </tr>
</table>
</body>
</html>

```

### (3) <thead>, <tbody>, <tfoot> 태그 - 표 구조 정의하기

방 이름	대상	크기	가격	제목 부분 <thead>
유채방	여성 도미토리	4인실	1인 20,000원	본문 <tbody>
동백방	동성 도미토리			
	가족 1팀			
천혜향방	-	2인실		
바깥채 전체를 렌트합니다				바닥 부분 <tfoot>

- 기본형: <thead>
 <tr> ... </tr>
 </thead>
 <tbody>
 <tr> ... </tr>
 </tbody>
 <tfoot>
 <tr> ... </tr>
 </tfoot>

[02/table-6.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>표 만들기</title>
<style>
th, td {
    width: 80px; /* 셀의 너비 */
    padding: 10px; /* 셀 테두리와 내용 사이의 간격(패딩) */
}

thead, tfoot {
    background: #eeeeee; /* thead와 tfoot의 배경 색 */
}
</style>
</head>
<body>
    <table border="1">
        <caption>제주특별자치도 학교현황(2015.4.1 기준)</caption>
        <thead>

```

```

        <tr>
            <th>구분</th>
            <th>학교수</th>
            <th>학급수</th>
            <th>학생수</th>
            <th>교원수</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <th>유치원</th>
            <td>117</td>
            <td>252</td>
            <td>5,547</td>
            <td>474</td>
        </tr>
        <tr>
            <th>초등학교</th>
            <td>111</td>
            <td>1,720</td>
            <td>37,686</td>
            <td>2,632</td>
        </tr>
        <tr>
            <th>중학교</th>
            <td>44</td>
            <td>699</td>
            <td>21,274</td>
            <td>1,412</td>
        </tr>
        <tr>
            <th>고등학교</th>
            <td>29</td>
            <td>676</td>
            <td>22,019</td>
            <td>1,433</td>
        </tr>
        <tr>
            <th>특수학교</th>
            <td>3</td>
            <td>90</td>
            <td>428</td>
            <td>160</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <th>합계</th>
            <td>300</td>
            <td>3,437</td>
            <td>86,954</td>
            <td>6,111</td>
        </tr>
    </tfoot>
</table>
</body>
</html>

```

#### (4) <col>, <colgroup> 태그 - 여러 열 묶어 스타일 지정하기

- <col> 태그: 한 열에 있는 모든 셀을 묶는다. 닫는 태그는 없다.
- <colgroup> 태그: 여러 개의 <col> 태그를 묶어 그룹으로 스타일을 적용한다.
- 기본형: <colgroup>

```

        <col>
        ...
    </colgroup>

```

[02/colgroup.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>표 만들기</title>
<style>
td {
    width: 100px; /* 셀 너비 */
    height: 30px; /* 셀 높이 */
}
</style>
</head>
<body>

    <table border="1">
        <caption>colgroup 연습</caption>
        <colgroup>
            <col>
            <col span="2" style="background-color: blue;">
            <col style="background-color: yellow">
        </colgroup>
        <tr>
            <td></td>
            <td></td>
            <td></td>
            <td></td>
        </tr>
        <tr>
            <td></td>
            <td></td>
            <td></td>
            <td></td>
        </tr>
    </table>

</body>
</html>

```

## [실습] 온라인 프로필 - 표 추가하기

p93 ~ p95

## [과제] 연습문제

1. 텍스트들이 미리 입력되어 있는 quiz-1.html 문서를 열고 다음 조건에 맞게 태그를 추가하라.

조건:

첫 번째 줄 텍스트는 1단계 제목으로 바꾼다.

제목 이하 텍스트는 3개 단락으로 나눈다.

- 첫 번째 단락: 내용을 강조하기 위해 굵게 표시한다.
- 두 번째 단락: 이탤릭체로 표시한다. (강조의 의미는 없음)
- 세 번째 단락: 읽기 쉽게 적당히 줄을 바꾸어 표시한다.



## 3 이미지와 하이퍼링크

### 3.1. 이미지

#### (1) 웹에서 사용하는 이미지 형식

- 웹 페이지에서 사용할 수 있는 이미지 파일은 파일 크기가 크지 않으면서도 화질은 좋게 유지해야 하기 때문에 몇 가지 파일 형식만 사용할 수 있다.
- 웹에서 사용할 수 있는 이미지 파일 형식

GIF(Graphic Interchange Form)	JPG나 PNG 형식에 비해 파일 크기가 작지만 표시할 수 있는 색상 수가 최대 256가지뿐으로, 웹 페이지에서 아이콘이나 볼릿 등 작은 이미지에 주로 사용. 투명한 배경이나 움직이는 이미지를 만들 수 있다.
JPG 또는 JPEG (Joint Photographic Experts Group)	사진을 웹 페이지에 넣기 위해 개발된 형식이기 때문에 GIF에 비해 다양한 색상과 명암을 표현할 수 있다. 디지털 카메라에서 저장하는 사진 파일은 대부분 JPG 형식이다.
PNG(Portable Network Graphics)	투명 배경을 만들 수도 있고 사물이 가진 색상도 최대한 그대로 유지되기 때문에 최근에 많이 사용되는 이미지 파일 형식이다.

#### (2) <img> 태그 - 이미지 삽입하기

- 기본형: 

[03/img-1.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>추천 관광지</title>
</head>
<body>
    <h1>사려니 숲길</h1>
    
</body>
</html>
```

#### (3) alt 속성 - 이미지를 설명해 주는 대체 텍스트

- 연결 속도가 느리거나 이미지가 제대로 표시할 수 없는 상황에서 이미지 자리에 내용이 표시된다.  
예) 

#### (4) width, height 속성 - 이미지 크기 조정하기

[03/img-2.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>추천 관광지</title>
</head>
<body>
    <h1>사려니 숲길</h1>
    
    
</body>
</html>
```

### (5) <figure>, <figcaption> 태그 - 이미지에 설명 글 붙이기

- <figure> 태그는 캡션을 붙일 대상을 지정한다(예, 이미지, 동영상 등). 여러 개의 미디어를 하나의 <figure> 태그로 묶을 수도 있다.
- <figcaption> 태그는 <figure> 태그로 묶은 미디어에 캡션을 붙인다.

[03/figcaption.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>추천 관광지</title>
</head>
<body>
    <h1>제주의 오름</h1>
    <p>거인 설문대할망이 제주도와 육지 사이에 다리를 놓으려고 치마폭에 흙을 담아 나를 때 치마
    틈새로 한줌씩 떨어진
        흙덩이들이 오름이 되었다는 전설이 있다.</p>
    <figure>
        
        <figcaption>완만하고 부드러운 용눈이 오름</figcaption>
    </figure>
</body>
</html>
```

### [실습] 온라인 프로필 - 이미지 삽입하기

p109 ~ p110

## 3.2. 링크 만들기

### (1) <a> 태그, href 속성 - 링크 만들기

- 기본형: <a href="연결할 문서나 사이트 경로">텍스트</a>
- 링크를 만드는 태그는 <a> 태그
- 반드시 href 속성을 함께 사용해서 어떤 대상으로 연결하는지 알려주어야 한다.
- 텍스트 링크의 밑줄과 글자 색 바꾸기: CSS를 이용해 텍스트 링크의 색을 바꾸고 밑줄을 없앨 수 있다.

예) <style>

```
    a {
        text-decoration:none;
        color:black;
    }
</style>
```

[03/link.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>링크 만들기</title>
</head>
<body>
    <h1>텍스트 링크 만들기</h1>
    <a href="http://www.easyspub.com">이지스퍼블리싱 홈페이지</a>
    <h1>이미지 링크 만들기</h1>
    <a href="http://www.easyspub.com"> 
    </a>
```



```
</body>
</html>
```

## (2) target 속성 - 새 탭에서 링크 열기

- 다른 사이트로 링크하거나 현재 페이지를 유지한 상태에서 링크 페이지를 표시할 때
- 현재 페이지는 그대로 유지하면서 새 창이나 새 탭에 표시
- <a> 태그의 target 속성: \_blank, \_self, \_parent, \_top
- 기본형: <a href="링크할 경로" target="\_blank">

[03/target.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>텍스트 링크</title>
<style>
a {
    text-decoration: none;
    color: black;
}
</style>
</head>
<body>
    <h1>텍스트 링크 만들기</h1>
    <p>
        <a href="http://www.easyspub.com">이시스퍼블리싱 홈페이지(현재 화면)</a>
    </p>
    <p>
        <a href="http://www.easyspub.com" target="_blank">이시스퍼블리싱 홈페이지(새 창
            또는 새 탭)</a>
    </p>
</body>
</html>
```

## (3) 한 페이지 안에서 점프하는 앵커 만들기

- 웹 문서가 너무 길 경우 문서 안에 팻말을 달아놓고 그 위치로 한번에 이동하는 기능을 앵커(anchor)라고 한다.
- 한 페이지의 내용이 너무 길 경우, 그리고 서로 구분될 수 있는 내용으로 구성되어 있을 경우 사용하면 편리하다.
- 기본형: <태그 id="앵커 이름"> 텍스트 또는 이미지 </a>  
<a href="#앵커 이름"> 앵커로 이동 </a>

[03/anchor.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>링크 만들기</title>
</head>
<body>
    <h1>앵커 만들기</h1>
    <p>웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한번에 이동하는
        링크를 만들 수 있는데, 이 기능을 앵커(anchor)라고 합니다.</p>
    <ul id="menu">
        <li><a href="#content1">메뉴1</a></li>
```

```

        <li><a href="#content2">메뉴2</a></li>
        <li><a href="#content3">메뉴3</a></li>
    </ul>
    <h2 id="content1">내용1</h2>
    <p>웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한번에 이동하는
링크를 만들 수
        있는데, 이 기능을 앵커(anchor)라고 합니다.</p>
    <p>웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한번에 이동하는
링크를 만들 수
        있는데, 이 기능을 앵커(anchor)라고 합니다.</p>
    <p>웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한번에 이동하는
링크를 만들 수
        있는데, 이 기능을 앵커(anchor)라고 합니다.</p>
    <p>
        <a href="#menu">[메뉴로]</a>
    </p>
    <h2 id="content2">내용2</h2>
    <p>웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한번에 이동하는
링크를 만들 수
        있는데, 이 기능을 앵커(anchor)라고 합니다.</p>
    <p>웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한번에 이동하는
링크를 만들 수
        있는데, 이 기능을 앵커(anchor)라고 합니다.</p>
    <p>웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한번에 이동하는
링크를 만들 수
        있는데, 이 기능을 앵커(anchor)라고 합니다.</p>
    <p>
        <a href="#menu">[메뉴로]</a>
    </p>
    <h2 id="content3">내용3</h2>
    <p>웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한번에 이동하는
링크를 만들 수
        있는데, 이 기능을 앵커(anchor)라고 합니다.</p>
    <p>웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한번에 이동하는
링크를 만들 수
        있는데, 이 기능을 앵커(anchor)라고 합니다.</p>
    <p>웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한번에 이동하는
링크를 만들 수
        있는데, 이 기능을 앵커(anchor)라고 합니다.</p>
    <p>
        <a href="#menu">[메뉴로]</a>
    </p>
</body>
</html>

```

## [실습] 온라인 프로필 - 링크 추가하기

p117 ~ p119

## [실습] 인터넷에 내 온라인 프로필 올리기

p119 ~ p122

### 3.3. SVG 이미지

#### (1) SVG 파일 형식이란?

- 비트맵 이미지(bitmap image)는 확대하면 이미지 테두리 부분이 울퉁불퉁해진다.  
(jpg, png)
- 벡터 이미지(vector image)는 아무리 확대하거나 축소해도 원래의 깨끗한 상태 그대로 유지된다.
- SVG(Scalable Vector Graphics) 파일 형식은 자연스럽게 크기 조절이 가능하다.
- SVG는 이미지이면서도 소스 코딩을 통해 만들고 편집할 수 있다.

[03/images/automobile.svg] 편집기에서 확인한 svg 파일

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg style="shape-rendering:geometricPrecision;text-rendering:geometricPrecision;image-
rendering:optimizeQuality" xmlns="http://www.w3.org/2000/svg" fill-rule="evenodd" xml:space="preserve"
height="100mm" width="200mm" version="1.1" clip-rule="evenodd" xmlns:xlink="http://www.w3.org/1999/xlink"
viewBox="0 0 20000 10000">
  <defs>
    <style type="text/css">
      <![CDATA[
        .str0 {stroke:#373435;stroke-width:20}
        .fil1 {fill:#4B4B4D}
        .fil2 {fill:#373435}
        .fil4 {fill:#ED3237}
        .fil0 {fill:#201E1E}
        .fil9 {fill:url(#id0)}
        .fil6 {fill:url(#id1)}
        .fil5 {fill:url(#id2)}
        .fil3 {fill:url(#id3)}
        .fil7 {fill:url(#id4)}
        .fil8 {fill:url(#id5)}
      ]]>
    </style>
    <linearGradient id="id0" y2="5400.8" gradientUnits="userSpaceOnUse" x2="1558.6" y1="4706.8"
x1="1572.9">
      ...
    </svg>
```

## (2) SVG 이미지 삽입하기

[03/insert-svg.html]

```
<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>Insert SVG</title>
</head>
<body>
  <h1>SVG 이미지 삽입하기</h1>
  
</body>
</html>
```

## (3) SVG 파일을 지원하지 않는 브라우저를 위해

- SVG 이미지 파일 대신 PNG 이미지 파일을 사용해야 한다.
- 사용자의 브라우저가 SVG 파일을 지원여부 테스트

[03/check-svg.html]

```
<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>Insert SVG</title>
<script src="modernizr-custom.js"></script>
</head>
<body>
  <h1>SVG 이미지 삽입하기</h1>
  
```

```
<script>
    if (!modernizr.svg) {
        $("img").attr("src", "images/muffin.png");
    }
</script>
</body>
</html>
```

## [과제] 연습문제

1. quiz-1.html 문서에서 '이미지 삽입하기' 제목 아래 images 폴더에 있는 umbrella.jpg 이미지를 삽입하라. 이때 이미지 크기가 크기 때문에 실제로 화면에 표시할 때는 320\*210 크기로 줄여 사용하라.

## 4 폼과 관련된 태그들

### 4.1 폼 만들기

#### (1) 웹에서 자주 만나는 폼



#### (2) <form> 태그 - 폼 만들기

- 기본형: <form [속성= "속성값" ]> 여러 폼 요소 </form>
- 사용할 수 있는 속성
  - method : 폼을 전송할 방식 선택
  - name : 폼을 식별하기 위한 폼의 이름
  - action : 폼을 전송할 서버 쪽의 스크립트 파일
  - target : action에서 지정한 스크립트 파일을 현재 창이 아닌 다른 위치에 열도록 지정
- autocomplete 속성: 사용자가 입력했던 내용을 기억했다가 비슷한 내용을 입력할 경우, 이전에 입력했던 내용을 힌트로 보여 준다.

[04/form.html]

```
<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>HTML5 폼</title>
</head>
<body>
    <form action="search.php" method="post">
        <input type="text" title="검색"> <input type="submit" value="검색">
    </form>
</body>
</html>
```

#### (3) <label> 태그 - 폼 요소에 레이블 붙이기

- 폼 요소에 레이블(텍스트)을 붙이는 태그
- 라디오 버튼이나 체크 박스에서 레이블 부분을 클릭해도 라디오 버튼과 체크 박스 버튼이 선택된다.
- 기본형: <label [속성="속성값"]> 레이블 <input ...> </label> 또는  
 <label for="id이름"> 레이블 </label>  
 <input id="id이름" [속성="속성값"]>
- 라디오 버튼이나 체크 박스와 연결해 놓았다면 텍스트만 터치해도 선택되어 사용이 훨씬 쉬워진다.

[04/withlabel.html]

```

<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>label 태그 사용</title>
<style>
ul {
    list-style: none;
}

li {
    margin: 20px;
}
</style>
</head>
<body>
    <form>
        <h3>수강 분야(다수 선택 가능)</h3>
        <ul>
            <li><input type="checkbox" value="grm">문법</li>
            <li><input type="checkbox" value="wr">작문</li>
            <li><input type="checkbox" value="rd">독해</li>
        </ul>
        <h3>수강 과목(1과목만 선택 가능)</h3>
        <ul>
            <li><label><input type="radio" name="subject"
                value="eng">영어회화</label></li>
            <li><label><input type="radio" name="subject" value="ch">중국어회화</label>
            </li>
            <li><label><input type="radio" name="subject" value="jp">일어회화</label>
            </li>
        </ul>
    </form>
</body>
</html>

```

#### (4) <fieldset>, <legend> 태그 - 품 요소 그룹으로 묶기

- <fieldset> 태그 : 품 요소를 그룹으로 묶는 태그
- <legend> 태그 : 그룹으로 묶은 구역에 제목을 붙이는 태그

[04/fieldset.html]

```

<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>fieldset과 legend 태그</title>
<style>
ul {
    list-style: none;
}

li {
    margin: 20px;
}

li label {
    width: 80px;
    float: left;
}

```

```

fieldset {
    margin: 15px;
}
</style>
</head>
<body>
    <form>
        <fieldset>
            <legend>개인 정보</legend>
            <ul>
                <li><label for="name">이름</label> <input type="text" id="name">
                </li>
                <li><label for="mail">메일 주소</label> <input type="text"
                    id="mail"></li>
            </ul>
        </fieldset>
        <fieldset>
            <legend>로그인 정보</legend>
            <ul>
                <li><label for="id">아이디</label> <input type="text" id="id">
                </li>
                <li><label for="pwd">비밀번호</label> <input type="text"
                    id="pwd">
                </li>
            </ul>
        </fieldset>
    </form>
</body>
</html>

```

## 4.2 사용자가 정보를 입력하는 <input> 태그

### (1) <input> 태그 살펴보기

- 폼에서 사용자 입력을 받기 위해 사용한 태그이다.

### (2) <input> 태그 - 입력 항목 만들기

- 기본형: <input type=" 유형 " [속성="속성값"]>
- id 속성은 여러 번 사용된 폼 요소를 구분하기 위해 사용한다.  
예) <input type="text" id="user-name" size="10">  
      <input type="text" id="addr" size="60">
- type 속성에서 사용 가능한 유형: hidden, text, search, tel, url 등

### (3) type="hidden" - 히든 필드 만들기

- 화면상의 폼에는 보이지 않지만 사용자가 입력을 마치고 폼을 서버로 전송할 때 서버로 함께 전송되는 요소이다.
- 기본형: <input type="hidden" name="이름" value="서버로 넘길 값">

### (4) type="text" - 텍스트 필드 만들기

- 한 줄짜리 일반 텍스트를 입력하는 필드이다.
- 기본형: <input type="text" [속성="속성값"]>

### (5) type="password" - 비밀번호 입력란 만들기

- 비밀번호 입력란을 만든다. 사용자가 입력하는 내용이 화면에 표시되지 않고 ‘\*’로 표시된다.
- 기본형: <input type="password" [속성="속성값"]>

```

<!doctype html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>웹 폼</title>
<style>
label {
    font-size: 14px; /* 글자 크기 */
    line-height: 25px; /* 줄 간격 */
}

label em {
    color: red; /* 글자색 */
    font-weight: 800; /* 글자 굵기 */
}
</style>
</head>
<body>
    <form>
        <fieldset>
            <label>아이디: <input type="text" id="user_id" size="10"></label>
            <label>비밀번호: <input type="password" id="user_pw" size="10"></label>
            <input type="submit" value="로그인">
        </fieldset>
    </form>
</body>
</html>

```

## (6) type="search", type="url", type="email", type="tel" - 분화된 테스트 필드

- type="search"을 사용했을 때, 일부 브라우저에서는 검색 창에 검색어를 입력했을 때 오른쪽에 x가 표시되어 입력했던 검색어를 손쉽게 지울 수 있다.
- type="url"을 사용했을 때, 이 필드에는 반드시 'http://'로 시작하는 사이트 주소를 입력해야 한다.
- type="email"로 지정만 하면 브라우저 자체에서 사용자가 입력한 내용이 메일 주소 형식(@문자가 들어 있는지 체크)에 맞는지 자동으로 체크해 준다.
- type="tel"로 지정하면 사용자가 입력한 정보가 일반 텍스트가 아니라 전화번호라는 사실을 인식한다.

[04/url-email-tel.html]

```

<!doctype html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>웹 폼</title>
<style>
ul {
    list-style: none;
}

li {
    margin: 10px;
}

li label {
    width: 120px;
    float: left;
}

```



```

        text-align: right;
        padding-right: 8px;
    }

    input[type="submit"] {
        text-align: center;
        width: 100%;
        height: 30px;
        margin-top: 15px;
    }
</style>
</head>
<body>
    <h1>회원 가입</h1>
    <form>
        <fieldset>
            <legend>로그인 정보</legend>
            <ul>
                <li><label for="user-id">아이디 </label> <input type="text"
                    id="user-id"></li>
                <li><label for="pwd1">비밀번호 </label> <input type="password"
                    id="pwd1"></li>
                <li><label for="pwd2">비밀번호 확인 </label> <input
                    type="password"
                    id="pwd2"></li>
            </ul>
        </fieldset>
        <fieldset>
            <legend>개인 정보</legend>
            <ul>
                <li><label for="user-name">이름 </label> <input type="text"
                    id="user-name"></li>
                <li><label for="mail">메일 주소</label> <input type="email"
                    id="mail"></li>
                <li><label for="phone">연락처</label> <input type="tel"
                    id="phone"></li>
                <li><label for="homep">블로그/홈페이지</label> <input type="url"
                    id="homep"></li>
            </ul>
        </fieldset>
        <input type="submit" value="가입하기">
    </form>
</body>
</html>

```

## (7) type="number" - 숫자 입력하기

- 입력 창에 숫자를 직접 입력하는 것이 기본이지만 일부 브라우저(파이어폭스)에 따라 스피너 박스가 표시되어 화살표를 클릭으로 숫자를 증감시킬 수 있다.

## (8) type="range" - 슬라이드 막대로 숫자 지정하기

- 슬라이드 막대를 움직여 숫자 값을 입력하게 한다.

[04/number-range.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>웹 품</title>
<style>
body {

```

```

        background-color: #fff;
    }

    form legend {
        font-size: 15px;
        font-weight: 600;
    }

    form label.reg {
        font-size: 14px;
        width: 150px;
        float: left;
    }

    form ul li {
        list-style: none;
        margin: 15px 0;
    }

    #member, #stuffs {
        width: 50px;
    }
</style>
</head>
<body>
    <form>
        <fieldset>
            <legend> 등록 정보</legend>
            <ul>
                <li><label class="reg"
for="member">참여인원<small>(최대10명)</small></label>
                <input type="number" id="member" value="1" min="0"
max="10"
                step="1"></li>
                <li><label class="reg" for="stuffs">지원물품<small>(1인당
5개)</small></label> <input
type="number" id="stuffs" value="1" min="0" max="50"
                step="5" size="5"></li>
                <li><label class="reg" for="satis">희망
단계<small>(하,중,상)</small></label>
                <input type="range" id="satis" value="1" min="1"
max="3"></li>
            </ul>
        </fieldset>
    </form>
</body>
</html>

```

## (9) type="radio", type="checkbox" - 라디오 버튼과 체크박스 넣기

- 원하는 항목을 선택할 때 사용하는 폼 요소이다.
- 한 개만 선택하도록 할 경우에 라디오 버튼을 사용하고, 두 개 이상 여러 가지를 선택해도 될 경우에 체크박스를 사용한다.
- 기본형: <input type="radio" [속성="속성값"]>  
<input type="checkbox" [속성="속성값"]>

[04/radio-checkbox.html]

```

<!doctype html>
<html lang="ko">
<head>
    <meta charset="utf-8">
    <title> 웹 폼</title>

```

```

<style>
  body {
    background-color:#fff;
  }
  form fieldset{
    margin-bottom:25px;
  }
  form legend{
    font-size:15px;
    font-weight:600;
  }
  form label.reg {
    font-size:14px;
    color:#390;
    font-weight:bold;
    width:110px;
    float:left;
    text-align:right;
    margin-right:10px;
  }
  form ul li{
    list-style:none;
    margin: 15px 0;
    font-size:14px;
  }

  #member, #stuffs {
    width:50px;
  }

  #pre {
    margin-left:15px;
  }
</style>
</head>
<body>
  <form>
    <fieldset>
      <legend>신청 과목</legend>
      <p>이 달에 신청할 과목을 선택하세요 (1과목만 가능)</p>
      <label><input type="radio" name="subject" value="speaking">회화</label>
      <label><input type="radio" name="subject" value="grammar">문법</label>
      <label><input type="radio" name="subject" value="writing">작문</label>
    </fieldset>
    <fieldset>
      <legend>메일링</legend>
      <p>메일로 받고 싶은 뉴스 주제를 선택해 주세요 (복수 선택 가능)</p>
      <label><input type="checkbox" name="mailing1" value="news">해외
      단신</label>
      <label><input type="checkbox" name="mailing2" value="dialog">5분
      회화</label>
      <label><input type="checkbox" name="mailing3" value="pops">모닝팝스</label>
    </fieldset>
  </form>
</body>
</html>

```

## (10) type="color" - 색상 선택 상자 표시하기

- 색상표에서 사용자가 색상을 선택할 수 있게 해준다.
- 기본형: <input type="color" value="기본색">

```

<!doctype html>
<html lang="ko">
  <head>
    <meta charset="utf-8">
    <title> 웹 폼</title>
  </head>
  <body>
    <form>
      <fieldset>
        <legend>과 티셔츠 설문</legend>
        <p>올해 과 티(T)를 만들려고 합니다. 원하는 색상을 추천해 주세요.</p>
        <label>선호색상 <input type="color" value="#00ff00"> </label>
      </fieldset>
    </form>
  </body>
</html>

```

## (11) type="date", type="month", type="week" - 날짜 표시하기

- type="date"로 지정하면 달력에서 날짜를 선택했을 때 필드에 "yyyy-mm-dd" 형식으로 연도와 월, 일이 표시된다.

```

[04/date.html]

<!DOCTYPE html>

<html lang="ko">
<head>
  <meta charset="utf-8">
  <title>date</title>
</head>
<body>
  <form>
    <h3>조회기간 선택</h3>
    <label><input type="date" id="start"></label>
    <label><input type="date" id="end"></label>
  </form>
</body>
</html>

```

## (12) type="time", type="datetime", type="datetime-local" - 시간 지정하기

- 날짜뿐만 아니라 시간도 지정할 수 있다.

```

[04/time.html]

<!DOCTYPE html>

<html lang="ko">
<head>
  <meta charset="utf-8">
  <title>time 속성</title>
</head>
<body>
  <form>
    <h3>대관시간을 선택하세요(오늘)</h3>
    <label>시작 시간<input type="time" value="09:00" id="start"></label>,
    <label>종료 시간<input type="time" value="18:00" id="end"></label>

    <h3>대관시간을 선택하세요(다른날짜)</h3>
    <label>시작 시간<input type="datetime-local"

```

```

        value="2016-03-02T09:00" id="start"></label>, <label>종료 시간<input
        type="datetime-local" value="2016-03-02T18:00" id="end"></label>

    </form>
</body>
</html>

```

### (13) type="submit", type="reset" - 서버 전송, 리셋 버튼 넣기

- type="reset"은 <input> 요소에 입력된 모든 정보를 지울 수 있다.
- type="submit"은 사용자가 폼에 입력한 정보를 서버로 전송하는 submit 버튼을 넣는다.

[04/submit-reset.html]

```

<!DOCTYPE html>

<html lang="ko">
<head>
    <meta charset="utf-8">
    <title>전송 - 리셋 </title>
    <style>
        body {
            padding: 20px;
        }
    </style>
</head>
<body>
    <h3>메일링 리스트 등록</h3>
    <form action="register.php" method="post">
        <label> 메일 주소 <input type="text"></label>
        <input type="submit" value="제출">
        <input type="reset" value="다시입력">
    </form>
</body>
</html>

```

### (14) type="image" - 이미지 버튼 넣기

- submit 버튼 대신 전송 이미지를 넣을 수 있다.
- 기본형: <input type="image" src="경로" alt="대체 텍스트">

[04/submit-image.html]

```

<!DOCTYPE html>

<html lang="ko">
<head>
    <meta charset="utf-8">
    <title>이미지 버튼 </title>
    <style>

    </style>
</head>
<body>
    <form>
        <label>아이디 <input type="text" size="15"></label>
        <label>비밀번호 <input type="password" size="15"></label>
        <input type="image" id="butt" src="images/login.jpg" alt="login">
    </form>
</body>
</html>

```

### (15) type="button" - 버튼 넣기

- 폼 안에 버튼 형태를 만든다. 이 버튼은 submit이나 reset 같은 자체 기능이 없고 오직 버튼만 넣기 때문에 스크립트 함수 등을 연결해 사용한다.

[04/button.html]

```
<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>폼 버튼</title>
</head>
<body>

    <form>
        <input type="button" value="새 탭 열기" onclick="window.open()">
    </form>

</body>
</html>
```

### (16) type="file" - 파일 첨부하기

- 웹 브라우저 화면에 [파일 선택]이나 [찾아보기] 등이 표시되는데 이 버튼을 클릭하여 파일을 선택하면 파일이 첨부된다.
- 기본형: <input type="file" [속성="속성값"]>

## 4.3 <input> 태그의 다양한 속성

### (1) autofocus 속성 - 입력 커서 표시하기

- 페이지를 불러오자마자 원하는 요소에 마우스 커서가 표시되도록 하는 것
- 예전에는 자바스크립트를 이용해야 했는데, HTML5에서는 autofocus라는 속성으로 쉽게 해결할 수 있다.

### (2) placeholder 속성 - 힌트 표시하기

- <input> 요소의 필드 안에 적당한 힌트 내용을 표시하고 있다가 그 필드를 클릭하면 내용이 사라지도록 하는 것
- 텍스트 필드 앞에 레이블을 사용하지 않고도 어떤 내용을 입력해야 할지 알려줄 수 있어 편리하다.

### (3) readonly 속성 - 읽기 전용 필드 만들기

- readonly 속성은 해당 필드를 읽기 전용으로 바꾼다.
- 필드 안에 내용이 있으면서 사용자에게 내용을 보여주지만 하고 사용자가 입력은 할 수 없게 한다.
- 읽기 전용으로 하려면 readonly="true" 또는 readonly

### (4) required 속성 - 필수 필드 지정하기

- [submit] 단추를 클릭해서 서버로 폼을 전송할 때 필수 필드에 필요한 내용이 모두 채워졌는지 검사한다. → HTML4에서는 자바스크립트로 처리했지만 HTML5에서는 required 속성 사용

### (5) min, max, step 속성

- min 속성과 max 속성은 각각 해당 필드의 최소값과 최대값을 지정한다.
- step 속성은 허용된 범위 내의 숫자의 일정한 간격을 가리킨다.

[04/attribute.html]

```

<!doctype html>
<html lang="ko">
<head>
<title>웹 폼</title>
<style>
body {
    background-color: #fff;
}

form fieldset {
    margin-bottom: 25px;
}

form legend {
    font-size: 15px;
    font-weight: 600;
}

form label.reg {
    font-size: 14px;
    width: 110px;
    color: #390;
    font-weight: bold;
    float: left;
    text-align: right;
    margin-right: 10px;
}

form ul li {
    list-style: none;
    margin: 15px 0;
    font-size: 14px;
}

#member, #stuffs {
    width: 50px;
}
</style>
</head>
<body>
    <h1>여름방학 특강 신청</h1>
    <form>
        <fieldset id="subject">
            <legend>수강과목</legend>
            <ul>
                <li><label class="reg" for="subj">영어회화(초급)</label> <input
type="text" id="subj" value="오전 9:00~11:00" readonly></li>
            </ul>
        </fieldset>
        <fieldset id="register">
            <legend>신청자</legend>
            <ul>
                <li><label class="reg" for="uname">이름</label> <input
type="text" id="uname" autofocus required></li>
                <li><label class="reg" for="uid">학번</label> <input type="text"
id="uid" placeholder="하이픈없이 입력" maxlength="8" required></li>
                <li><label class="reg" for="uclass">학과</label> <select
id="uclass">
                    <option value="archi">건축공학과</option>
                    <option value="mechanic">기계공학과</option>
                    <option value="indust">산업공학과</option>
                    <option value="elec">전기전자공학과</option>
                    <option value="computer"
selected>컴퓨터공학과</option>
                </select>
            </ul>
        </fieldset>
    </form>

```

```

                                <option value="chemical">화학공학과</option>
                                </select></li>
                            </ul>
                        </fieldset>
                    </fieldset>
                    <legend>교재 주문</legend>
                    <ul>
                        <li><label class="reg" for="book">교재</label> <input
type="number" id="book" value="1" min="1" max="3"></li>
                        <li><label class="reg" for="wsheet">워크시트</label> <input
type="number" id="wsheet" value="1" min="1" max="3"></li>
                        <li><label class="reg" for="group">단체주문</label> <input
type="number" id="group" value="10" min="10" max="100" step="10">
                        </li>
                    </ul>
                </fieldset>
            </fieldset>
            <button type="submit" value="submit">신청하기</button>
            <button type="reset" value="reset">다시쓰기</button>
        </fieldset>

    </form>
</body>
</html>

```

## (6) size, minlength, maxLength 속성 - 길이, 최소 길이, 최대 길이 속성

[04/join.html]

```

<!doctype html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>웹 폼</title>
<style>
ul {
    list-style: none;
}
li {
    margin: 10px;
}
</style>
</head>
<body>

    <h1>회원 가입을 환영합니다</h1>
    <form>
        <ul>
            <li><label>아이디: <input type="text" id="user_id"
size="10" minlength="4" maxlength="15"></label><small
style="color: red;"> 4~15자리 이내의 영문과 숫자</small></li>
            <li><label>이메일: <input type="email" id="user_email"></label></li>
            <li><label>비밀번호: <input type="password" id="user_pwd"></label></li>
            <li><label>비밀번호 확인: <input type="password"
id="check_pwd"></label></li>
            <li><input type="submit" value="회원가입"></li>
        </ul>
    </form>

</body>
</html>

```



## [실습] 상품 주문서 작성하기

p174 ~ p178

### 4.4 여러 데이터 나열에서 보여주기

#### (1) <select>, <optgroup>, <option> 태그 - 드롭다운 목록 만들기

- 최대한 공간을 적게 사용하면서 여러 가지 옵션을 표시하는 드롭다운 목록 만들기
- 기본형: <select 속성="속성값">  
    <option value="값" [속성="속성값"]> 내용1 </option>  
    <option value="값" [속성="속성값"]> 내용2 </option>  
    <option value="값" [속성="속성값"]> 내용3 </option>  
    ...  
    </select>
- <select> 태그의 속성: size 속성이나 multiple 속성을 이용하면 드롭다운 목록의 크기나 선택할 항목의 개수를 조절할 수 있다.
- <option> 태그의 속성: value 속성, selected 속성
- <optgroup> 태그: 옵션끼리 묶기

[04/select1.html]

```
<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>웹 폼</title>
<style>
body {
    background-color: #fff;
}

form fieldset {
    margin-bottom: 25px;
}

form legend {
    font-size: 15px;
    font-weight: 600;
}

.reg {
    font-size: 14px;
    width: 110px;
    color: #390;
    font-weight: bold;
    float: left;
    text-align: right;
    margin-right: 10px;
}

form ul li {
    list-style: none;
    margin: 15px 0;
    font-size: 14px;
}
</style>
</head>
<body>
    <h1>여름방학 특강 신청</h1>
    <form>
        <fieldset>
            <legend>수강 신청인</legend>
```

```

        <ul>
            <li><label class="reg" for="id">학번</label> <input type="text"
                id="id" autofocus></li>
            <li><label class="reg" for="name">이름</label> <input type="text"
                id="name"></li>
            <li><label class="reg" for="class">학과</label> <select
                id="class">
                    <option value="archi">건축공학과</option>
                    <option value="mechanic">기계공학과</option>
                    <option value="indust">산업공학과</option>
                    <option value="elec">전기전자공학과</option>
                    <option value="computer">컴퓨터공학과</option>
                    <option value="chemical">화학공학과</option>
                </select></li>
        </ul>
    </fieldset>
</form>
</body>
</html>

```

## (2) <datalist>, <option> 태그

- 텍스트 필드에 입력할 수 있는 값들을 목록 형태로 제시
- 텍스트 필드에 직접 값을 입력하는 것이 아니라 제시한 값 중에서 선택하면 자동으로 입력됨
- 기본형: <input type="text" list="데이터 목록 id">

```

    <datalist id="데이터 목록 id">
        <option> ... </option>
        <option> ... </option>
        <option> ... </option>
        ...
    </datalist>

```

[04/datalist.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>웹 폼</title>
<style>
body {
    background-color: #fff;
}

form fieldset {
    margin-bottom: 25px;
}

form legend {
    font-size: 15px;
    font-weight: 600;
}

.reg {
    font-size: 14px;
    width: 110px;
    color: #390;
    font-weight: bold;
    float: left;
    text-align: right;
}

```

```

        margin-right: 10px;
    }

    form ul li {
        list-style: none;
        margin: 15px 0;
        font-size: 14px;
    }
</style>
</head>
<body>
    <h1>여름방학 특강 신청</h1>
    <form>
        <fieldset>
            <legend>수강 신청인</legend>
            <ul>
                <li><label class="reg" for="id">학번</label> <input type="text"
                    id="id" autofocus></li>
                <li><label class="reg" for="name">이름</label> <input type="text"
                    id="name"></li>
                <li><label class="reg" for="class">학과</label> <select
                    id="class">
                        <optgroup label="공과대학">
                            <option
                                value="archi">건축공학과</option>
                            <option
                                value="mechanic">기계공학과</option>
                            <option
                                value="indust">산업공학과</option>
                            <option
                                value="elec">전기전자공학과</option>
                            <option
                                value="computer">컴퓨터공학과</option>
                            <option
                                value="chemical">화학공학과</option>
                        </optgroup>
                        <optgroup label="인문대학">
                            <option
                                value="history">사학과</option>
                            <option
                                value="lang">어문학부</option>
                            <option value="philo">철학</option>
                        </optgroup>
                    </select></li>
            </ul>
        </fieldset>
        <fieldset>
            <legend>수강 과목을 선택하세요</legend>
            <ul>
                <li><span class="reg">관심분야</span> <label
                    for="interest"></label>
                    <input type="text" id="interest" list="choices">
                    <datalist
                        id="choices">
                            <option value="grammar">
                                label="문법"></option>
                            <option value="voca" label="어휘"></option>
                            <option value="speaking">
                                label="회화"></option>
                            <option value="listening">
                                label="리스닝"></option>
                            <option value="news">
                                label="뉴스청취"></option>
                        </datalist></li>
            </ul>
        </fieldset>
    </form>
</body>

```

```

        </fieldset>
    </form>
</body>
</html>

```

### (3) <textarea> 태그 - 여러 줄 입력하는 텍스트 영역 만들기

- 한 줄 이상의 문장을 입력할 때 사용하는 폼
- 기본형: <textarea [속성="속성값"]> 내용 </textarea>

[04/textarea.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>웹 폼</title>
<style>
form fieldset {
    margin: 10px 0;
}

form legend {
    font-size: 18px;
    color: #0066FF;
    font-weight: 600;
}

form label.reg {
    font-size: 14px;
    width: 120px;
    float: left;
}

form label em {
    font-size: 15px;
    color: red;
    font-weight: 800;
}

form ul li {
    list-style: none;
    margin: 10px 0;
}

form .easys {
    text-align: left;
    font-size: 12px;
    color: blue;
}

form fieldset.sendform {
    text-align: center;
}
</style>
</head>
<body>
    <form>
        <fieldset class="login">
            <legend>로그인</legend>
            <label for="user_id">아이디 </label> <input type="text" id="user_id"
                size="10" autofocus> <label for="user_pw">비밀번호 </label>
            <input
                type="password" id="user_pw" size="10"> <input type="submit"

```

```

        value="로그인">
    </fieldset>
    <fieldset class="register">
        <legend>가입하기</legend>
        <ul>
            <li><label class="reg" for="user_name">아이디 <em>*</em></label>
                <input
                    type="text" id="new_id" size="20" autocomplete="on"
                    required>
            </li>
            <li><label class="reg" for="new_pw1">비밀번호 <em>*</em></label>
                <input
                    type="password" id="new_pw1" size="20" required></li>
            <li><label class="reg" for="new_pw2">비밀번호 확인
                <input type="password" id="new_pw2" size="20"
                    required></li>
            <li><label class="reg" for="user_name">이름 <em>*</em></label>
                <input
                    type="text" id="user_name" size="20" required></li>
            <li><label class="reg" for="user_mail">메일 주소
                <input type="email" id="user_mail" size="20"
                    required></li>
            <li><label class="reg" for="user_tel">전화번호 </label> <input
                type="tel" id="user_tel" size="20"></li>
        </ul>
    </fieldset>
    <fieldset class="easys">
        <legend>이지스퍼블리싱</legend>
        <textarea name="intro" cols="60" rows="5">
열심히 사는 사람들의 손을 잡아주는 곳 - 이지스 퍼블리싱

우리는 책을 내기 전에 다시 한번 물어봅니다
"이 책이 사람들에게 도움이 되는가?"

더 쉽게, 더 빠르게 지식을 전달하고 싶습니다.
이지스퍼블리싱의 책과 앱을 만나보세요.
        </textarea>
    </fieldset>
    <fieldset class="sendform">
        <input type="submit" value="가입하기"> <input type="reset"
            value="다시쓰기">
    </fieldset>
</form>
</body>
</html>

```

## 4.5 기타 다양한 폼 요소들

### (1) <button> 태그 - 버튼 넣기

- 문서에 버튼을 넣고 폼을 전송하거나 리셋할 수 있다
- 기본형: <button [type="submit | reset | button"]> 내용 </button>

[04/button-css.html]

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>웹 폼</title>
<style>
.subm { /* 버튼 스타일 */

```

```

        display: block; /* 블록 레벨 요소 */
        background-color: #fff; /* 배경색 */
        border: 1px solid #dedede; /* 테두리 */
        cursor: pointer; /* 마우스 포인터 */
        padding: 5px 10px 6px 7px; /* 패딩 */
    }

    .subm img { /* 버튼 내 왼쪽 이미지 */
        border: 0; /* 테두리 없음 */
        padding: 0; /* 패딩 없음 */
        width: 16px; /* 가로 크기 */
        height: 16px; /* 세로 크기 */
    }

    .subm:hover { /* 버튼 위로 마우스 포인터 올렸을 때 스타일 */
        background-color: #e6efc2; /* 배경색 */
        border: 1px solid #c6d880; /* 테두리 */
        color: #529215; /* 글자색 */
    }
</style>
</head>

<body>
    <form>
        <button type="submit" class="subm">
             전송하기
        </button>
    </form>
</body>
</html>

```

## (2) <output> 태그 - 결과 출력하기

- 계산 결과를 표시하는 태그
- 기본형: <output [속성="속성값"]> 내용 </output>

```

[04/output.html]

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>웹 폼</title>
</head>

<body>
    <form oninput="result.value=parseInt(num1.value)+parseInt(num2.value)">
        <input type="number" name="num1" value="0"> +<input
            type="number" name="num2" value="0"> =
        <output name="result" for="num"></output>
    </form>
</body>
</html>

```

## (3) <progress> 태그 - 진행 상태 보여주기

- 작업의 진행 상태를 나타내는 태그. 단위가 없다.

```

[04/progress.html]

<!doctype html>
<html lang="ko">
<head>

```

```

<meta charset="utf-8">
<title>웹 품</title>
<style>
body {
    background-color: #fff;
}

ul li {
    list-style: none;
    margin: 15px 0;
    font-size: 14px;
}
</style>
</head>
<body>

    <script>
        function dwn() {
            var bar = document.getElementById("bar"), progress = document
                .getElementById("progress"), loaded = 0;
            var load = function() {
                loaded += 5;
                bar.value = loaded;

                if (loaded == 100) {
                    clearInterval(dummyLoad);
                }
            };

            var dummyLoad = setInterval(function() {
                load()
            }, 500);
        }
    </script>
    <input type="button" value="다운로드 시작" onClick="dwn()">
    <progress id="bar" value="0" max="100">
        <span id="progress"> </span>
    </progress>

</body>
</html>

```

#### (4) <meter> 태그 - 값이 차지하는 크기 표시하기

- 지정된 범위 안에서 해당 값이 어느 정도 차지하는지 표현.

[04/meter.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>웹 품</title>
<style>
body {
    background-color: #fff;
}

ul li {
    list-style: none;
    margin: 15px 0;
    font-size: 14px;
}
</style>
</head>
<body>

```

```

        </ul>
        <li><label>점유율 0.8 </label> <!-- 전체 크기 1을 기준으로 0.8만큼 차지합니다 -->
</li>
</ul>
</li>
<li><label>사용량 64%</label> <!-- 전체 100 중에서 64만큼 차지합니다 --> <meter
value="0.8"></meter></li>
<li><label>사용량 64%</label> <!-- 전체 100 중에서 64만큼 차지합니다 --> <meter
min="0" max="100" value="64"></meter></li>
<li><label>트래픽 초과</label> <!-- 전체 크기는 1024~10240까지인데 높다고 설정한
8192보다 현재 값 9216이 더 큼 -->
<meter min="1024" max="10240" low="2048" high="8192" value="9216"></meter>
</li>
<li><label>적절한 트래픽</label> <!-- 전체 1 중에서 현재 0.5를 차지하고 있으며
적정도를 0.8로 설정했습니다 -->
<meter value="0.5" optimum="0.8"></meter></li>
</ul>
</li>
</ul>
</body>
</html>

```

## [과제] 연습문제



## 5 CSS 기초

### 5.1 스타일과 스타일 시트

#### (1) 왜 스타일을 사용할까?

- 웹 문서의 내용과 상관없이 디자인만 바꿀 수 있다.
  - HTML로는 웹 사이트의 내용을 CSS로는 웹 문서의 디자인을 구성한다는 아이디어가 바로 웹 표준의 시작이다.
  - 유지보수가 쉽다.
- 다양한 기기에 맞게 탄력적으로 바뀌는 문서를 만들 수 있다.

#### (2) 스타일의 형식

- 선택자(selector)는 스타일 규칙을 어디에 적용할 것인지를 나타낸다.
- 기본형: 선택자 { 속성:속성값; }

예) p { text-align: center; } /\* 텍스트 단락을 중앙에 정렬하는 스타일 규칙 \*/  
h2 {  
    font-size: 20px;  
    color: orange;  
} /\* 2단계 제목의 글자 크기 20px, 글자 색상 주황색 \*/

#### (3) 스타일을 표기하는 방법

- 스타일 규칙은 세미콜론(;)으로 구분해 중괄호({}) 안에 나열한다.
- 한 줄로 길게 나열해도 되지만 개발자가 읽기 쉽고 유지보수가 쉽게 여러 줄에 걸쳐 작성하는 것이 좋다.

##### 스타일의 예

```
<style>
h1 {font-size:30px; color:orange;}
h2 {
    font-size:20px;
    color:purple;
}
p {
    color:blue;
}
</style>
```

#### (4) 스타일 주석

- /\*와 \*/ 사이에 내용을 입력하는데 주석을 한 줄만 입력하거나 여러 줄을 입력할 수도 있다.

#### (5) 스타일과 스타일 시트

- 스타일 시트: 스타일 규칙들을 한눈에 확인하고 필요할 때마다 수정하기도 쉽도록 한 군데 묶어 놓은 것이다.
- 내부 스타일 시트
  - 문서 안에서 사용할 스타일 규칙들을 정의한 것이다.
  - <head> </head> 태그 안에서 정의해야 하고 <style> </style> 태그 사이에 작성한다.
- 외부 스타일 시트
  - 별도의 스타일 파일을 만들어 연결해 사용하는 것이다.
  - 기본형: <link href="외부 스타일 파일 경로" rel="stylesheet" type="text/css">
- 인라인 스타일: 간단한 스타일 정보라면 스타일 시트를 사용하지 않고 스타일을 적용할

대상에 직접 표시하는 방법이다. 즉, 해당 태그에 `style="속성:속성값;"` 형태로 스타일을 바꿀 수 있다.

예) `<p style="color:blue;">` 인라인 스타일 `</p>`

[05/internal.html]

```
<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>내부 스타일시트</title>
<style>
ul {
    color: blue; /* 목록의 글자색 */
    list-style-type: square; /* 볼릿 형태 : 사각형 */
}
</style>
</head>
<body>

    
    <h1>세계 3대 미항</h1>
    <ul>

        <li>시드니(Sydney), 호주</li>
        <li>리우데자네이루(Rio de Janeiro), 브라질</li>
        <li>나폴리(Naples), 이탈리아</li>

    </ul>

</body>
</html>
```

[05/external.html]

```
<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>내부 스타일시트</title>
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>

    
    <h1>세계 3대 미항</h1>
    <ul>

        <li>시드니(Sydney), 호주</li>
        <li>리우데자네이루(Rio de Janeiro), 브라질</li>
        <li>나폴리(Naples), 이탈리아</li>

    </ul>

</body>
</html>
```

[05/style.css]

```
ul {
    color: blue;
    list-style-type: square;
}
```

## 5.2 주요 선택자

### (1) 전체 선택자 - 모든 요소에 스타일 적용하기

- 전체 선택자에서는 \*(별표)를 선택자로 사용한다.

```
예) * {  
    margin:0;  
    padding:0;  
}
```

### (2) 태그 선택자 - 특정 태그에 스타일 적용하기

- 기본형: 태그 { 스타일 }

[05/tag-selector.html]

```
<!DOCTYPE html>  
<html lang="ko">  
<head>  
<title>블루베리(Blueberry)</title>  
<style>  
h2 {  
    color: blue;  
}  
  
p {  
    font-size: 12px;  
    margin-left: 20px;  
}  
</style>  
</head>  
<body>  
    <h1>블루베리에 관한 연구</h1>  
    <h2>블루베리와 항산화 효능</h2>  
    <p>블루베리는 항산화제인 안토시아닌과 폴리페놀을 다량 포함하고 있습니다.</p>  
    <p>매사추세츠 보스턴에 있는 USDA 노화에 관한 인류 영양 연구센터 (the USDA Human Nutrition  
        Research Center on Aging) 의 자료에 의하면 블루베리는 과일 중에서 가장 항산화 작용이  
        뛰어난 과일이라고  
        합니다. </p>  
    <h2>블루베리와 노화</h2>  
    <p>USDA 인류 영양 연구센터(the USDA Human Nutrition Research Center) 실험실에서  
        신경과학자들은 쥐들에게 블루베리를 먹임으로써 노화에 의한 인지능력의 손실을 예방해 준다는  
        사실을 발견하였습니다.</p>  
</body>  
</html>
```

### (3) 클래스 선택자 - 특정 부분에 스타일 적용하기

- 특정 부분에만 스타일을 적용할 때 사용한다. 또한 문서 안에서 여러 번 반복해서 적용할 수 있다.
- 기본형: .클래스 이름 { 속성:속성값; 속성:속성값; ... }

[05/class-selector1.html]

```
<!DOCTYPE html>  
<html lang="ko">  
<head>  
<title>블루베리(Blueberry)</title>  
<style>  
.bluetext {  
    color: blue; /* 글자색 */  
}
```

```

p {
    font-size: 12px; /* 글자 크기 */
    margin-left: 20px; /* 왼쪽 마진 여백 */
}
</style>
</head>
<body>
    <h1>블루베리에 관한 연구</h1>
    <h2 class="bluetext">블루베리와 항산화 효능</h2>
    <p class="bluetext">블루베리는 항산화제인 안토시아닌과 폴리페놀을 다량 포함하고 있습니다.</p>
    <p>매사추세츠 보스턴에 있는 USDA 노화에 관한 인류 영양 연구센터 (the USDA Human Nutrition Research Center on Aging) 의 자료에 의하면 블루베리는 과일 중에서 가장 항산화 작용이 뛰어난 과일이라고 합니다. </p>
    <h2>블루베리와 노화</h2>
    <p>USDA 인류 영양 연구센터(the USDA Human Nutrition Research Center) 실험실에서 신경과학자들은 쥐들에게 블루베리를 먹임으로써 노화에 의한 인지능력의 손실을 예방해 준다는 사실을 발견하였습니다.</p>
</body>
</html>

```

#### (4) id 선택자 - 특정 부분에 스타일 적용하기

- 클래스 선택자와 달리 문서 안에서 한 번만 적용한다.
- 기본형: #id 이름 { 속성:속성값; 속성:속성값; ... }

[05/id-selector.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>id 선택자</title>
<style>
#container {
    background: #ff6a00; /* 배경색 */
    width: 400px; /* 너비 */
    height: 200px; /* 높이 */
    margin: 0 auto; /* 가로로 중앙에 배치 */
}
</style>
</head>
<body>
    <div id="container"></div>
</body>
</html>

```

#### (5) 그룹 선택자 - 둘 이상 요소에 같은 스타일 적용하기

- 여러 선택자에 같은 속성을 적용할 수 있다.
- 기본형: 요소1, 요소2, ... {속성;속성값;...}

예) a, p { color: #ffffff; }

### 5.3 캐스캐이딩 스타일 시트(CSS)

#### (1) 캐스캐이딩(Cascading)의 의미

- CSS은 Cascading Style Sheet의 줄임말로 ‘ 위에서 아래로 흐르는 스타일 시트 ’ 즉, 단계적으로 스타일이 적용된다는 뜻을 지고 있다.

## (2) 스타일 우선순위

- 스타일 규칙의 중요도, 적용 범위에 따라 우선순위가 결정되고 그 우선순위에 따라 위에서 아래로 스타일이 적용된다.
- 얼마나 중요한가 - 중요도(Importance)
  - 사용자 스타일 시트가 최우선!
  - 중요 스타일 !important
  - 기본적인 브라우저 스타일 시트
- 얼마나 한정지을 수 있는가 - 명시도(Specificity): 적용범위가 좁을수록 우선순위가 높다.
  - 인라인 스타일: 태그 안에 style 속성을 사용해 해당 태그에만 스타일을 적용
  - id 스타일: 지정한 부분에만 적용되는 스타일이지만 한 문서 안에 한 번만 적용
  - 클래스 스타일: 지정한 부분에만 적용되는 스타일이지만 한 문서 안에 여러 번 적용
  - 태그 스타일: 웹 문서에 사용된 특정 태그에 모두 똑같이 적용
- 소스에서의 순서(Source Order)
  - 같은 요소에 여러가지 스타일이 겹쳐질 때 가장 나중에 정의된 스타일을 적용한다.

[05/cascading1.html]

```
<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>캐스캐이딩 - 중요성</title>
<style>
p {
    font-style: italic; /* 글자 스타일 */
    color: blue; /* 글자색 */
}

p {
    color: red !important; /* 글자색, 우선 적용 */
}
</style>
</head>
<body>

    <h1>세계 3대 미항</h1>
    
    <p>시드니(Sydney), 호주</p>
    <p>리우데자네이루(Rio de Janeiro), 브라질</p>
    <p>나폴리(Naples), 이탈리아</p>

</body>
</html>
```

## (3) 스타일 상속

- 자식 요소에서 별도로 스타일을 지정하지 않으면 부모 요소에 있는 스타일 속성들이 자식 요소로 전달되는 것이다.

[05/inherit.html]

```
<!DOCTYPE html>

<html lang="ko">
<head>
    <meta charset="utf-8">
    <title>뉴욕타임즈 10대 식품</title>
    <style>
        body {
            font-family:"돋움"; /* 글꼴 */
        }
    </style>
</head>
<body>
```

```

        color:darkgreen; /* 글자색 */
    }
</style>
</head>
<body>
<h1>뉴욕 타임즈에서 발표한 세계 10대 슈퍼푸드</h1>
<ul>
<li>블루베리</li>
<li>귀리</li>
<li>토마토</li>
<li>시금치</li>
<li>적포도주</li>
<li>견과류</li>
<li>브로콜리</li>
<li>연어</li>
<li>마늘</li>
<li>녹차</li>
</ul>
</body>
</html>

```

## 5.4 CSS3와 CSS 모듈

애니메이션이나 이미지 편집처럼 다른 프로그램의 도움을 받아야 했던 효과들을 CSS3 소스 코드로만으로 해결할 수 있기 때문에 웹 표준 기술에서 중요한 위치를 차지하고 있다.

### (1) CSS3란?

- CSS2나 CSS1에 비해 좀 더 정교하고 화려한 화면을 구성할 수 있고 애니메이션도 지원한다.
- 여러 모듈로 쪼개어 개발 중 → 일부는 완성되었지만 대부분 아직도 개발 중

### (2) CSS3와 브라우저 접두사(prefix)

- CSS3 표준 규약이 완성되지 않아 대부분의 CSS3 속성을 사용하려면 속성 이름 앞에 브라우저를 식별할 수 있는 접두사(prefix)를 붙여서 사용해야 한다.

-webkit-	웹킷(Webkit) 방식의 브라우저용(사파리, 크롬 등)
-moz-	게코(Gecko) 방식의 브라우저용(모질라, 파이어폭스 등)
-o-	오페라 브라우저
-ms	마이크로소프트 인터넷 익스플로러

[05/transform.html]

```

<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Transform</title>
<style>
.box {
    position: absolute;
    left: 50px;
    top: 70px;
    width: 100px;
    height: 60px;
    background: #fff;
    border: 2px solid green;
    text-align: center;
    line-height: 60px;

```

```
}  
  
.box:hover {  
    -webkit-transform: rotate(15deg);  
    -moz-transform: rotate(15deg);  
    -o-transform: rotate(15deg);  
    -ms-transform: rotate(15deg);  
    transform: rotate(15deg);  
}  
</style>  
</head>  
  
<body>  
    <div class="box">Mouse Over</div>  
</body>  
</html>
```

## 6 텍스트 관련 스타일

### 6.1 글꼴 관련 스타일

#### (1) font-family 속성 - 글꼴 지정하기

- 웹 문서에서 사용할 글꼴 지정
- body 태그를 비롯해 p 태그나 h1 태그처럼 텍스트를 사용하는 요소들에서 사용
- 기본형: font-family:"글꼴 이름","글꼴 이름", "글꼴 이름"

예) p {font-family:"굴림";} /\* 텍스트 단락의 글꼴을 ‘굴림’ \*/  
body {font-family:"맑은 고딕", "돋움", "굴림";}

#### (2) @font-face 속성 - 웹 폰트 사용하기

- 웹 폰트(web font): 웹 문서를 작성할 때 웹 문서 안에 글꼴 정보도 함께 저장했다가 사용자가 웹 문서에 접속하면 글꼴을 사용자 시스템으로 다운로드하게 하는 방식이다.
- 트루타입 글꼴(\*.ttf)을 웹 폰트로 사용하려면 eot 파일과 woff 파일로 변환해야 한다.
- 기본형: @font-face {  
font-family:글꼴 이름;  
src:url(글꼴 파일 경로) format(파일 유형);  
}

[06/webfont1.html] 웹 폰트 사용하기

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>요안도라</title>
<style>
body {
    background-color: #09280B;
}
@font-face{
    font-family:'myBand';
    src:url('Band.eot');
    src:url('Band.ttf') format('truetype');
    src:url('Band.woff') format('woff');
}
h1{
    font-family:myBand;
    font-size:3em;
    color:yellow;
}

</style>
</head>

<body>
<h1>Joandora</h1>
</body>
</html>

<!DOCTYPE html>

<html lang="ko">
<head>
    <meta charset="utf-8">
    <title>웹 폰트 사용하기</title>
    <style>
        @font-face {
            font-family: 'trana'; /* 글꼴 */
```



```

        src: local('trana'),
            url('trana.eot'),
            url('trana.woff') format('woff'),
            url('trana.ttf') format('truetype');
    }
    .w-font {
        font-family:'trana', sans-serif; /* 웹 폰트 지정 */
    }
    p {
        font-size:30px; /* 글자 크기 */
    }
</style>
</head>
<body>
    <p>Using Default Fonts</p>
    <p class="w-font">Using Trana Fonts</p>
</body>
</html>

```

[06/webfont2.html] 구글 웹 폰트 사용하기

```

<!DOCTYPE html>

<html lang="ko">
<head>
    <meta charset="utf-8">
    <title>웹 폰트 사용하기</title>
    <style>
        @import url(http://fonts.googleapis.com/earlyaccess/nanumgothic.css); /* 구글 웹 폰트 */
        .ng-font {
            font-family:'Nanum Gothic', 돋움; /* 웹 폰트 지정 */
        }
        p {
            font-size:30px; /* 글자 크기 */
        }
    </style>
</head>
<body>
    <p>브라우저 기본 글꼴 사용</p>
    <p class="ng-font">나눔고딕 웹 폰트 사용</p>
</body>
</html>

```

### (3) font-size 속성 - 글자 크기 조절하기

- 기본형: font-size: 속성 값
- 속성 값: 절대 크기, 상대 크기, 숫자, 백분율
- 기본 값은 상대 크기인 medium
- font-size 속성은 상속된다.
- px 단위를 사용하면 폰트 크기가 고정되기 때문에 창 크기가 작은 모바일 기기로 볼때 작은 화면 안에 작은 글씨로 표시된다.
- em 단위 사용하면 부모 요소의 폰트에 비례하여 상대적 값을 계산해 글자 크기를 조절한다.

[06/font-size.html]

```

<!DOCTYPE html>

<html lang="ko">
<head>
    <meta charset="utf-8">

```

```

<title>폰트 크기</title>
<style>
h1 {
    font-size: 3em; /* 글자 크기 */
}

p {
    font-size: 1em; /* 글자 크기 */
}
</style>
</head>
<body>
    <h1>3em의 크기를 가진 제목</h1>
    <p>1em의 크기를 가진 단락</p>
</body>
</html>

```

#### (4) font-weight 속성 - 글자 굵기 지정하기

- 글자의 굵기 지정. 사용할 수 있는 값은 키워드와 숫자
- 기본형: font-weight: normal | bold | lighter | 100 | 200 | ...

#### (5) font-variant 속성 - 작은 대문자로 표시하기

- 영문 소문자를 작은 대문자로 표시
- 작은 대문자는 원래 대문자와 비슷하게 보이지만, 크기가 작고 글자의 가로,세로 비율이 약간 다르다.
- 기본형: font-variant: normal | small-caps

[06/font-vary.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>Font Variant & Weight</title>
<style>
.accent {
    font-variant: small-caps; /* 작은 대문자 */
    font-weight: bold; /* 굵게 */
}
</style>
</head>
<body>
    <h1>세계 3대 미항</h1>
    <p>
        <span class="accent">시드니(Sydney)</span>, 호주
    </p>
    <p>
        <span class="accent">리우데자네이루(Rio de Janeiro)</span>, 브라질
    </p>
    <p>
        <span class="accent">나폴리(Naples)</span>, 이탈리아
    </p>
</body>
</html>

```

#### (6) font-style 속성 - 글자 스타일 지정하기

- 글자를 이탤릭체로 표현할 것인지의 여부를 결정
- italic이나 oblique를 선택하면 글자가 이탤릭체로 표시됨
- 기본형: font-style: normal | italic | oblique

[06/font-style.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>폰트 스타일</title>
<style>
p {
    font-style: italic;
} /* 이탤릭체로 */
p#txt {
    font-style: normal;
} /* p 요소 중 id=txt인 부분은 보통체로 */
</style>
</head>
<body>
    <h1>세계 3대 미항</h1>
    <p>시드니(Sydney), 호주</p>
    <p>리우데자네이루(Rio de Janeiro), 브라질</p>
    <p id="txt">나폴리(Naples), 이탈리아</p>
</body>
</html>
```

## (7) font 속성 - 글꼴 속성을 한꺼번에 묶어서 표현하기

- font-size와 line-height는 대시(/)로 연결해 함께 표현한다.
- font 속성에 표시하지 않는 값들은 기본 값들로 사용한다.

예) p { font:12pt/14pt 굴림 } /\* <p> 태그 텍스트 크기 12pt, 줄간격 14pt, 글꼴 굴림 \*/  
p { font:medium/110% "굴림 돋움" bold } /\* <p> 태그 텍스트 크기 medium, 줄간격 110%, 글꼴 굴림 돋움 진하게 표시 \*/

[06/font.html]

```
<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>글꼴 스타일</title>
<style>
p.txt {
    font: italic 12px/24px 돋움; /* 글꼴 스타일, 크기, 줄간격, 글꼴 */
}
</style>
</head>
<body>
    <p class="txt">여러 요소가 함께 사용된 웹 문서 안에서 텍스트가 눈에 띄게 하려면 내용에 맞는 글꼴과
    글자
        크기, 그리고 글자색을 선택하는 것이 중요합니다.</p>
    <p>이럴 때 사용할 수 있는 것이 글꼴 속성입니다.</p>
    <p style="font: caption">[font:caption] 캡션에 어울리는 글꼴 스타일</p>
    <p style="font: icon">[font:icon] 아이콘에 어울리는 글꼴 스타일</p>
    <p style="font: menu">[font:menu] 드롭다운 메뉴에 어울리는 글꼴 스타일</p>
    <p style="font: message-box">[font:message-box] 대화 상자에 어울리는 글꼴 스타일</p>
    <p style="font: small-caption">[font:small-caption] 작은 캡션에 어울리는 글꼴
        스타일</p>
    <p style="font: status-bar">[font:status-bar] 상태표시줄에 어울리는 글꼴 스타일</p>
</body>
</html>
```

## 6.2 텍스트 스타일

### (1) color 속성 - 글자 색 지정하기

- 기본형: color : <색상>

[06/color.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>뉴욕타임즈 10대 식품</title>
<style>
h1 {
    color: rgb(0, 200, 0);
} /* rgb 값 사용 - 녹색 계열 */
h2 {
    color: blue;
} /* 색상 이름 사용 - 파랑 */
.accent {
    color: #f00;
} /* 16진수 사용 - 빨강, #ff0000으로도 사용 */
img {
    float: right; /* 오른쪽 정렬 */
    margin-right: 10px; /* 오른쪽 마진 여백 */
}
</style>
</head>
<body>
    <h1>세계 10대 슈퍼푸드</h1>
    
    <h2>마늘(Garlic)</h2>
    <p>일해백리(日海百利)는 마늘의 별명이다. 한가지 해가 있고 백가지 이로움이 있다는 뜻이다. 그 한 가지
해란, 아린
        맛으로 인해 위에 부담을 준다는 것이다.</p>
    <p>
        마늘 특유의 아린 맛은 <span class="accent">알리신</span>이라는 성분 때문으로, 살균 및 항균
        작용을
        하며 마늘에 들어 있는 아연이 피로 회복에 도움을 준다.
    </p>
</body>
</html>
```

### (2) text-decoration 속성 - 텍스트에 줄 표시하기/없애기

- 텍스트에 밑줄을 긋거나 가로지르는 줄 같은 것을 표시
- 기본형: text-decoration : none | underline | overline | line-through

[06/text-decoration.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>뉴욕타임즈 10대 식품</title>
<style>
a {
    text-decoration: none;
} /* 밑줄 없앰 */
.edited {
    text-decoration: line-through;
}
```

```

} /* 취소선 */
</style>
</head>
<body>
    <h2>토마토(Tomato)</h2>
    <p>
        [<a href="https://www.fitbug.com/g/Superfoods-tomatoes"
            target="_blank">외부 링크</a>]
    </p>
    <p>
        토마토는 비타민A, C가 풍부한 <span class="edited">과일이다.</span> 채소다. 칼륨과 같은
        각종 미네랄은 혈액의 산성도를 낮추는 역할을 해주며 혈압을 내리고 혈관을 튼튼하게 해준다.
        토마토에 들어있는 피코펜이라는 성분은
        뛰어난 항암 작용을 보이며 잘 알려져 있듯이 블루베리와 함께 대표적인 항산화 음식이기도
        하다.
    </p>
</body>
</html>

```

### (3) text-transform 속성 - 텍스트 대·소문자 변환하기

- 텍스트들을 대문자 혹은 소문자로 변환.
- 한글에는 영향을 주지 않고 영문자에만 적용
  - capitalize : 각 단어의 첫 번째 글자를 대문자로 변환하고
  - uppercase : 단어의 모든 글자를 대문자로 변환
  - lowercase : 단어의 모든 글자를 소문자로 변환
  - none: 변환하지 않음
- 기본형: text-transform : none | capitalize | uppercase | lowercase | full-width

### (4) text-shadow 속성 - 텍스트에 그림자 효과 추가하기

- 텍스트에 그림자 효과를 추가해서 텍스트를 좀더 입체적으로 보이게 속성 값을 조절하면  
서 글자가 더욱 선명하게 보이도록 할 수도 있다.
- 마치 그래픽으로 처리한 듯한 텍스트를 만들 수도 있다.
- 본문 내용에서 텍스트 그림자 효과를 자주 사용하면 너무 지저분해 보이지만 사이트 제  
목이나 강조해야 할 글자에는 그림자를 추가함으로써 눈에 띄게 만들 수 있다.

예) .shadow3 {

```

    color:#fff; /* 글자색 */
    text-shadow:7px 7px 5px #000; /* 텍스트 그림자 */
}

```

[06/text-shadow1.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>text-transform</title>
<style>
h1 {
    font-size: 100px; /* 글자 크기 */
    font-family: "Arial Rounded MT"; /* 글꼴 */
}

.shadow1 {
    color: orange; /* 글자색 */
    text-shadow: 1px 1px; /* 텍스트 그림자 */
}

```

```

.shadow2 {
    text-shadow: 5px 5px 3px #f00; /* 텍스트 그림자 */
}

.shadow3 {
    color: #fff; /* 글자색 */
    text-shadow: 7px -7px 5px #000; /* 텍스트 그림자 */
}
</style>
</head>
<body>
    <h1 class="shadow1">HTML5</h1>
    <h1 class="shadow2">HTML5</h1>
    <h1 class="shadow3">HTML5</h1>
</body>
</html>

```

## (5) white-space 속성

- 텍스트에 함께 입력된 공백을 어떻게 처리할 것인지 지정
- 사용할 수 있는 값
  - normal: 연속하는 공백을 하나로 처리. 기본값.
  - nowrap: 연속하는 공백을 하나로 처리하고 줄바꿈을 하지 않는다.
  - pre: 연속하는 공백을 원본 그대로 표시하고 줄바꿈하지 않는다.
  - pre-line: 연속하는 공백을 하나로 취급하고 줄바꿈을 한다.
  - pre-wrap: 연속하는 공백을 원본 그대로 표시하고 줄바꿈을 한다.

예) body { white-space: nowrap; } /\* 공백이 길더라도 줄바꿈 없이 한 칸만 인식\*/

## (6) letter-spacing과 word-spacing 속성 - 텍스트 간격 조절하기

- 글자와 글자 사이의 간격 조절
- 강조하고 싶은 글자나 <h> 태그를 이용해 크게 표시한 글자들
  - 글자 사이의 간격을 조절해서 좀더 여유 있게 표시하면 읽기 편하다.

예) .accent { letter-spacing: 3px; } /\* 글자 사이의 간격을 3px로 지정 \*/

```

[06/letter-spacing.html]

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>letter-spacing와 word-spacing</title>
<style>
h1 {
    font-size: 40px;
} /* 글자 크기 */
.letter1 {
    letter-spacing: 0.2em; /* 자간 */
}

.letter2 {
    letter-spacing: 0.5em; /* 자간 */
}
</style>
</head>
<body>
    <h1>HTML5</h1>
    <h1 class="letter1">HTML5</h1>

```

```

<h1 class="letter2">HTML5</h1>
</body>
</html>

```

## 6.3 문단 스타일

### (1) direction 속성

- 텍스트의 쓰기 방향을 지정한다.

예) h3 { direction:rtl; } /\* 제목을 오른쪽에서 왼쪽으로 표시 \*/

### (2) text-align 속성 - 텍스트 정렬하기

- 문단의 텍스트 정렬 방법 지정
- 사용할 수 있는 속성값은 left 와 right, center, justify
- 기본값은 브라우저에 따라 달라지며, 부모 요소의 값을 상속한다.

[06/text-align.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>뉴욕타임즈 10대 식품</title>
<style>
p {
    border: 1px solid #ccc; /* 테두리 - 1픽셀짜리 회색 실선 */
    padding: 10px; /* 테두리와 내용 사이의 패딩 여백 */
    margin: 10px; /* 단락 주변의 마진 여백 */
}

.align-left {
    text-align: left;
} /* 왼쪽 정렬 */
.align-right {
    text-align: right;
} /* 오른쪽 정렬 */
.align-center {
    text-align: center;
} /* 가운데 정렬 */
.align-justify {
    text-align: justify;
} /* 양쪽 정렬 */
</style>
</head>
<body>

    <h1>텍스트 정렬</h1>

    <!-- 왼쪽 정렬 -->
    <p class="align-left">Integer elementum massa at nulla placerat
        varius. Suspendisse in libero risus, in interdum massa. Vestibulum ac
        leo vitae metus faucibus gravida ac in neque. Nullam est eros,
        suscipit sed dictum quis, accumsan a ligula.</p>

    <!-- 오른쪽 정렬 -->
    <p class="align-right">Integer elementum massa at nulla placerat
        varius. Suspendisse in libero risus, in interdum massa. Vestibulum ac
        leo vitae metus faucibus gravida ac in neque. Nullam est eros,
        suscipit sed dictum quis, accumsan a ligula.</p>

    <!-- 가운데 정렬 -->
    <p class="align-center">Integer elementum massa at nulla placerat

```

```
varius. Suspendisse in libero risus, in interdum massa. Vestibulum ac
leo vitae metus faucibus gravida ac in neque. Nullam est eros,
suscipit sed dictum quis, accumsan a ligula.</p>

<!-- 양쪽 정렬 -->
<p class="align-justify">Integer elementum massa at nulla placerat
varius. Suspendisse in libero risus, in interdum massa. Vestibulum ac
leo vitae metus faucibus gravida ac in neque. Nullam est eros,
suscipit sed dictum quis, accumsan a ligula.</p>

</body>
</html>
```

(3) text-justify 속성 - 정렬 시 공백 조절하기

(4) text-indent 속성 - 텍스트 들여쓰기

- 텍스트가 많은 문서일 경우 여러 문단으로 이루어지는데, 문단의 첫 글자를 얼마나 들여 쓸지 지정

(5) line-height 속성 - 줄간격 조절하기

- 여러 줄로 구성된 텍스트 단락에서 줄간격 조절
- 숫자나 크기 값, 백분율 사용 가능.
- 숫자나 크기 값을 지정하면 글자 크기를 기준으로 계산함.

```
예) p {font-size:10pt; line-height:1.2;} /* 숫자를 사용 */
    p {font-size:10pt; line-height:120%;} /* 백분율 사용 */
    p {font-size:10pt; line-height:12pt;} /* 크기 값 사용 */
```

```
[06/line-height.html]

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>줄 간격</title>
<style>
p {
    font-size: 15px;
    border: 1px solid gray;
    padding: 10px;
}

.big-line {
    line-height: 2; /* 글자 크기 2배만큼의 줄간격 */
}

.small-line {
    line-height: 0.7; /* 글자 크기 0.7배만큼의 줄간격 */
}
</style>
</head>
<body>

    <h2>블루베리(Blueberry)</h2>
    <p>블루베리의 대표적인 기능은 항산화로 비타민A, C, E가 풍부하고 안토니시아민, 페놀 등이 활성 산소를
없애 노화를
        억제한다. 이밖에 블루베리의 안토니시아민은 눈의 피로와 시력 저하를 회복시키는 효능을
가지고 있다.</p>
    <p class="small-line">블루베리의 대표적인 기능은 항산화로 비타민A, C, E가 풍부하고 안토니시아민,
        페놀 등이 활성 산소를 없애 노화를 억제한다. 이밖에 블루베리의 안토니시아민은 눈의 피로와
시력 저하를 회복시키는 효능을
```



```

        가지고 있다.</p>
        <p class="big-line">블루베리의 대표적인 기능은 항산화로 비타민A, C, E가 풍부하고 안토니시아민, 페놀
        등이 활성 산소를 없애 노화를 억제한다. 이밖에 블루베리의 안토니시아민은 눈의 피로와 시력
        저하를 회복시키는 효능을 가지고
        있다.</p>
    </body>
</html>

```

## (6) text-overflow 속성 - 넘치는 텍스트 표기하기

- 기본형: text-overflow: clip | ellipsis

[06/text-overflow.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>슈퍼푸드</title>
<style>
.content {
    border: 1px solid #ccc; /* 테두리 */
    width: 300px; /* 단락의 너비 */
    white-space: nowrap; /* 줄바꿈 없음 */
    overflow: hidden; /* 넘치는 부분 감춤 */
    text-overflow: ellipsis; /* 말줄임표 */
}

.content:hover {
    overflow: visible; /* 넘치는 부분 보여줌 */
}
</style>
</head>
<body>
    <h2>귀리(Oat) </h2>
    <p class="content">귀리는 베타글루칸(항암 및 면역증강작용을 가지고 있는 불소화성 다당류) 성분을
    포함하고 있다.</p>
</body>
</html>

```

## [실습] 상품 소개 페이지 - 텍스트 스타일 활용하기

p254 ~ p258

## 6.4 목록과 링크 스타일

용도: 텍스트 나열, 사이트 메뉴, 문서 레이아웃(사진 갤러리)

### (1) list-style-type 속성 - 목록의 불릿과 번호 스타일 지정하기

- 기본형: list-style-type : none | <순서 없는 목록의 불릿> | <순서 목록의 번호>

예) ul { list-style-type: circle; } /\* 순서 없는 목록의 불릿을 circle로 표시 \*/

[06/ordered.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>불릿 바꾸기</title>
<style>

```

```

.book1 {
    list-style-type: lower-alpha; /* 소문자 알파벳 */
}

.book2 {
    list-style-type: upper-roman; /* 대문자 로마 숫자 */
}
</style>
</head>
<body>
    <h1>도서 시리즈</h1>
    <ol class="book1">
        <li>Do it! 시리즈</li>
        <li>된다 시리즈</li>
        <li>DCM 프로 사진가</li>
        <li>데이터과학 시리즈</li>
    </ol>

    <ol class="book2">
        <li>Do it! 시리즈</li>
        <li>된다 시리즈</li>
        <li>DCM 프로 사진가</li>
        <li>데이터과학 시리즈</li>
    </ol>
</body>
</html>

```

## (2) list-style-image 속성 - 불릿 대신 이미지 넣기

- 불릿 기호 대신 원하는 이미지로 삽입하는 속성
- 사용할 수 있는 값은 불릿 이미지 파일 경로
- 기본형: list-style-image : none | url(이미지 파일 경로)

```

[06/image-bullet.html]

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>불릿 바꾸기</title>
<style>
ul {
    list-style-image: url('images/dot.png'); /* 불릿으로 사용할 이미지 */
}
</style>
</head>
<body>
    <h1>이지스퍼블리싱</h1>
    <ul>
        <li>회사소개</li>
        <li>도서</li>
        <li>자료실</li>
        <li>질문답변</li>
        <li>동영상강의</li>
    </ul>
</body>
</html>

```

## (3) list-style-position 속성 - 목록에 들여쓰기 효과 내기

- 목록의 불릿이나 번호는 실제 내용 바깥쪽에 표시됨.
- list-style-position을 이용하면 내용이 있는 부분에 표시(들여쓰기 효과)

- 기본형: list-style-position : inside | outside

[06/list-position.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>볼릿 바꾸기</title>
<style>
.inside {
    list-style-position: inside;
}
</style>
</head>
<body>
    <h3>list-style-position을 지정하지 않음</h3>
    <ul>
        <li>회사소개</li>
        <li>도서</li>
        <li>자료실</li>
        <li>질문답변</li>
        <li>동영상강의</li>
    </ul>
    <h3>list-style-position : inside</h3>
    <ul class=inside>
        <li>회사소개</li>
        <li>도서</li>
        <li>자료실</li>
        <li>질문답변</li>
        <li>동영상강의</li>
    </ul>
</body>
</html>
```

#### (4) list-style 속성 - 리스트 속성 한꺼번에 표시하기

- list-style-type과 list-style-position, list-style-image 속성을 한꺼번에 표시하는 속성
- 각 속성의 순서는 중요하지 않다.

예) article ol li {  
list-style:upper-alpha inside; /\* 알파벳 대문자 사용, 안으로 들여쓰기 적용 \*/

## [과제] 연습문제

p266 ~ p267

## 7 색상과 배경을 위한 스타일

### 7.1 웹에서 색상 표현하기

#### (1) 16진수 표기법

- #ffffff 처럼 #과 함께 6자리의 16진수로 표시.
- 앞에서부터 두자리씩 묶어 빨강, 초록, 파랑의 양.
- 하나도 섞이지 않았을 때는 00, 가득 섞였을 때는 ff로 표시.
- 사용할 수 있는 값은 000000(검은색)에서부터 fffffff(흰색)까지.  
예) p { color:#000000; } /\* 글자색 검정 \*/  
.accent { color:#ff0000; } /\* accent 클래스 스타일의 색 빨강 \*/

#### (2) rgb와 rgba 표기법

- color:rgb(255,0,0)처럼 세 자리의 숫자로 표시.
- 앞의 숫자부터 빨강, 초록, 파랑의 양.
- 하나도 섞이지 않았을 때는 0, 가득 섞였을 때는 255로 표시
- 투명도를 조절할 때는 color:rgb(255,0,0,.3) 처럼 마지막에 알파값 추가
- 알파값은 불투명도를 나타내는 값으로 0~1 값 중에서 사용(1은 불투명, 0은 완전 투명)  
예) #box1 { background-color:rgb(255,0,0); } /\* box1이라는 스타일의 배경색은 빨강 개열 \*/
- 기본형: rgba(red 값, green 값, blue 값, alpha 값);  
hsla(hue 값, saturation 값, lightness 값, alpha 값);
- 투명도를 조절할 때는 마지막에 알파값 추가
- 알파값은 불투명도를 나타내는 값으로 0~1 값 중에서 사용(1은 불투명, 0은 완전 투명)

예) 반투명한 빨간색 : rgba(255,0,0,0.5) 또는 rgba(255,0,0,.5),  
hsla(360,100%,50%,0.5)

#### (3) hsl과 hsla 표기법

- color:hsl(240, 100%, 50%) 처럼 세 자리의 숫자로 표시
- 앞의 숫자부터 색상(hue), 채도(saturation), 밝기(light)의 양.
- 예) #area2 { background:hsl(360,100%,75%); } /\* 색상은 빨강, 채도 100%, 밝기 75%인 색 \*/

#### (4) 색상 이름 표기법

- red나 yellow, black 처럼 잘 알려진 색상 이름으로 표시한다.

[07/color.html]

```
<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>Color in Web</title>
<style>
body {
    background: #222;
}

.text1 {
    color: rgb(255, 255, 255);
}
```

```

.text2 {
    color: rgba(255, 255, 255, 0.8);
}

.text3 {
    color: rgba(255, 255, 255, 0.5);
}

.text4 {
    color: rgba(255, 255, 255, 0.2);
}
</style>
</head>
<body>
    <h1 class="text1">HTML5+CSS3</h1>
    <h1 class="text2">HTML5+CSS3</h1>
    <h1 class="text3">HTML5+CSS3</h1>
    <h1 class="text4">HTML5+CSS3</h1>
</body>
</html>

```

## (5) 색상 추출 사이트 이용하기

- 색상 추출 사이트인 Color Picker([www.colorpicker.com](http://www.colorpicker.com))로 색상을 추출한다.

## 7.2 배경색과 배경 이미지

### (1) background-color 속성 - 배경색 지정하기

- 16진수나 rgb 값, rgba 값 또는 색상 이름을 사용한다.
- background-color 값은 상속되지 않는다.
- 기본형: background-color : <색상>

[07/bg-color.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>배경색</title>
<style>
body {
    background-color: #0094ff; /* 문서 전체 배경색 */
}

div {
    background-color: white; /* div 영역 배경색 */
    width: 90%; /* 너비 */
    padding: 15px; /* 패딩 */
    border: 1px solid black; /* 테두리 */
}

p {
    line-height: 30px;
}
</style>
</head>
<body>
    <div>
        <h1>초콜릿(Chocolate)</h1>
        <p>초콜릿은 카카오 콩을 재료로 가공한 식품이다. 숙성한 카카오 콩을 볶은 뒤 이를
        갈아서 만든 카카오 매스와 지방
        성분만으로 만들어진 코코아 버터를 혼합하여 만드는데, 설탕 등의 다른 재료를
        더 넣어 만들기도 한다. 카카오 매스의 함량에

```

```

    </div>
    </body>
</html>

```

따라 다크 초콜릿, 밀크 초콜릿, 화이트 초콜릿으로 구분한다.</p>

## (2) background-clip 속성 - 배경 적용 범위 조절하기

- 기본형: background-clip : border-box | padding-box | content-box
- 배경 이미지나 배경색을 어디까지 적용할지 범위 지정
  - border-box : 테두리 부분까지 적용
  - padding-box : 패딩 부분까지 적용
  - content-box : 마진 부분까지 적용

[07/bg-clip.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>배경 이미지</title>
<style>
div {
    width: 300px;
    padding: 35px;
    border: 5px dotted #222;
    background: skyblue;
    margin-bottom: 20px;
}

.bg1 {
    background-clip: border-box;
}

.bg2 {
    background-clip: padding-box;
}

.bg3 {
    background-clip: content-box;
}
</style>
</head>
<body>
    <div class="bg1">
        <p>
            <strong>토마토 : </strong> 토마토는 비타민A, C가 풍부할 뿐 아니라 희귀한
            성분인 비타민 H와 P까지
            포함된 과일이다. 칼륨과 같은 각종 미네랄은 혈액의 산성도를 낮추는 역할을
            해주며 토마토에 들어있는 피코펜이라는 성분은
            뛰어난 항암 작용을 보이며 잘 알려져 있듯이 블루베리와 함께 대표적인 항산화
            음식이기도 하다.
        </p>
    </div>
    <div class="bg2">
        <p>
            <strong>토마토 : </strong> 토마토는 비타민A, C가 풍부할 뿐 아니라 희귀한
            성분인 비타민 H와 P까지
            포함된 과일이다. 칼륨과 같은 각종 미네랄은 혈액의 산성도를 낮추는 역할을
            해주며 토마토에 들어있는 피코펜이라는 성분은
            뛰어난 항암 작용을 보이며 잘 알려져 있듯이 블루베리와 함께 대표적인 항산화
            음식이기도 하다.
        </p>
    </div>

```

```

<div class="bg3">
  <p>
    <strong>토마토 : </strong> 토마토는 비타민A, C가 풍부할 뿐 아니라 희귀한
    성분인 비타민 H와 P까지
    포함된 과일이다. 칼륨과 같은 각종 미네랄은 혈액의 산성도를 낮추는 역할을
    해주며 토마토에 들어있는 피코펜이라는 성분은
    뛰어난 항암 작용을 보이며 잘 알려져 있듯이 블루베리와 함께 대표적인 항산화
    음식이기도 하다.
  </p>
</div>
</body>
</html>

```

### (3) background-image 속성 - 웹 요소에 이미지 넣기

- 기본형: background-image: url(파일경로)
- 배경 이미지 지정하기
- 배경 이미지 파일 경로 지정
- 상대 경로 또는 절대 경로 사용

예) body { background-image: url("bg1.png"); }  
 #area { background-image: url("bg2.png"); }

### (4) background-repeat 속성 - 배경 이미지 반복하기

- 배경 이미지 반복 여부와 반복 방향 지정

### (5) background-size 속성 - 배경 이미지 크기 조절하기

- 요소 크기에 따라 배경 이미지를 여러 가지 크기로 사용
- 기본형: background-size: auto | contain | cover | <크기 값> | <백분율>

[07/bg-size.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>배경 이미지</title>
<style>
div {
  width: 500px;
  height: 500px;
  border: 2px double black;
  margin-bottom: 20px;
}

.bg1 {
  background: url('images/bg4.jpg') no-repeat left top;
  background-size: auto;
}

.bg2 {
  background: url('images/bg4.jpg') no-repeat left top;
  background-size: 200px 150px;
}

.bg3 {
  background: url('images/bg4.jpg') no-repeat left top;
  background-size: 60% 40%;
}

```

```

.bg4 {
    background: url('images/bg4.jpg') no-repeat left top;
    background-size: contain;
}

.bg5 {
    background: url('images/bg4.jpg') no-repeat left top;
    background-size: cover;
}

.bg6 {
    background: url('images/bg4.jpg') no-repeat left top;
    background-size: 100% 100%;
}
</style>
</head>
<body>
    <div class="bg1"></div>
    <div class="bg2"></div>
    <div class="bg3"></div>
    <div class="bg4"></div>
    <div class="bg5"></div>
    <div class="bg6"></div>
</body>
</html>

```

## (6) background-position 속성 - 배경 이미지 위치 조절하기

- 배경 이미지를 반복하지 않고 한번만 표시할 경우에 특정한 위치에 배경 이미지를 배치할 수 있음
- 백분율(%) 표기법: 이미지의 가로 위치와 세로 위치를 %로 나타냄.  
예) background-position: 0% 0% , background-position : 20% 60%
- 길이(px) 표기법 : 배경 이미지의 위치를 직접 길이로 지정  
예) background-position: 30px 20px;
- 키워드 표기법: 가로 배치는 left와 center, right 중에서 선택, 세로 배치는 top과 center, bottom 중에서 선택한다.  
예) background-position: center bottom

[07/bg-position.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>배경 이미지</title>
<style>
div {
    position: relative;
    left: 20%;
    width: 60%;
    height: auto;
    border: 1px gray solid;
    padding: 10px;
    margin-bottom: 20px;
}

#bg1 {
    background-image: url('images/bg2.jpg');
    background-repeat: no-repeat;
    background-position: left top;
    padding-top: 150px;
}

```



```
#bg2 {
    background-image: url('images/bg2.jpg');
    background-repeat: no-repeat;
    background-position: center;
}
</style>
</head>
<body>
    <div id="bg1">
        <h1>초콜릿(Chocolate)</h1>
        <p>초콜릿은 카카오 콩을 재료로 가공한 식품이다. 숙성한 카카오 콩을 볶은 뒤 이를
갈아서 만든 카카오 매스와 지방 성분만으로 만들어진 코코아 버터를 혼합하여 만드는데, 설탕 등의 다른 재료를
더 넣어 만들기도 한다. 카카오 매스의 함량에 따라 다크 초콜릿, 밀크 초콜릿, 화이트 초콜릿으로 구분한다.</p>
    </div>
    <div id="bg2">
        <h1>초콜릿(Chocolate)</h1>
        <p>초콜릿은 카카오 콩을 재료로 가공한 식품이다. 숙성한 카카오 콩을 볶은 뒤 이를
갈아서 만든 카카오 매스와 지방 성분만으로 만들어진 코코아 버터를 혼합하여 만드는데, 설탕 등의 다른 재료를
더 넣어 만들기도 한다. 카카오 매스의 함량에 따라 다크 초콜릿, 밀크 초콜릿, 화이트 초콜릿으로 구분한다.</p>
    </div>
</body>
</html>
```

## (7) background-origin 속성 - 배경 이미지 위치 기준 조절하기

- 기본형: background-origin: border-box | padding-box | content-box
- 배경 이미지 위치의 기준을 어디로 할지 지정할 수 있다.
  - border-box : 박스 모델의 가장 외곽인 테두리(border)까지 포함해서 기준
  - padding-box : 테두리를 빼고 패딩(padding) 범위 안에서 기준
  - content-box : 내용 부분만 기준

[07/bg-origin.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>배경 이미지</title>
<style>
div {
    width: 300px;
    padding: 35px;
    border: 15px solid #eee;
    background: url('images/bg5.png') no-repeat;
    margin-bottom: 20px;
}

.bg1 {
    background-origin: padding-box;
}

.bg2 {
    background-origin: border-box;
}

.bg3 {
    background-origin: content-box;
}
}
```

```

</style>
</head>
<body>
    <div class="bg1">
        <p>
            <strong>토마토 : </strong> 토마토는 비타민A, C가 풍부할 뿐 아니라 희귀한
            성분인 비타민 H와 P까지
            포함된 과일이다. 칼륨과 같은 각종 미네랄은 혈액의 산성도를 낮추는 역할을
            해주며 토마토에 들어있는 피코펜이라는 성분은
            뛰어난 항암 작용을 보이며 잘 알려져 있듯이 블루베리와 함께 대표적인 항산화
            음식이기도 하다.
        </p>
    </div>
    <div class="bg2">
        <p>
            <strong>토마토 : </strong> 토마토는 비타민A, C가 풍부할 뿐 아니라 희귀한
            성분인 비타민 H와 P까지
            포함된 과일이다. 칼륨과 같은 각종 미네랄은 혈액의 산성도를 낮추는 역할을
            해주며 토마토에 들어있는 피코펜이라는 성분은
            뛰어난 항암 작용을 보이며 잘 알려져 있듯이 블루베리와 함께 대표적인 항산화
            음식이기도 하다.
        </p>
    </div>
    <div class="bg3">
        <p>
            <strong>토마토 : </strong> 토마토는 비타민A, C가 풍부할 뿐 아니라 희귀한
            성분인 비타민 H와 P까지
            포함된 과일이다. 칼륨과 같은 각종 미네랄은 혈액의 산성도를 낮추는 역할을
            해주며 토마토에 들어있는 피코펜이라는 성분은
            뛰어난 항암 작용을 보이며 잘 알려져 있듯이 블루베리와 함께 대표적인 항산화
            음식이기도 하다.
        </p>
    </div>
</body>
</html>

```

## (8) background-attachment 속성 - 배경 이미지 고정하기

- 기본형: background-attachment: scroll | fixed
- 배경 이미지 고정 → 스크롤 막대를 움직이더라도 내용만 스크롤되므로 마치 배경 이미  
지 위에 내용이 떠 있는 것처럼 보임.
- scroll : 배경 이미지도 내용 따라서 스크롤 됨. 기본값
- fixed : 배경 이미지가 고정되어 스크롤 되지 않음

[07/bg-attachment.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>배경 이미지</title>
<style>
body {
    background-image: url('images/bg3.jpg');
    background-repeat: no-repeat;
    background-position: right bottom;
    background-attachment: fixed;
}

div {
    background: rgba(255, 255, 255, 0.6);
    width: 80%;
    height: auto;
}

```

```

margin-left: 10%;
}
</style>
</head>
<body>
    <div>
        <h2>블루베리(Blueberry)</h2>
        <p>블루베리의 대표적인 기능은 항산화로 비타민A, C, E가 풍부하고 안토니시아민, 페놀
등이 활성 산소를 없애
        노화를 억제한다. 이밖에 블루베리의 안토니시아민은 눈의 피로와 시력 저하를
회복시키는 효능을 가지고 있다.</p>
        <h2>토마토(Tomato)</h2>
        <p>토마토는 비타민A, C가 풍부할 뿐 아니라 희귀한 성분인 비타민 H와 P까지 포함된
과일이다. 칼륨과 같은 각종
        미네랄은 혈액의 산성도를 낮추는 역할을 해주며 토마토에 들어있는
피코펜이라는 성분은 뛰어난 항암 작용을 보이며 잘 알려져
있듯이 블루베리와 함께 대표적인 항산화 음식이기도 하다.</p>
        <h2>견과류 (Dry Nuts)</h2>
        <p>땅콩, 잣, 호두, 아몬드, 마카다미아, 피칸, 파스타치오 등 견과류에는 불포화지방산이
많이 들어있다.
        불포화지방산은 상온에서 고체인 포화지방산에 비해 인체 내부에서도 액체로
남아있을 수 있는 지방이다. 특히 아몬드는 많은
        미네랄과 식이 섬유를 함유하고 있으며 노화 방지에 좋은 비타민E가
풍부하다.</p>
    </div>
</body>
</html>

```

## (9) background 속성 - 속성 하나로 사용하기

- 배경과 관련된 속성을 한꺼번에 줄여서 표현

```

예) body {
    background: url(bg.jpg) center no-repeat fixed;
}

```

## 7.3 그라데이션 효과로 배경 꾸미기

### (1) 그라데이션 브라우저 접두사

- 현재의 표준화된 구문 외에도 지원하지 않는 브라우저를 위해 접두사를 붙여 입력해야 한다.

[07/linear-position.html] 그라데이션 브라우저 접두사

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>선형 그라데이션</title>
<style>
div {
    width: 500px;
    height: 300px;
    border-radius: 10px;
}

.grad {
    background: blue; /* css3 미지원 브라우저 */
    background: -webkit-linear-gradient(left top, blue, white); /* 초기 모던 브라우저 */
    background: -moz-linear-gradient(right bottom, blue, white);
    background: -o-linear-gradient(right bottom, blue, white);
    background: linear-gradient(to right bottom, blue, white);
}

```

```

}
</style>
</head>
<body>
    <div class="grad"></div>
</body>
</html>

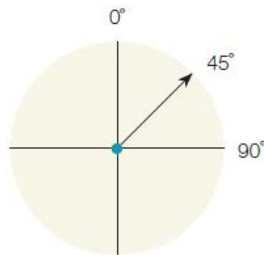
```

## (2) 선형 그라데이션

- 기본형: `linear-gradient (각도 또는 위치, color-stop, [color-stop,...]);`
- 수직 방향이나 수평 방향으로, 혹은 대각선 방향으로 색상이 일정하게 변하는 것
- 선형 그라데이션을 지정할 때는 방향과 색상이 필요하다.
- 위치:
  - `to` 와 함께 사용해서 끝나는 위치를 지정한다.
  - `to left` : 오른쪽에서 왼쪽으로, `to right` : 왼쪽에서 오른쪽으로,
  - `to bottom` : 위에서 아래로, `to top` : 아래에서 위로
- 각도:
  - 그라데이션의 각도는 선형 그라데이션 방향을 지정한다.
  - 맨 윗부분이 `0deg`이고, 시계 방향으로 회전하면서 `90deg`, `180deg`가 된다.
  - `270deg`는 `-90deg`와 같다.

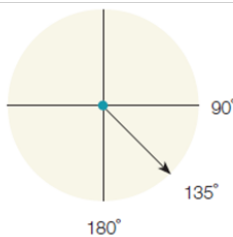
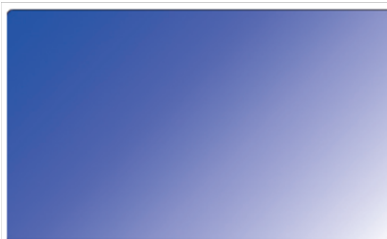
예)

```
linear-gradient(45deg, #ff0000, #ffffff) /* 45도, 빨간색, 흰색 */
```



```
background:-linear-gradient(-135deg, blue, white);
```

/\* 왼쪽 위에서 오른쪽 아래로, 파란색에서 흰색으로 \*/



```
background:-linear-gradient(to bottom, blue, white);  
/* 위에서 아래로 파란색에서 흰색으로 */
```



```
background: -linear-gradient(to bottom, #06f, white 30%, #06f)  
/* 위에서 아래로 시작 색은 #06f, 30% 위치에서 흰색으로 바뀌고 다시 #06f 색으로 바뀜 */
```



### (3) 원형 그라데이션

- 기본형: radial-gradient(위치, 모양, color-stop, [color-stop...])
- 원 또는 타원의 중심에서부터 동심원을 그리며 바깥 방향으로 색상이 바뀐다.
- 위치:
  - 시작 원(또는 타원)의 중심과 끝나는 원(또는 타원)의 크기
  - 시작하는 원의 중심
  - 키워드(가로:left, center, right, 세로: top, center, bottom) 또는 백분율.
  - 생략하면 center center로 인식
- 모양:
  - 그라데이션의 모양을 원(circle)과 타원(ellipse) 중에서 지정
- 크기:
  - closest-side : 그라데이션 가장자리가 그라데이션 중심에서 가장 가까운 요소의 모서리와 만난다.
  - closest-corner : 그라데이션 가장자리가 그라데이션 중심에서 가장 가까운 요소의 코너에 닿도록 한다
  - farthest-side : 그라데이션 가장자리가 그라데이션 중심에서 가장 멀리 떨어져 있는 모서리와 만날 때까지 그린다.
  - farthest-corner : 그라데이션의 가장자리가 그라데이션의 중심에서 가장 멀리 떨어져 있는 코너에 닿도록 한다. 기본 값.

예)

```
background: -prefix-radial-gradient(10% 10%, ellipse, white,  
blue);  
/* 10% 10%에서 시작하여 흰색에서 파란색으로 번지는 타원형 그라데이션 */
```



시작 위치 10% 10%, ellipse(타원 모양)

```
-prefix-radial-gradient(circle, yellow, green);
/* 노란색에서 초록색으로 번지는 원형 그라데이션 */
```



circle(원 모양)

#### (4) 그라데이션 반복

- repeating-linear-gradient: 선형 그라데이션을 반복한다.
- repeating-radial-gradient: 원형 그라데이션을 반복한다.

[07/repeat-gradient3.html] 그라데이션 반복

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>원형 그라데이션 반복</title>
<style>
div {
    width: 500px;
    height: 300px;
    border-radius: 10px;
    margin: 10px;
}

.grad1 {
    background: #ccc; /* css3 미지원 브라우저 */
    background: -webkit-repeating-radial-gradient(circle, white, #ccc 10%);
    /* 초기 모던 브라우저 */
    background: -moz-repeating-radial-gradient(circle, white, #ccc 10%);
    /* 초기 모던 브라우저 */
    background: -o-repeating-radial-gradient(circle, white, #ccc 10%);
    /* 초기 모던 브라우저 */
    background: repeating-radial-gradient(circle, white, #ccc 10%);
    /* 최신 모던 브라우저 */
}

.grad2 {
    background: #ccc; /* css3 미지원 브라우저 */
    background: -webkit-repeating-radial-gradient(circle, white, white 10%, #ccc 10%,
        #ccc 20%); /* 초기 모던 브라우저 */
    background: -moz-repeating-radial-gradient(circle, white, white 10%, #ccc 10%, #ccc
        20%); /* 초기 모던 브라우저 */
    background: -o-repeating-radial-gradient(circle, white, white 10%, #ccc 10%, #ccc
        20%); /* 초기 모던 브라우저 */
    background: repeating-radial-gradient(circle, white, white 10%, #ccc 10%, #ccc 20%);
}
```

```
        /* 최신 모던 브라우저 */
    }
</style>
</head>
<body>
    <div class="grad1"></div>
    <div class="grad2"></div>
</body>
</html>
```

## [과제] 연습문제

p302 ~ p303

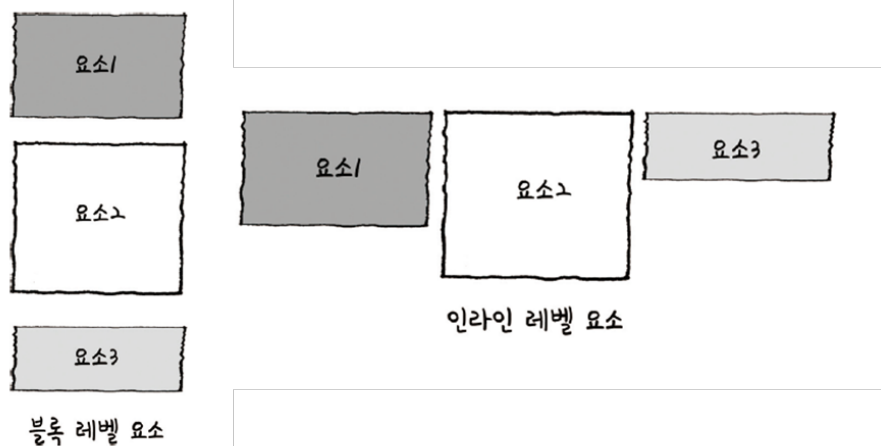
## 8. 레이아웃을 위한 스타일

### 8.1. CSS와 박스 모델

CSS의 박스 모델(box model)은 그림이나 텍스트처럼 사각 영역을 갖는 요소를 통칭한다. 박스 모델 스타일을 이용하면 여백이나 테두리 등을 자세히 설정할 수 있다. → 웹 문서에서 다양하게 활용

#### (1) 블록 레벨 요소와 인라인 레벨 요소

- 블록 레벨 (block-level) 요소
  - 요소를 삽입했을 때 혼자서 한 줄을 차지하는 요소
  - 해당 요소의 왼쪽이나 오른쪽에 다른 요소가 올 수 없다.
  - <p>, <h1>~<h6>, <ul>, <ol>, <div>, <blockquote>, <form>, <hr>, <table>, <fieldset>, <address>
- 인라인 레벨 (inline-level) 요소
  - 줄을 차지하지 않는 요소
  - 화면에 표시되는 콘텐츠만큼만 영역을 차지하고, 그 외의 공간에는 다른 요소가 와도 상관없다.
  - 한 줄에 여러 요소를 표시.
  - <img>, <object>, <br>, <sup>, <span>, <input>, <textarea>, <label>, <button>



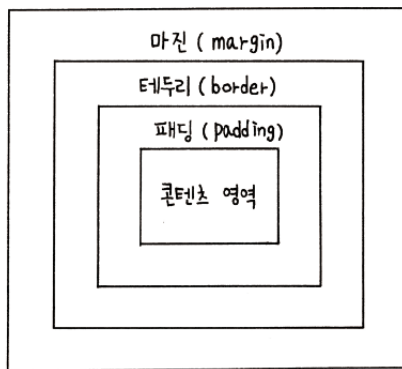
[08/inline.html]

```
<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>블록 레벨과 인라인 레벨</title>
</head>
<body>
    <h3>시간이란..</h3>
    <p>내일 죽을 것처럼 오늘을 살고</p>
    <p>
        영원히 살 것처럼 <span>내일을 꿈꾸어라.</span>
    </p>
</body>
</html>
```

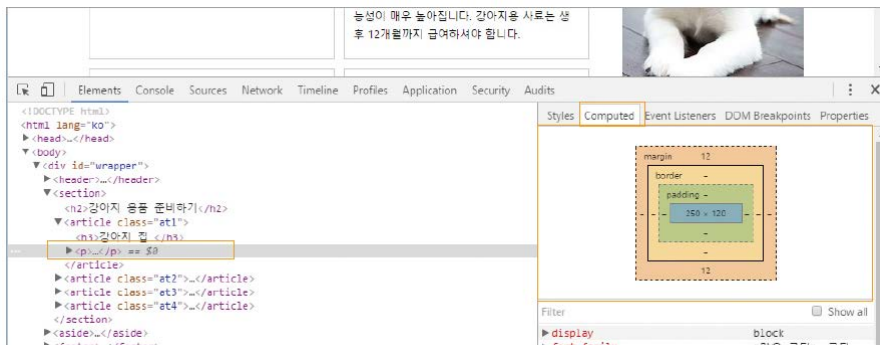
#### (2) 박스 모델(box model) - 박스 형태의 콘텐츠

- 웹 문서의 블록 레벨 요소들은 모두 박스 형태이다.



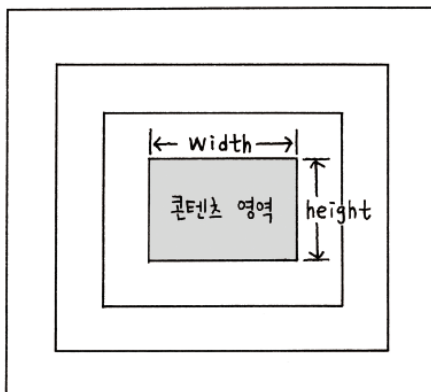


- 08/boxmodel.html 문서의 소스



### (3) width, height 속성 - 콘텐츠 영역의 크기

- 기본형: width: 속성 값; height: 속성 값;
- 박스 모델은 마진이나 패딩을 이용해 적절한 크기를 만든다.
- 필요하면 width 속성 / height 속성을 사용해 크기를 정할 수 있다.
- 사용할 수 있는 값은 길이 값이나 백분율(%)
- 따로 값을 지정하지 않으면 auto가 기본값  
(내용에 따라 너비를 자동으로 계산)



[08/width-height.html]

```
<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>박스모델</title>
<style>
.box1 {
    width: 200px; /* 고정 너비 */
}
```

```

        height: 100px; /* 높이 */
        background: #ff6a00; /* 배경색 */
    }

    .box2 {
        width: 50%; /* 가변 너비 - 브라우저 창 너비의 50% */
        height: 100px; /* 높이 */
        background: #0094ff; /* 배경색 */
    }

    div {
        margin: 10px; /* div 간의 여백 */
    }
</style>
</head>
<body>
    <div class="box1"></div>
    <div class="box2"></div>
</body>
</html>

```

#### (4) display 속성 - 화면 배치 방법 결정하기

- 기본형: display: none | contents | block | inline | inline-block | table | table-cell 등
- 해당 요소가 화면에 어떻게 보여질 것인지 지정한다.
- block 속성 값: 해당 요소를 블록 레벨로 지정한다. <img> 태그 같은 인라인 레벨 요소도 블록 레벨 요소로 바꿀 수 있다.
- inline 속성 값: 블록 레벨 요소를 인라인 레벨로 바꿀 수 있다.
- inline-block 속성 값: 요소를 인라인 레벨로 배치하면서 내용에는 블록 레벨 속성을 지정한다.
- none 속성 값: 해당 요소를 화면에 아예 표시하지 않는다.

```

예) ul li {
    display:inline; /* 항목을 가로로 배치 */
    float: left; /* 왼쪽부터 배치 */
}

```

[08/display-1.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>display 속성</title>
<style>
div {
    margin: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

#inline img {
    margin: 10px;
}

#block img {
    display: block;
    margin: 10px;
}
</style>

```

```

</head>
<body>
    <div id="inline">
          
    </div>
    <div id="block">
          
    </div>
</body>
</html>

```

[08/display-2.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>display 속성</title>
<style>
nav ul {
    list-style-type: none;
}

nav ul li {
    display: inline;
}

nav ul li a {
    color: black;
    text-decoration: none;
}
</style>
</head>
<body>
    <nav>
        <ul>
            <li><a href="#">애완견 종류</a></li>
            <li><a href="#">입양하기</a></li>
            <li><a href="#">건강돌보기</a></li>
            <li><a href="#">더불어살기</a></li>
        </ul>
    </nav>
</body>
</html>

```

[08/display-3.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>display 속성</title>
<style>
nav {
    width: 100%;
    height: 60px;
    background-color: #069;
}

nav ul {
    list-style-type: none;
}

```

```

}

nav ul li {
    display: inline-block;
    margin: 20px;
}

nav ul li a {
    color: white;
    text-decoration: none;
}

h2 {
}
</style>
</head>
<body>
    <nav>
        <ul>
            <li><a href="#">애완견 종류</a></li>
            <li><a href="#">입양하기</a></li>
            <li><a href="#">건강돌보기</a></li>
            <li><a href="#">더불어살기</a></li>
        </ul>
        <h2>애완견 종류</h2>
    </nav>
</body>
</html>

```

## 8.2. 테두리 관련 속성들

### (1) border-style 속성 - 테두리선 스타일 지정하기

- 기본형: border-style:속성 값;
- 예) border-style:dotted

[08/border-1.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>박스모델</title>
<style>
div {
    width: 200px;
    height: 100px;
    display: inline-block;
    margin: 15px;
    border-width: 5px; /* 테두리 굵기 */
}

.box1 {
    border-style: solid;
} /* 실선 */
.box2 {
    border-style: dotted;
} /* 점선 */
.box3 {
    border-style: dashed;
} /* 선으로 된 점선 */
</style>
</head>

```

```

<body>
    <div class="box1"></div>
    <div class="box2"></div>
    <div class="box3"></div>
</body>
</html>

```

## (2) border-width 속성 - 테두리 두께 지정하기

- 기본형: border-width: 키워드 | 크기값;
- 예) border-width: thin thin thin 5px;

## (3) border-color 속성 - 테두리 색상 지정하기

- 기본형: border-color: 속성 값;
- 예) border-color: red; /\* 빨간색 테두리 \*/

[08/border-2.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>박스모델</title>
<style>
div {
    width: 200px;
    height: 100px;
    display: inline-block;
    margin: 15px;
    border-style: dashed; /* 테두리 스타일 - 선으로 된 점선 */
    border-width: 2px; /* 테두리 굵기 - 2px */
}

.box1 {
    border-color: red;
} /* 색상 - 빨강 */
.box2 {
    border-color: blue;
} /* 색상 - 파랑 */
</style>
</head>
<body>
    <div class="box1"></div>
    <div class="box2"></div>
</body>
</html>

```

## (4) border 속성 - 테두리 스타일 묶어서 지정하기

- 방향에 따라서 두께와 색상, 스타일을 한꺼번에 묶어서 지정 (border-top, border-right, border-bottom, border-left, border)
- 두께와 색상, 스타일의 순서는 상관없다.
- 특정 방향에만 테두리를 표시하려고 할 때 편리

[08/border-3.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>박스모델</title>

```

```

<style>
h1 {
    padding-bottom: 5px;
    border-bottom: 3px solid #ccc; /* 아랫 부분 - 3px짜리 회색 실선 */
}

p {
    padding: 10px;
    border: 2px dotted black; /* 모든 방향 - 3px짜리 검정 점선 */
}
</style>
</head>
<body>
    <h1>박스 모델</h1>
    <p>박스 모델은 실제 콘텐츠 영역, 박스와 콘텐츠 영역 사이의 여백인 패딩(padding), 박스의 테두리(border), 그리고 여러 박스 모델 간의 여백인 마진(margin) 등의 요소로 구성되어 있습니다.</p>
</body>
</html>

```

## (5) border-radius 속성 - 박스 모서리 둥글게 만들기

- 기본형: border-radius:속성 값;
- 예) div {
 border-radius:20px; /\* 모서리 부분 반지름 20px \*/
 border:3px solid red; /\* 테두리 3px 두께의 빨간색 실선 \*/
 }

[08/radius-1.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>박스모델</title>
<style>
div {
    width: 300px; /* 너비 */
    height: 200px; /* 높이 */
    margin: 20px; /* 영역간 마진 */
    display: inline-block; /* 가로로 배치 */
}

.round {
    border: 2px solid red; /* 2px짜리 빨강 실선 */
    border-radius: 20px; /* 모서리 20px 만큼 라운딩 */
}

#bg {
    background: url(images/pic1.jpg) no-repeat; /* 배경 이미지 */
    background-size: cover; /* 영역을 다 채우게 */
}
</style>
</head>
<body>
    <div class="round"></div>
    <div class="round" id="bg"></div>
</body>
</html>

```

## (6) box-shadow 속성 - 선택한 요소에 그림자 효과 내기

- 기본형: box-shadow: 수평거리 값|수직거리 값|blur 값|확장거리 값|색상 값|inset;
- 예) div {  
    box-shadow: 2px 2px 5px 0 black;  
}

[08/box-shadow.html]

```
<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>박스모델</title>
<style>
div {
    width: 200px;
    height: 100px;
    display: inline-block;
    margin: 15px;
    border: 2px solid;
    border-radius: 20px;
}

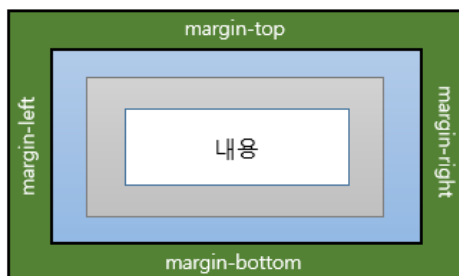
.box1 {
    box-shadow: 2px -2px 5px 0px black;
}

.box2 {
    box-shadow: 5px 5px 15px 5px gray;
}
</style>
</head>
<body>
    <div class="box1"></div>
    <div class="box2"></div>
</body>
</html>
```

## 8.3. 여백을 조절하는 속성들

### (1) margin 속성 - 요소 주변 여백 설정하기

- 요소 주변 여백
- margin-top, margin-right, margin-bottom, margin-left  
(요소의 바깥 위쪽 마진, 오른쪽 마진, 아래쪽 마진, 왼쪽 마진)
- margin 값이 4개라면 그 순서는 top -> right -> bottom -> left (시계방향) 이다.



예) p { margin: 30px 50px 30px 50px; } /\* 위아래 마진-30px, 좌우마진-50px \*/

[08/margin.html]

```

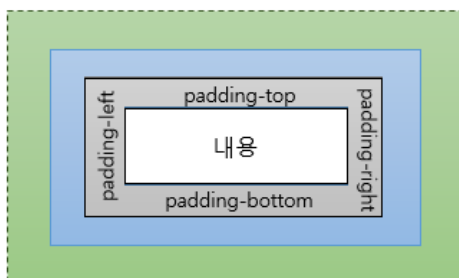
<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>박스모델</title>
<style>
div {
    width: 200px; /* 너비 */
    height: 100px; /* 높이 */
    background: #0094ff; /* 배경색 */
}

.box1 {
    margin: 30px 50px 30px 50px;
} /* 마진 - 30px 50px 30px 50px */
.box2 {
    margin: 30px 50px;
} /* 마진 - 30px 50px 30px 50px */
.box3 {
    margin: 50px;
} /* 마진 - 50px 50px 50px 50px */
.box4 {
    margin: 30px 5px 10px 5px;
} /* 마진 - 30px 5px 10px 5px */
</style>
</head>
<body>
    <div class="box1"></div>
    <div class="box2"></div>
    <div class="box3"></div>
    <div class="box4"></div>
</body>
</html>

```

## (2) padding 속성 - 콘텐츠 영역과 테두리 사이 여백 설정하기

- 콘텐츠 영역과 테두리 사이 여백
- padding-top, padding-right, padding-bottom, padding-left  
(박스의 위쪽 패딩과 오른쪽 패딩, 아래쪽 패딩, 왼쪽 패딩)
- padding (네 방향의 패딩을 한꺼번)



[08/padding.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>박스모델</title>
<style>
div {
    width: 200px; /* 너비 */
    height: auto; /* 높이 */
    background: #0094ff; /* 배경색 */
}

```



```

display: inline-block; /* 가로로 배치 */
margin: 15px; /* 마진 - 15px 15px 15px 15px */
color: white; /* 글자색 */
}

.box1 {
padding: 10px 30px 10px 30px;
} /* 패딩 - 10px 30px 10px */
.box2 {
padding: 10px 30px;
} /* 패딩 - 10px 30px 10px 30px */
.box3 {
padding: 10px;
} /* 패딩 - 10px 10px 10px 10px */
</style>
</head>
<body>
<div class="box1">패딩(padding)이란 콘텐츠 영역과 테두리 사이의 여백을 말합니다.</div>
<div class="box2">패딩(padding)이란 콘텐츠 영역과 테두리 사이의 여백을 말합니다.</div>
<div class="box3">패딩(padding)이란 콘텐츠 영역과 테두리 사이의 여백을 말합니다.</div>
</body>
</html>

```

## [실습] 박스 모델로 콘텐츠 정리하기

p332 ~ p334

## [과제] 연습문제

p335

## 9. CSS 포지셔닝

### 9.1. 여러 요소를 원하는 형태로 배치하기

#### (1) CSS 포지셔닝이란?

- 단순한 사이트부터 여러 단으로 구성된 사이트까지 웹 문서의 레이아웃을 만든다.

#### (2) box-sizing 속성 - 박스 너비 기준 정하기

- 기본형: box-sizing: content-box | border-box
- content-box 속성값은 width 속성 값을 콘텐츠 영역 너비 값으로 사용한다.(기본값)
- border-box 속성값은 박스 모델 전체 너비 값으로 사용한다.

[09/box-sizing.html]

```
<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>box-sizing</title>
<style>
.box1 {
    box-sizing: content-box; /* 콘텐츠 영역 기준 */
    width: 300px; /* 콘텐츠 영역 너비 300px */
    height: 150px; /* 높이 */
    margin: 10px; /* 마진 */
    padding: 30px; /* 패딩 */
    border: 2px solid red; /* 테두리 */
}

.box2 {
    box-sizing: border-box; /* 테두리까지(박스 전체) 기준*/
    width: 300px; /* 박스 모델 전체 너비 */
    height: 150px; /* 박스 높이 */
    margin: 10px; /* 마진 */
    padding: 30px; /* 패딩 */
    border: 2px solid red; /* 테두리 */
}
</style>
</head>
<body>

    <div class="box1">box-sizing = "content-box"</div>
    <div class="box2">box-sizing = "border-box"</div>

</body>
</html>
```

#### (3) float 속성 - 왼쪽이나 오른쪽으로 배치하기

- 기본형: float: left | right | none
- 웹 요소를 문서 위에 떠 있게 한다. 문서의 레이아웃 지정.
- float:left; - 문서의 왼쪽으로 배치
- float:right; - 문서의 오른쪽으로 배치

[09/float2.html]

```
<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>박스모델</title>
<style>
```

```

.box1 {
    padding: 20px;
    margin-right: 10px;
    background: #ffd800;
    float: left;
}

.box2 {
    background: #0094ff;
    padding: 20px;
    margin-right: 10px;
    float: left;
}

.box3 {
    background: #00ff21;
    padding: 20px;
    float: left;
}

.box4 {
    background: #ffffff;
    padding: 20px;
    float: right;
    border: 1px solid black;
}
</style>
</head>
<body>
    <div class="box1">박스1</div>
    <div class="box2">박스2</div>
    <div class="box3">박스3</div>
    <div class="box4">박스4</div>
</body>
</html>

```

#### (4) clear 속성 - float 속성 해제하기

- 기본형: clear: none | left | right | both
- float 속성을 이용해 웹 페이지 요소를 왼쪽이나 오른쪽에 배치하면 그 다음에 넣는 다른 요소들에도 똑 같은 속성이 전달된다. 따라서 float 속성이 더 이상 유용하지 않다고 아려 주는 속성이 필요하다.
- clear 속성에서 사용할 수 있는 값
  - left : 왼쪽으로 떠있지 않도록
  - right : 오른쪽으로 떠있지 않도록.
  - both : 왼쪽이나 오른쪽 어디에도 떠있지 않도록
  - none : 떠 있는 상태. 기본 값.

[09/float3.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>박스모델</title>
<style>
div {
    padding: 20px;
    margin: 10px;
}

.box1 {

```

```

        background: #ffd800;
        float: left;
    }

    .box2 {
        background: #0094ff;
        float: left;
    }

    .box3 {
        background: #00ff21;
    }

    .box4 {
        background: #a874ff;
        clear: both;
    }
</style>
</head>
<body>
    <div class="box1">박스1</div>
    <div class="box2">박스2</div>
    <div class="box3">박스3</div>
    <div class="box4">박스4</div>
</body>
</html>

```

## [실습] 2단 레이아웃 만들기 1 & 2

p345 ~ p351

### (5) position 속성 - 배치 방법 지정하기

- 기본형: position: static | relative | absolute | fixed
- 웹 문서 안의 요소들을 자유자재로 배치해 주는 속성
- 웹 문서를 만들 때 중요하게 사용하는 속성 중 하나
- 사용할 수 있는 값
  - static : 요소를 문서의 흐름에 맞춰 배치.
  - relative : 이전 요소에 자연스럽게 연결해서 배치.
  - absolute : 원하는 위치를 지정하여 배치.
  - fixed : 지정한 위치에 고정해서 배치하는데 요소가 화면에서 잘릴 수도 있다

[09/relative.html]

```

<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>CSS 포지셔닝</title>
<style>
    .box1 {
        float: left;
        width: 100px;
        background: #ffd800;
        margin-right: 10px;
        padding: 20px;
    }

    .box2 {
        position: relative;
        left: -50px;
        top: 30px;
    }

```

```

        width: 300px;
        background: #0094ff;
        float: left;
        padding: 20px;
    }
</style>
</head>
<body>
    <div class="box1">박스1</div>
    <div class="box2">박스2</div>
</body>
</html>

```

## (6) visibility 속성 - 요소를 보이게 하거나 보이지 않게 하기

- 특정 요소를 화면에 보이거나 보이지 않게 설정하는 속성

[09/visibility.html]

```

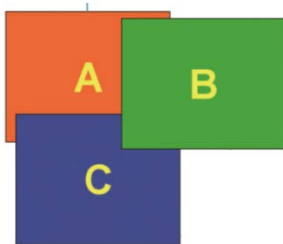
<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>visibility 속성</title>
<style>
img {
    margin: 10px;
    padding: 5px;
    border: 1px solid black;
}

.invisible {
    visibility: hidden;
}
</style>
</head>
<body>
    
    
    
</body>
</html>

```

## (7) z-index 속성 - 요소 쌓는 순서 정하기

- 요소 쌓는 순서 정하기
- z-index 값이 크면 값이 작은 요소보다 위에 쌓인다.
- z-index 값을 명시하지 않으면 1부터 시작해서 1씩 커진다.



[09/z-index.html]

```

<!doctype html>

```

```

<html lang="ko">
<head>
<meta charset="utf-8">
<title>z-index 속성</title>
<style>
div#wrapper {
    position: relative;
}

div#box {
    position: absolute;
    width: 100px;
    height: 100px;
    border: 1px solid black;
    font-size: 26px;
}

.b1 {
    left: 50px;
    top: 50px;
    background: #ff0000;
    z-index: 1;
}

.b2 {
    left: 110px;
    top: 70px;
    background: #ffd800;
    z-index: 3;
}

.b3 {
    left: 70px;
    top: 110px;
    background: #0094ff;
    z-index: 1;
}
</style>
</head>
<body>
    <div id="wrapper">
        <div id="box" class="b1">1</div>
        <div id="box" class="b2">2</div>
        <div id="box" class="b3">3</div>
    </div>
</body>
</html>

```

## 9.2. 다단으로 편집하기

### (1) column-width - 단의 너비 고정하고 다단 구성하기

- 단의 너비를 고정해 놓고 화면 분할
- 화면이 커지면 단의 개수가 많아지고 화면이 좁아지면 단의 개수가 줄어든다.

[09/col-width.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>HTML5 레이아웃</title>
<style>
body {

```

```

font-family: "맑은 고딕", "고딕", "굴림";
}

.multi {
  -webkit-column-width: 120px;
  -moz-column-width: 120px;
  column-width: 120px;
}
</style>
</head>
<body>
  <header>
    <h2>건강에 좋은 식품 - Super Food</h2>
  </header>
  <div class="multi">
    <p>
      <b>코코넛 오일 </b>: 남미 파나마에서는 “ 코코넛오일 한 컵이면 모든 병이
      낫는다 ” 라고 이야기할 정도로 그 효능이
      뛰어나다. 코코넛 오일에 들어있는 중사슬 지방산은 콜레스테롤 수치에 나쁜 영향을
      주지 않는 건강한 지방산이다. 또 라우르산이
      풍부해 세균을 물리치고 염증을 가라앉히는 효과가 있다. 하루 권장량은 성인기준
      2~4스푼(30~40mL) 정도이다.
    </p>
    <p>
      <b>블루베리 </b>: 블루베리는 우리 몸이 산화하는 현상을 막아주는 '항산화 물질'이
      가장 많은 식품으로 알려졌다. 미국
      농무부 인간영양연구소에서 밝힌 자료에 따르면 블루베리의 '안토시아닌'이라는
      물질은 시력 향상에 뛰어난 효과가 있다. 그뿐만
      아니라 다이어트에도 좋다. 블루베리의 폴리페놀 성분은 체내 지방세포를 녹여주는
      역할을 한다. 하루 권장량은 성인기준 50g
      전후로, 20~30알 정도이다.
    </p>
    <p>
      <b>아몬드 </b>: 혈관질환에 좋은 오메가3, 섬유소, 비타민 E가 풍부해 항산화 효과가
      높으며 콜레스테롤 수치를 낮추는
      L-아르기닌 등의 좋은 영양소도 들어 있다. 또한, 식욕을 감소시키고 뇌 내 행복감을
      촉진하는 세로토닌 분비를 증가시키는
      것으로 나타났다. 특히 껍질에 비타민 E, 플라보노이드, 카테킨 등의 항산화 성분이
      풍부하니 껍질째 먹는 것이 좋다. 하루
      권장량은 30g(23알 내외) 정도이며, 과도하게 섭취하면 장의 흡수기능이 저하되니
      주의해야 한다.
    </p>
  </div>
</body>
</html>

```

## (2) column-count 속성 - 단의 개수 고정하고 다단 구성하기

- 단의 개수를 정해 놓고 화면 분할
- 브라우저 창의 너비와 상관없이 단의 개수를 항상 일정하게 유지해야 하기 때문에 창의 너비가 커지면 단의 너비도 커진다.

[09/col-count.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>HTML5 레이아웃</title>
<style>
body {
  font-family: "맑은 고딕", "고딕", "굴림";
}

```

```

.multi {
    -webkit-column-count: 3;
    -moz-column-count: 3;
    column-count: 3;
}
</style>
</head>
<body>
    <header>
        <h2>건강에 좋은 식품 - Super Food</h2>
    </header>
    <div class="multi">
        <p>
            <b>코코넛 오일 </b>: 남미 파나마에서는 “ 코코넛오일 한 컵이면 모든 병이
            낫는다 ” 라고 이야기할 정도로 그 효능이
            뛰어났다. 코코넛 오일에 들어있는 중사슬 지방산은 콜레스테롤 수치에 나쁜 영향을
            주지 않는 건강한 지방산이다. 또 라우르산이
            풍부해 세균을 물리치고 염증을 가라앉히는 효과가 있다. 하루 권장량은 성인기준
            2~4스푼(30~40mL) 정도이다.
        </p>
        <p>
            <b>블루베리 </b>: 블루베리는 우리 몸이 산화하는 현상을 막아주는 '항산화 물질'이
            가장 많은 식품으로 알려졌다. 미국
            농무부 인간영양연구소에서 밝힌 자료에 따르면 블루베리의 '안토시아닌'이라는
            물질은 시력 향상에 뛰어난 효과가 있다. 그뿐만
            아니라 다이어트에도 좋다. 블루베리의 폴리페놀 성분은 체내 지방세포를 녹여주는
            역할을 한다. 하루 권장량은 성인기준 50g
            전후로, 20~30알 정도이다.
        </p>
        <p>
            <b>아몬드 </b>: 혈관질환에 좋은 오메가3, 섬유소, 비타민 E가 풍부해 항산화 효과가
            높으며 콜레스테롤 수치를 낮추는
            L-아르기닌 등의 좋은 영양소도 들어 있다. 또한, 식욕을 감소시키고 뇌 내 행복감을
            촉진하는 세로토닌 분비를 증가시키는
            것으로 나타났다. 특히 껍질에 비타민 E, 플라보노이드, 카테킨 등의 항산화 성분이
            풍부하니 껍질째 먹는 것이 좋다. 하루
            권장량은 30g(23알 내외) 정도이며, 과도하게 섭취하면 장의 흡수기능이 저하되니
            주의해야 한다.
        </p>
    </div>
</body>
</html>

```

### (3) column-gap 속성 - 단과 단 사이 여백 지정하기

- 둘 이상의 단으로 구성할 경우 단과 단 사이의 여백

### (4) column-rule 속성 - 구분선의 색상, 스타일, 너비 지정하기

- 단과 단 사이에 구분선 표시
  - column-rule-color : 구분선 색상
  - column-rule-style : 구분선 스타일
  - column-rule-width : 구분선 두께

### (5) break-after 속성 - 다단 위치 지정하기

- 다단을 나눌 위치: 특정 요소의 앞이나 뒤
  - break-before:column,
  - break-before:avoid-column
  - break-after:column,
  - bread-after:avoid-column



[09/col-break2.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>HTML5 레이아웃</title>
<style>
body {
    font-family: "맑은 고딕", "고딕", "굴림";
}

section {
    padding: 20px;
    width: 600px;
    margin: 20px auto;
    border: 1px solid #ccc;
    border-radius: 10px;
}

.multi {
    -webkit-column-count: 4;
    -moz-column-count: 4;
    column-count: 4;
    -webkit-column-rule: 2px dotted #000;
    -moz-column-rule: 2px dotted #000;
    column-rule: 2px dotted #000;
    text-align: justify;
}

.multi h3 {
    break-before: column;
}
</style>
</head>
<body>

    <section>
        <h2>건강에 좋은 식품 - Super Food</h2>
        <h3>코코넛 오일</h3>
        <p>남미 파나마에서는 “ 코코넛오일 한 컵이면 모든 병이 낫는다 ” 라고 이야기할 정도로 그
효능이 뛰어나다. 코코넛
오일에 들어있는 중사슬 지방산은 콜레스테롤 수치에 나쁜 영향을 주지 않는 건강한
지방산이다. 또 라우르산이 풍부해 세균을
물리치고 염증을 가라앉히는 효과가 있다. 하루 권장량은 성인기준 2~4스푼(30~40mL)
정도이다.</p>

        <div class="multi">
            <h3>아보카도</h3>
            <p>나무에서 열리는 버터라고 불릴 만큼 몸에 좋은 지방인 오메가 3와 단백질, 비타민
E와 같은 항산화 성분이
풍부한 과일이다. 알파 카로틴이 풍부하게 들어 있어 여러 항암 효과가
뛰어난 것으로 알려져 있다. 그뿐만 아니라 세포의
콜라겐 합성을 도와 피부 탄력을 높이는 데 도움을 준다. 하루 권장량은
성인기준 1개이며, 꾸준히 섭취하면 콜레스테롤
수치를 낮춰준다.</p>
            <h3>케일</h3>
            <p>케일은 브로콜리, 양배추, 유채, 청경채와 같은 배추과 식물로써 세계보건기구에서
최고의 녹황색 채소로
선정되었다. 베타카로틴과 엽록소가 풍부해 체내 유전자를 회복하도록 돕는
항암효과가 뛰어나다. 또한, 위염, 위궤양에 좋은
비타민 U와 유산균을 늘리는 비타민 M과 같은 식물성 영양소를 함유하고
있다. 하루 권장량은 성인기준 150g 정도이다.</p>
            <h3>블루베리</h3>
            <p>블루베리는 우리 몸이 산화하는 현상을 막아주는 '항산화 물질'이 가장 많은
식품으로 알려졌다. 미국 농무부
인간영양연구소에서 밝힌 자료에 따르면 블루베리의 '안토시아닌'이라는
```

물질은 시력 향상에 뛰어난 효과가 있다. 그뿐만 아니라 다이어트에도 좋다. 블루베리의 폴리페놀 성분은 체내 지방세포를 녹여주는 역할을 한다. 하루 권장량은 성인기준 50g 전후로, 20~30알 정도이다.

### 아몬드

혈관질환에 좋은 오메가3, 섬유소, 비타민 E가 풍부해 항산화 효과가 높으며 콜레스테롤 수치를 낮추는 L-아르기닌 등의 좋은 영양소도 들어 있다. 또한, 식욕을 감소시키고 뇌 내 행복감을 촉진하는 세로토닌 분비를 증가시키는 것으로 나타났다. 특히 껍질에 비타민 E, 플라보노이드, 카테킨 등의 항산화 성분이 풍부하니 껍질째 먹는 것이 좋다.

## (6) column-span 속성 - 여러 단을 하나로 합치기

- 값은 1 또는 all

## 9.3. 표 스타일

### (1) caption-side 속성 - 표 제목 위치 정하기

- 기본형: caption-side: top | bottom
- 표에 대한 제목은 <caption> 태그를 이용해 표시한다. caption-side 속성을 이용하면 캡션의 위치를 위쪽(top) 혹은 아래쪽(bottom)으로 옮길 수 있다.
- 예) caption { caption-side:bottom; } /\* 표 아래 캡션 표시 \*/

### (2) border 속성 - 표 테두리 스타일 지정하기

- 표의 셀 테두리 스타일 지정

[09/table-css1.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>표 스타일</title>
<style>
body {
    font-family: "맑은 고딕", "고딕", "굴림";
}

.table1 {
    border: 1px solid black;
}

.table1 td {
    border: 1px dotted black;
    padding: 10px;
    text-align: center;
}
</style>
</head>
<body>
    <table class="table1">
        <caption>프로축구 경기 일정</caption>
        <tr>
            <td>울산</td>
            <td>울산 vs 인천</td>
        </tr>
    </table>
</body>
```

```

        <tr>
            <td>부산</td>
            <td>부산 vs 대전</td>
        </tr>
        <tr>
            <td>서울</td>
            <td>서울 vs 강원</td>
        </tr>
    </table>
</body>
</html>

```

### (3) border-collapse 속성 - 테두리 통합, 분리하기

- 기본형: border-collapse: collapse | separate
- 표의 바깥 테두리와 셀의 각 테두리를 합칠 것인지 결정
  - collapse 테두리 통합.
  - separate 테두리 분리. 기본 값.

[09/table-css2.html]

```

<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>표 스타일</title>
<style>
body {
    font-family: "맑은 고딕", "고딕", "굴림";
}

.table1 {
    border: 1px solid black;
    border-collapse: collapse;
}

.table1 td {
    border: 1px dashed black;
    padding: 10px;
    text-align: center;
}
</style>
</head>
<body>
    <table class="table1">
        <caption>프로축구 경기 일정</caption>
        <tr>
            <td>울산</td>
            <td>울산 vs 인천</td>
        </tr>
        <tr>
            <td>부산</td>
            <td>부산 vs 대전</td>
        </tr>
        <tr>
            <td>서울</td>
            <td>서울 vs 강원</td>
        </tr>
    </table>
</body>
</html>

```

#### (4) border-spacing 속성 - 인접한 셀 테두리 사이 거리 지정하기

- 기본형: border-spacing: <크기>
- 테두리를 분리해서 표시할 경우 인접한 셀 테두리 사이의 거리
- 직접 크기와 단위를 사용해서 거리를 지정한다.

#### (5) empty-cells 속성 - 빈 셀의 표시 여부 지정하기

- 기본형: empty-cells: show | hide
- 테두리를 분리해서 표시할 경우 빈 셀의 표시 여부
  - show : 빈 셀 주위에 테두리를 그려 빈 셀 표시. 기본 값.
  - hide : 빈 셀 주위에 테두리를 그리지 않는다.

#### (6) width, height 속성 - 표 너비와 높이 지정하기

- 다른 웹 요소들의 너비와 높이를 지정할 때 처럼 표와 셀의 너비와 높이를 지정한다.

#### (7) table-layout 속성 - 콘텐츠에 맞게 셀 너비 지정하기

- 기본형: table-layout: fixed | auto
- 예) table { table-layout: fixed; } /\* 셀 너비 고정 \*/

#### (8) text-align 속성 - 셀 안에서 수형 정렬하기

- 기본형: text-align: left | right | center
- 텍스트의 가로 정렬 방법

#### (9) vertical-align 속성 - 셀 안에서 수직 정렬하기

- 기본형: vertical-align: top | bottom | middle
- 텍스트의 세로 정렬 방법

[09/table-align.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>Table CSS</title>
<style>
td {
    height: 100px;
    padding: 20px;
}

.va1 {
    vertical-align: top;
}

.va2 {
    vertical-align: bottom;
}

.va3 {
    vertical-align: middle;
}
</style>
</head>
<body>
    <table border="1">
        <caption>vertical-alignment</caption>
        <tr>
            <td class="va1">alignment</td>
```

```

<td class="va2">alignment</td>
<td class="va3">alignment</td>
</tr>
</table>
</body>
</html>
```

## [과제] 연습문제

p378

## 10. HTML5와 시맨틱 태그

### 10.1. HTML4 문서 vs HTML5 문서

#### (1) HTML4 문서

- HTML4에서는 HTML 태그가 문서 내용들을 화면에 보여주는 역할밖에 하지 못했기 때문에, 화면을 구성하는 역할은 대부분 `<div>` 태그에게 맡겨 놓았고, 수많은 `<div>` 태그들을 구분하는 것은 id 속성을 이용했다.

예) `<div id="header">`

`<!-- 헤더와 메뉴 -->`

`</div>`

`<div id="content">`

`<!-- 본문 내용 -->`

`</div>`

`<div id="footer">`

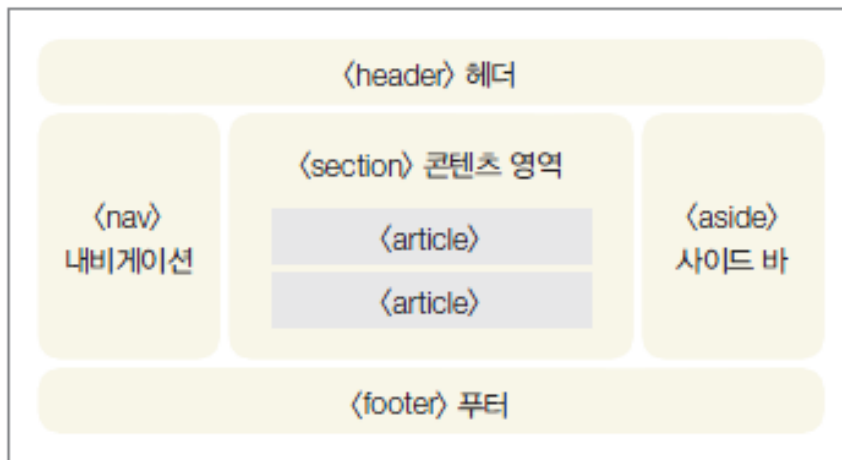
`<!-- 푸터 -->`

`</div>`

- 개발자가 문서구조를 파악하기 어렵고 화면 낭독기나 검색 엔진에서 웹 소스를 읽을 때 도 어느 부분이 메뉴이고 본문인지 구별할 수 없었다.

#### (2) 시맨틱 태그가 사용된 HTML5 문서

- HTML5에서는 웹 문서의 레이아웃을 어느 정도 표준화할 수 있겠다고 생각하고, 아래 그림에서 보는 것처럼 레이아웃을 만들 때 사용하는 주요 태그를 미리 약속해 놓고 해당 태그만 보면 어느 부분이 헤더이고 어느 부분이 실제 내용인지 쉽게 알 수 있도록 했다.



HTML5의 웹 표준 문서 구조

- 이런 요소들을 시맨틱 요소(semantic element)라고 하며 시맨틱(semantic)은 ‘의미가 통하는’이란 형용사로 각 요소가 스스로 뜻을 갖는다는 의미이다.

[10/html5doc.html]

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8">
<title>내가 처음 만드는 HTML 문서</title>
</head>
<body>
  <header>
    <h1>애완견 종류</h1>
    <nav>
```

```

        <ul>
            <li><a href="#">애완견 종류</a></li>
            <li><a href="#">입양하기</a></li>
            <li><a href="#">건강돌보기</a></li>
            <li><a href="#">더불어살기</a></li>
        </ul>
    </nav>
</header>
<section>
    <h2>활달한 강아지</h2>
    <article>
        ...
    </article>
    <article>
        ...
    </article>
</section>
<footer>
    <p>published by funnycom</p>
</footer>
</body>
</html>

```

### (3) 왜 시맨틱 태그로 레이아웃을 만들어야 할까?

- 웹 사이트의 본문 내용을 검색해야 한다면 <section>이나 <article> 태그 부분만 찾아 검색하면 된다.
- 화면 낭독기 같은 웹 보조 기구를 이용하는데 시맨틱 태그를 통해 사이트 제목과 내용을 정확히 전달할 수 있다. (웹 접근성)

## 10.2. 문서 구조를 위한 HTML5 시맨틱 태그들

### (1) <header> 태그 - 머리말(제목) 지정하기

- 사이트 전체의 제목 부분이 될 수도 있고, <article>의 제목 부분이 될 수도 있다.
- 기본형: <header> 내용 </header>

### (2) <nav> 태그 - 문서를 연결하는 내비게이션 링크

- 같은 사이트 안의 문서나 다른 사이트의 문서로 연결하는 링크를 나타낸다.
- 기본형: <nav> 내용 </nav>

### (3) <section> 태그 - 주제별 콘텐츠 영역 나타내기

- <header> 태그나 <footer> 태그와 비교해서 콘텐츠 영역을 구분짓는 용도
- <section> 태그 안에 또 다른 <section> 태그를 넣을 수도 있고, 실제 콘텐츠가 들어 있는 <article> 태그를 넣을 수도 있다.
- 기본형: <section> 내용 </section>

### (4) <article> 태그 - 실제 콘텐츠 내용 넣기

- 블로그의 포스트나 웹사이트의 내용, 사용자가 등록한 코멘트, 독립적인 웹 콘텐츠 항목이 여기에 해당된다.
- 독립적으로 배포하거나 재사용하더라도 완전히 하나의 콘텐츠가 된다면 <article> 태그를 쓴다.
- 기본형: <article> 내용 </article>

### (5) <aside> 태그 - 본문 이외의 내용 표시하기

- 본문 내용 외에 주변에 표시되는 기타 내용들을 나타낸다.
- 블로그 등에서 왼쪽이나 오른쪽, 혹은 하단에 표시되는 광고나 링크 같은 사이드 바를 표

- 시할 때 사용한다.
- 사이드 바는 필수 요소가 아니기 때문에 필요할 경우에만 사용한다.
- 기본형: <aside> 내용 </aside>

#### (6) <iframe> 태그 - 외부 문서 삽입하기

- 웹 문서 안에 다른 웹 문서를 가져와 표시한다.
- 기본형: <iframe src="삽입할 문서" [속성="속성값"]> </iframe>

#### (7) <footer> 태그 - 제작 정보와 저작권 정보 표시하기

- 사이트 제작자의 연락처 정보를 표시
- <footer> 태그 안에는 <header> 태그를 비롯하여 <section>, <article> 등 다른 레이아웃 태그 들을 모두 사용할 수 있고, 이런 태그를 이용해 푸터 안에 다양한 정보를 포함할 수 있다.
- 기본형: <footer> 내용 </footer>

#### (8) <address> 태그 - 사이트 제작자 정보, 연락처 정보 나타내기

- <footer> 태그 안에서 웹 페이지 제작자의 이름이나 제작자의 웹 페이지, 또는 피드백을 위한 이메일 주소 같은 연락처 정보를 넣는 데 사용된다.
- 실제 우편 주소를 넣기 위한 태그는 아니다. → 실제 우편물 주소는 <p> 태그 사용
- 최근 업데이트 날짜 같은 정보를 입력할 때도 사용해서는 안된다.
- 기본형: <address> 내용 </address>

#### (9) <div> 태그는 언제 사용할까?

- 콘텐츠를 묶어 시각적인 효과를 적용하려고 할 때 즉, 콘텐츠에 CSS를 적용할 때 사용한다.

[직접 해보세요!] HTML5 시맨틱 태그로 웹 표준에 맞는 문서 만들기, 스타일 시트(CSS) 적용해 보기

### 10.3. IE8 이하 버전에서는 어떻게 하나요?

#### (1) 새로운 시맨틱 태그 지원 상황

- HTML5의 새로운 시맨틱 태그는 대부분의 웹 브라우저에서는 사용할 수 있지만 IE8에서는 지원하지 않고 오페라 미니에서는 일부만 지원된다.

#### (2) IE8 이하에서 시맨틱 태그를 사용하려면?

- CSS에서 블록 레벨로 정의하기
  - 자신만의 영역을 가질 수 있는 블록 레벨 태그로 바뀌어 준다.
  - 예) header, section, nav, article, footer {  
display: block;  
}
- 시맨틱 태그 직접 정의하기
  - 브라우저에서 태그들을 이해할 수 있도록 자바스크립트를 이용해 태그들을 만들어 준다.
  - 예) <script>  
document.createElement( ' article ' );  
document.createElement( ' section ' );  
document.createElement( ' aside ' );  
document.createElement( ' nav ' );  
document.createElement( ' header ' );  
document.createElement( ' footer ' );  
...



</script>

- HTML5 Shiv 사용하기

- 자바스크립트를 이용해 HTML5 태그를 직접 정의해서 사용하는 것이 번거롭기 때문에 이런 과정을 간단하게 자바스크립트 파일로 만들어서 다운로드할 수 있게 해놓은 것이 HTML5 Shiv 이다.

예) <!-- [if lt IE 9]>

<script src="js/html5shiv.js"></script>

<![endif] -->

## 11. HTML5와 멀티미디어

### 11.1. 웹과 멀티미디어

#### (1) 플러그인 프로그램

- 웹에서 음악을 듣거나 동영상을 보기 위해서 별도의 플러그인을 이용 (윈도우 미디어 플레이어나 플래시 플레이어, 실버라이트 등) → 플러그인은 설치하기 번거롭거나 아예 불가능한 경우도 있어서 스마트폰이나 태블릿 PC 등에서 동영상을 보거나 음악을 듣지 못하는 경우가 많다.

#### (2) <object> 태그와 <embed> 태그 - 플러그인 사용하기

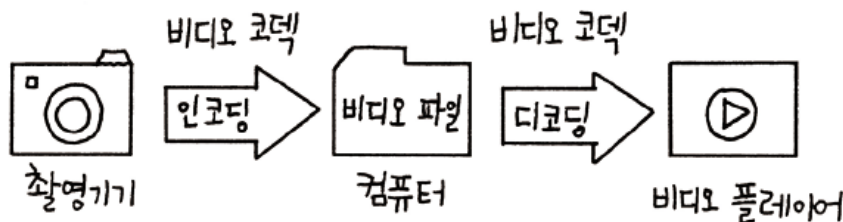
- 아직까지 HTML5를 지원하지 못하는 브라우저가 많기 때문에 플러그인 방법도 알아두어야 한다. HTML5에서도 웹문서안에 플러그인을 포함시킬 수 있다.
- <object> 태그는 동영상 파일 뿐만 아니라 플래시 무비, 자바 애플릿 등 다양한 미디어를 넣을 수 있다.
- <embed> 태그는 <object> 태그를 지원하지 않는 이전 브라우저에서 사용된다.

예) <object>

```
<param name="fileName" value="without.wmv"></param>
</object>
<embed src="spring.swf">
```

#### (3) 멀티미디어의 웹 표준화

- 웹 브라우저가 설치된 모든 기기에 적용되는 웹 표준이므로 플러그인 프로그램 없이 웹 브라우저 자체에서 멀티미디어를 재생할 수 있다. 다만, 아직까지 웹 브라우저마다 재생할 수 있는 파일 종류가 다르기 때문에 어떤 파일 형식이 표준이 될 것인지가 논의되는 중이다.
- 인코딩 : 전문 촬영 장비뿐만 아니라 캠코더나 휴대폰 등 여러 장치를 이용해 비디오를 촬영한 후 원본 비디오를 최대한을 압축해서 컴퓨터에서 사용할 수 있는 비디오 파일로 변환하는 과정
- 디코딩 : 비디오 파일에 저장되어 있는 비디오 정보를 가져와서 비디오 플레이어에 보여 주는 과정
- 코덱 : 인코딩과 디코딩을 수행하는 것
- HTML5와 비디오 코덱: H.264/AVC, v8/v9, 오그 테오라(Ogg Theora)
- HTML5와 오디오 코덱: MPEG-1 AUDIO Layer3, Ogg Vorbis



#### (4) HTML5 비디오 변환하기

- mp4, ogv, webm 형식으로 파일을 준비하면 대부분의 최신 브라우저에서 직접 비디오를 재생할 수 있다.
- 카카오인코더로 webm 파일과 mp4 파일로 변환하기
- 팟인코더로 ogv 파일로 변환하기

### 11.2. 오디오 & 비디오 재생하기

- mp3 파일과 oga 파일, ogg 파일을 갖추고 있어야 한다.

### (1) <audio> 태그 - 오디오 파일 삽입하기

- HTML5에서 배경 음악이나 효과음 등 오디오 삽입.
- 기본형: <audio src= "오디오 파일 경로" [속성] [속성= "속성값" ]></audio>

속성	설명
autoplay	오디오를 자동 재생합니다.
controls	웹 화면에 컨트롤 막대를 표시합니다. 컨트롤 막대에는 재생/멈춤, 진행 바, 볼륨 등이 표시됩니다.
loop	오디오를 반복 재생합니다.
muted	오디오를 재생해 진행하지만 소리는 끕니다.
preload	재생 버튼을 눌러 재생하기 전에 오디오 파일을 다운로드해 준비해 둡니다.

### (2) <video> 태그 - 비디오 파일 삽입하기

- mp4 라는 비디오 파일 형식은 크롬이나 사파리 브라우저에서만 재생할 수 있다.
- 파이어폭스에서 열어보면 비디오가 나타나지 않는다.
- 아직까지 HTML5 <video> 태그에서 사용할 파일 형식의 표준이 정해지지 않았다.
- 예) <video src="gate.mp4" width="450" height="300" controls>

### (3) <source> 태그 - 여러 미디어 파일 한꺼번에 지정하기

- 브라우저에 따라 지원하는 비디오 코덱이 다르기 때문에 <source> 태그를 이용해 각 브라우저에서 지원하는 파일들을 따로 지정한다.

```
<source src="video.ogv" type="video/ogg; codecs='theora,vorbis'" >
```

속성	설명
src	미디어 파일의 경로를 지정하는 필수 속성으로 파일 경로를 지정할 때는 경로에 공백이 있으면 안 됩니다.
type	웹 브라우저가 해당 미디어 파일을 재생할 수 있는지 여부를 확인할 수 있도록 미디어 파일의 유형을 알려 줍니다.
codecs	비디오 코덱을 지정합니다.

[11/video3.html]

```
<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>비디오 재생하기</title>
<style>
body {
    margin-top: 20px;
    margin-left: 50px;
}
</style>
</head>
<body>

    <video controls>
        <source src="media/Painting.ogv" type="video/ogg">
        <source src="media/Painting.mp4" type="video/mp4">
        <source src="media/Painting.webm" type="video/webm">
    </video>

</body>
</html>
```

#### (4) 이전 브라우저에서는 어떻게 해야 할까?

- <video> 태그를 지원하지 않는 이전 브라우저를 고려해 HTML5 지원 브라우저가 필요하다 는 대체 텍스트를 표시할 수 있다.
- 플래시 무비로 변환한 후, <embed> 태그나 <object> 태그를 사용해 플러그인으로 삽입할 수도 있다.

```
<video controls>
  <source src="media/Painting.mp4" type="video/mp4">
  <source src="media/Painting.webm" type="video/webm">
  <source src="media/Painting.ogv" type="video/ogg">
  이 영상을 보기 위해서는 HTML5를 지원하는 브라우저가 필요합니다.
</video>

<video controls>
  <source src="media/Painting.mp4" type="video/mp4">
  <source src="media/Painting.webm" type="video/webm">
  <source src="media/Painting.ogv" type="video/ogg">
  <object data="media/Painting.swf" type="application/x-shockwave-flash"></object>
</video>
```

#### (5) <audio> 태그와 <video> 태그의 속성

- width, height: 비디오 크기 조절
- controls: 미디어 파일에 컨트롤 막대 표시, 표시되는 컨트롤 막대는 웹 브라우저마다 다르며 사용할 수 있는 명령도 조금씩 다르다.
- preload: 파일 다운로드 여부, 즉 재생하기 전에 비디오 파일을 모두 다운로드할 것인지 또는 일부 정보만 다운로드할 것인지 여부를 지정한다. (none, metadata, auto)
- muted: 소리는 끄고 화면만 재생
- autoplay: 자동 재생, 비디오 파일을 다운로드하자마자 자동으로 재생한다. 모바일 기기에서는 자동 재생되지 않는다.
- loop: 반복 재생, 속성 값 없이 loop 라고 하면 된다.
- poster: 문제 상황 표시

#### [실습] 웹 문서에 멀티미디어 파일 넣기

p424 ~ p426

#### (6) <track> 태그 - 비디오 화면에 자막 추가하기

- HTML5에서는 <track> 태그를 이용해서 외부 자막 파일을 연결할 수 있다.
- 기본형: <track kind="자막종류" src="경로" srclang="언어" label="제목" default>

#### [실습] 동영상으로 홈페이지 대문 만들기

p436 ~ p438

## 12. 다재다능한 CSS3 선택자

### 12.1. 연결 선택자(combination selector)

#### (1) 하위 선택자 - 지정한 모든 하위 요소에 스타일 적용하기

- 기본형: 상위요소 하위요소
- 부모 요소에 포함된 모든 하위 요소에 스타일이 적용된다.
- 예) section p { color: blue; }

#### (2) 자식 선택자 - 자식 요소에만 스타일 적용하기

- 기본형: 부모요소 > 자식요소

- 하위 요소 전체에 스타일을 적용하는 것이 아니라 자식요소에만 스타일을 적용할 수 있다.
- 예) `section > p { color: blue; }`

### (3) 인접 형제 선택자 - 가장 가까운 형제 요소에 스타일 적용하기

- 기본형: `요소1 + 요소2`
- 문서 구조상 같은 부모를 가진 형제 요소 중 첫 번째 동생 요소에만 적용된다.
- 예) `h1 + p { text-decoration: underline; }`

### (4) 형제 선택자 - 형제 요소에 스타일 적용하기

- 기본형: `요소1 ~ 요소2`
- 모든 형제 요소에 다 적용된다.
- 예) `h1 ~ p { text-decoration: underline; }`

## 12.2. 속성 선택자

### (1) [속성] 선택자 - 지정한 속성에 스타일 적용하기

- 기본형: `[속성]`
- 지정한 속성을 가지고 있는 요소를 찾아 스타일을 적용한다. ‘[ ‘ 와 ‘ ] ’ 사이에 찾고자 하는 속성을 지정한다.
- 예) `a [href] { background: yellow; }`

### (2) [속성 = 값] 선택자 - 특정 값을 갖는 속성에 스타일 적용하기

- 예) `input [type="text"] {  
    width: 150px;  
    background-color: skyblue;  
    border: 1px solid #000;  
}`

### (3) [속성 ~= 값] 선택자 - 여러 값 중 특정 값이 포함된 속성에 스타일 적용하기

- 선택된 값은 일부만 일치하는 것이 아니라 한 단어로 일치해야 한다. 예를들면 하이픈(-)으로 연결한 단어에 스타일을 적용되지 않는다.

[12/attr3.html]

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>속성 선택자</title>
<style>
ul {
    list-style: none;
}

li {
    display: inline;
    float: left;
    margin: 10px;
}

li a {
    padding: 5px 20px;
    font-size: 14px;
    color: blue;
```

```

        text-decoration: none;
    }

    .flat {
        background: blue;
        color: white;
    }

    [class~="button"] {
        border: 2px solid black;
        box-shadow: rgba(0, 0, 0, 0.4) 5px 5px;
    }
</style>
</head>

<body>
    <ul>
        <li><a href="#">메뉴 1</a></li>
        <li><a href="#">메뉴 2</a></li>
        <li><a href="#" class="button">메뉴 3</a></li>
        <li><a href="#" class="flat button">메뉴 4</a></li>
    </ul>
</body>
</html>

```

#### (4) [속성 != 값] 선택자 - 특정 값이 포함된 속성에 스타일 적용하기

- 속성 값이 지정한 값이거나 “ 값- ” 으로 시작하면 스타일을 적용한다. 다시 말해 하이픈 (-)으로 연결한 단어가 있더라도 스타일을 적용한다.

#### (5) [속성 ^= 값] 선택자 - 특정 값으로 시작하는 속성에 스타일 적용하기

- 캐럿(^)이 붙으면 지정한 문자로 시작하는 속성 값에 대해서만 스타일을 적용한다.

#### (6) [속성 \$= 값] 선택자 - 특정 값으로 끝나는 속성에 스타일 적용하기

- 예) a [href \$= ".zip" ] {background:url(zip\_icon.gif)}

#### (7) [속성 \*= 값] 선택자 - 값의 일부가 일치하는 속성에 스타일 적용하기

기본형	스타일 적용 경우	예시 - 선택자	예시 - 적용되는 요소
[속성]	지정한 '속성'일 때	[href]	<a href="#">메뉴 1</a>
[속성 = 값]	'값'에 일치할 때	[target="_blank"]	<a href="....." target="_blank">HTML</a>
[속성 ~= 값]	여러 값들 중 특정 '값'이 포함되어 있을 때(단어별)	[class~="button"]	<a href="#" class="flat button">메뉴 4</a>
[속성 != 값]	'값'이 포함되어 있을 때 (하이픈 포함 단어별)	[class != "us"]	<a href="#" title="us">영어</a> <a href="#" title="us-english">영어</a>
[속성 ^= 값]	'값'으로 시작할 때	[title^="eng"]	<a href="#" title="english">영어</a>
[속성 \$= 값]	'값'으로 끝날 때	[href \$="xls"]	<a href="intro.xls">엑셀 파일</a>
[속성 *= 값]	'값'이 속성 값의 일부일 때	[href *= "w3"]	<a href="http://www.w3c.org/TR/html">HTML 사이트</a>

### 12.3. 가상 클래스와 가상 요소

메뉴에서 몇 번째 항목이라든가 단락의 첫 번째 글자 등의 경우, 클래스 이름 앞에 콜론(:)을 붙여 표시하는 가상 클래스와 클래스 이름 앞에 콜론 두 개(::)를 붙여 표시하는 가상 요소를 사용한다.

슈도 클래스(pseudo class)라고도 하는데, 웹 문서의 소스에는 실제로 존재하지 않지만 필요에 의해 임의로 가상의 선택자를 지정하여 사용하는 것을 말한다.

표기	설명
:link	링크
:visited	사용자가 방문했던 링크
:active	대상 요소가 활성화되었을 때. 클릭했을 때 등
:hover	사용자가 대상 요소를 가리키고 있을 때. 롤오버(rollover) 등
:focus	대상 요소가 포커스 되었을 때
:root	문서 안의 루트 요소에 스타일을 적용한다.
:nth-child(n)	부모 요소의 n번째의 자식 요소에 스타일을 적용한다.
:nth-last-child(n)	부모 요소의 뒤로부터 n번째의 자식 요소에 스타일을 적용한다.
:nth-of-type(n)	같은 유형의 요소 중에서 n 번째 나타나는 요소에 스타일을 적용한다.
:nth-last-of-type(n)	같은 유형의 요소 중에서 끝에서부터 세어 n번째 나타나는 요소에 스타일을 적용한다.
:first-child	첫번째 자식 요소에 스타일을 적용한다.
:last-child	마지막 자식 요소에 스타일을 적용한다.
:first-of-type	형제 중에서 첫번째 요소에 스타일을 적용한다.
:last-of-type	형제 중에서 마지막 요소에 스타일을 적용한다.
:only-child	자식 요소가 하나 뿐인 부모 요소에 스타일을 적용한다.
:only-of-type	해당 요소가 하나 뿐인 요소에 스타일을 적용한다.
::before	특정 요소의 내용 앞에 지정한 내용을 만든다.
::after	특정 요소의 내용 뒤에 지정한 내용을 만든다.
::selection	사용자가 드래그해서 선택하는 영역의 스타일을 지정한다.
:not( )	괄호 안에 있는 요소를 제외한 요소에 스타일을 적용한다.

```
예) a:link { color:black; text-decoration:none; }  
    a:visited { color:black; }  
    a:active { color:black; }  
    a:hover { color:red; text-decoration:underline; }
```

#### (1) 사용자 동작에 반응하는 가상 클래스

- :link 가상 클래스 선택자 - 방문하지 않은 링크에 스타일 적용
- :visited 가상 클래스 선택자 - 방문한 링크에 스타일 적용
- :hover 가상 클래스 선택자 - 웹 요소에 마우스 커서를 올려놓을 때의 스타일 적용
- :active 가상 클래스 선택자 - 웹 요소를 활성화했을 때(누르고 있을 때)의 스타일 적용
- :focus 가상 클래스 선택자 - 웹 요소에 초점이 맞추어졌을 때의 스타일 적용
- [주의] :link, :visited, :hover, :active 선택자 순서대로 정의해야 한다. 순서가 바뀌면 정의하더라도 제대로 적용되지 않는다.

#### (2) UI 요소 상태에 따른 가상 클래스

- :enabled와 :disabled 가상 클래스 선택자 - 요소를 사용할 수 있을 때와 없을 때의 스타일 적용
- :checked 가상 클래스 선택자 - 라디오 박스나 체크 박스에서 해당 항목을 선택했을 때의 스타일 지정

[12/states.html]

```
<!doctype html>
<html lang="ko">
<head>
<title>가상 클래스</title>
<style>
body {
    background-color: #fff;
}

form fieldset {
    margin-bottom: 25px;
}

form legend {
    font-size: 15px;
    font-weight: 600;
}

input.disabled {
    background: #ddd;
    border: 1px #ccc solid;
}

input:checked+span {
    color: blue;
}
</style>
</head>
<body>
    <form>
        <fieldset>
            <legend>사용자 정보</legend>
            <label>이름 <input type="text" disabled/></label>
        </fieldset>
        <fieldset>
            <legend>신청 과목</legend>
            <p>이 달에 신청할 과목을 선택하세요</p>
            <label><input type="radio" name="subject"
value="speaking"><span>회화</span></label>
            <label><input type="radio" name="subject"
value="grammar"><span>문법</span></label>
            <label><input type="radio" name="subject"
value="writing"><span>작문</span></label>
        </fieldset>
    </form>
</body>
</html>
```

### (3) 구조 가상 클래스

웹 문서 구조를 기준으로 특정 위치에 있는 요소를 찾아 스타일을 지정할 때 사용한다.

- :root 가상 클래스 선택자 - 문서 전체에 적용하기
- :nth-child(n)와 :nth-last-child(n) 가상 클래스 선택자 - 자식 요소의 위치에 따라 스타일 적용하기
- :nth-of-type(n), :nth-last-of-type(n) 가상 클래스 선택자 - 특정 태그 위치에 스타일 적용하기
- :first-child, :last-child 가상 클래스 선택자 - 첫 번째, 마지막 요소에 스타일 적용하기
- :first-of-type, :last-of-type 가상 클래스 선택자 - 형제 관계 요소의 위치에 따라



스타일 적용하기

- :only-child, :only-of-type 가상 클래스 선택자 - 하나뿐인 자식 요소에 스타일 적용하기

#### (4) 그 외 가상 클래스

- :target 가상 클래스 선택자 - 앵커 목적지에 스타일 적용하기  
예) #intro:target { background-color: yellow; }
- :not 가상 클래스 선택자 - 특정 요소가 아닐 때 스타일 적용하기  
예) p:not(#ex) { color: blue; }

#### (5) 가상 요소

가상 클래스 선택자가 여러 태그 중에서 원하는 태그를 선택하기 위해 사용하는 선택자라면 가상 요소는 내용의 일부만 선택해 스타일을 적용할 때 사용한다.

- ::first-line 요소와 ::first-letter 요소 - 첫 번째 줄, 첫 번째 글자에 스타일 적용하기
- ::before, ::after 요소 - 내용의 앞뒤에 콘텐츠 추가하기  
예) p::before {  
    content: "내용: ";  
}

### [과제] 연습문제

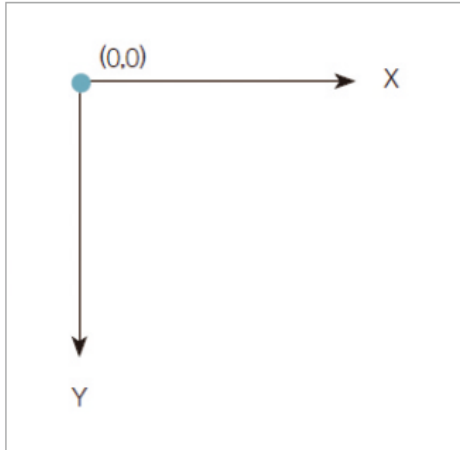
p467

## 13. CSS3와 애니메이션

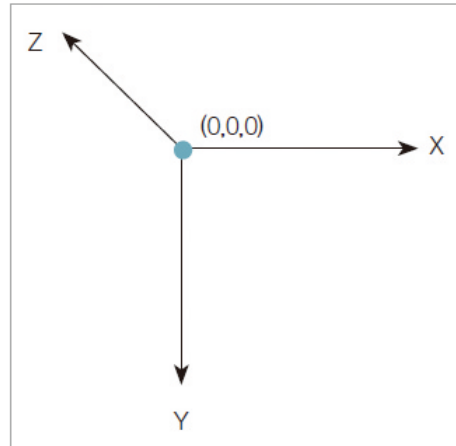
### 13.1. 변형

#### (1) 2차원 변형과 3차원 변형

- 2차원 변형(2D transform): 단순히 수평이나 수직으로 이동하고 회전하는 것을 말한다.
- 3차원 변형(3D transform): x축과 y축에 원근감을 주는 z축을 추가해 변형시키는 것을 말한다.



2차원 좌표계



3차원 좌표계

#### (2) transform과 변형 함수

- 기본형: transform: 변형함수;
- 예) .photo { transform: translate(50px,100px); }
- 2차원 변형함수: translate(x,y), translateX(tx), translateY(ty) 등
- 3차원 변형함수: translate3d(tx,ty,tz), rotateX(각도), rotateY(각도) 등

#### (3) translate 변형 함수 - 요소 이동시키기

- 기본형: transform:translate(tx,ty)  
transform:translate3d(tx,ty,tz)

#### (4) scale 변형 함수 - 요소 확대/축소하기

- 기본형: transform:scale(sx,sy)  
transform:scale3d(sx,sy,sz)

#### (5) rotate 변형 함수 - 요소 회전하기

- 기본형: transform:rotate(각도)  
transform:rotate3d(rx,ry,rz,각도)

#### (6) skew 변형 함수 - 요소를 비틀어 왜곡하기

- 기본형: transform:skew(ax,ay)  
transform:skewX(ax)  
transform:skewY(ay)

### 13.2. 변형과 관련된 속성들

#### (1) transform-origin 속성 - 변형 기준점 설정하기

- 기본형: transform-origin: <x축> <y축> <z축> | initial | inherit ;

속성 값	설명
<x축>	원점 기준의 x 좌표값으로 길이 값이나 <백분율>, left, center, right 중에서 사용할 수 있습니다.
<y축>	원점 기준의 y 좌표값으로 길이 값이나 <백분율>, top, center, bottom 중에서 선택할 수 있습니다.
<z축>	원점 기준의 z 좌표값으로 길이 값만 사용할 수 있습니다.

## (2) perspective, perspective-origin 속성 - 원근감 표현하기

- 기본형: perspective: <크기> | none;  
perspective-origin: <x축 값> | <y축 값>

속성 값	설명
<크기>	원래 위치에서 사용자가 있는 방향으로 얼마나 이동하는지를 픽셀 크기로 지정합니다.
none	perspective를 지정하지 않습니다. 기본 값입니다.★

속성 값	설명
<x축 값>	웹 요소가 x축에서 어디에 위치하는지를 지정합니다. 사용할 수 있는 값은 길이 값이나 백분율, left, right, center입니다. 기본 값은 50%입니다.
<y축 값>	웹 요소가 y축에서 어디에 위치하는지를 지정합니다. 사용할 수 있는 값은 길이 값이나 백분율, top, center, bottom입니다. 기본 값은 50%입니다.

## (3) transform-style 속성 - 3D 변형 적용하기

- 기본형: transform-style: flat | preserve-3d

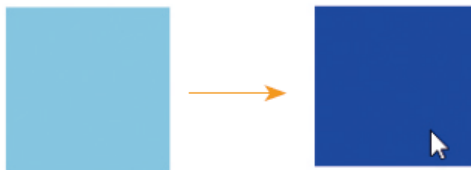
## (4) backface-visibility 속성 - 요소의 뒷면 표시하기

- 기본형: backface-visibility: visible | hidden

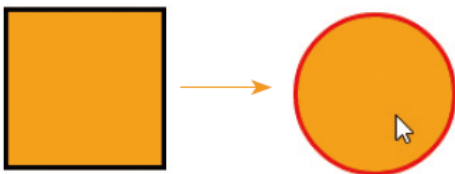
## 13.3. 트랜지션

### (1) 트랜지션이란?

- 웹 요소의 배경 색이 바뀌거나 도형의 테두리가 원형으로 바뀌는 것처럼 스타일 속성이 바뀌는 것을 말한다.



사각형의 배경 색이 바뀌는 트랜지션



사각형의 모양과 테두리 색이 바뀌는 트랜지션

## (2) transition-property 속성 - 트랜지션을 적용할 속성 지정하기

- 기본형: transition-property: all | none | <속성 이름>

```
transition-property:all; /* 해당 요소의 모든 속성에 트랜지션 적용 */
transition-property:background-color; /* 해당 요소의 배경 색에 트랜지션 적용 */
transition-property:width, height; /* 해당 요소의 너비와 높이에 트랜지션 적용 */
```

### (3) transition-duration 속성 - 트랜지션 진행 시간 지정하기

- 기본형: transition-duration: <시간>

### (4) transition-timing-function 속성 - 트랜지션 속도 곡선 지정하기

- 기본형: transition-timing-function: linear | ease | ease-in | ease-out | ease-in-out | cubic-bezier(n,n,n,n)

속성 값	설명
linear	시작부터 끝까지 똑같은 속도로 트랜지션을 진행합니다.
ease	처음에는 천천히 시작하고 점점 빨라지다가 마지막에는 천천히 끝납니다. 기본 값입니다.★
ease-in	시작을 느리게 합니다.
ease-out	느리게 끝납니다.
ease-in-out	느리게 시작하고 느리게 끝납니다.
cubic-bezier(n,n,n,n)	베지에 함수를 직접 정의해 사용합니다. n에서 사용할 수 있는 값은 0~1입니다.

[13/tr-function.html]

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Transition timing function</title>
<!--[if lt IE 9]>
<script src="html5shiv.js"></script>
<![endif]-->
<style>
#ex div {
    float: left;
    width: 100px;
    height: 50px;
    margin: 5px 10px;
    padding: 5px;
    color: white;
    background-color: #006aff;
    border-radius: 5px;
    text-align: center;
    font-weight: bold;
}

#ex:hover div {
    height: 400px;
}

#ex .ease {
    -moz-transition: 3s ease;
    -o-transition: 3s ease;
    -webkit-transition: 3s ease;
    transition: 3s ease;
}

#ex .linear {
    -webkit-transition: 3s linear;
    -moz-transition: 3s linear;
    -o-transition: 3s linear;
    transition: 3s linear;
}

#ex .ease-in {
```

```

        -webkit-transition: 3s ease-in;
        -moz-transition: 3s ease-in;
        -o-transition: 3s ease-in;
        transition: 3s ease-in;
    }

    #ex .ease-out {
        -webkit-transition: 3s ease-out;
        -moz-transition: 3s ease-out;
        -o-transition: 3s ease-out;
        transition: 3s ease-out;
    }

    #ex .ease-in-out {
        -webkit-transition: 3s ease-in-out;
        -moz-transition: 3s ease-in-out;
        -o-transition: 3s ease-in-out;
        transition: 3s ease-in-out;
    }
</style>
</head>

<body>
    <div id="ex">
        <div class="ease">ease</div>
        <div class="linear">linear</div>
        <div class="ease-in">ease-in</div>
        <div class="ease-out">ease-out</div>
        <div class="ease-in-out">ease-in-out</div>
    </div>
</body>
</html>

```

## (5) transition-delay 속성 - 지연 시간 설정하기

- 기본형: transition-delay: <시간>

## (6) transition 속성 - 트랜지션 속성 한꺼번에 표기하기

- 기본형: transition: <transition-property 값> | <transition-duration 값> | <transition-timing-function 값> | <transition-delay 값>

## 13.4. 애니메이션

### (1) CSS와 애니메이션

- CSS 애니메이션은 시작해 끝나는 동안 원하는 곳 어디서든 스타일을 바꾸며 애니메이션을 정의할 수 있다.
- 키프레임(keyframe) : 애니메이션 중간에 스타일이 바뀌는 지점
- CSS 애니메이션에서 사용하는 속성

속성	설명
@keyframes	애니메이션이 바뀌는 지점을 설정합니다.
animation-delay	애니메이션 지연 시간을 지정합니다.
animation-direction	애니메이션 종료 후 처음부터 시작할지, 역방향으로 진행할지를 지정합니다.
animation-duration	애니메이션 실행 시간을 설정합니다.
animation-fill-mode	애니메이션이 종료되었거나 지연되어 애니메이션이 실행되지 않는 상태일 때 요소의 스타일을 지정합니다.
animation-iteration-count	애니메이션 반복 횟수를 지정합니다.
animation-name	@keyframes로 설정해 놓은 중간 상태의 이름을 지정합니다.
animation-play-state	애니메이션을 멈추거나 다시 시작합니다.
animation-timing-function	애니메이션의 속도 곡선을 지정합니다.
animation	animation 하위 속성들을 한꺼번에 묶어 지정합니다.

## (2) @keyframes 속성 - 애니메이션 지점 설정하기

- 기본형: @keyframes <이름> {  
                                   <선택자> { <스타일> }  
                                   }

## (3) animation-name 속성 - 애니메이션 이름 지정하기

- 기본형: animation-name: <키프레임 이름> | none

## (4) animation-duration 속성 - 애니메이션 실행 시간 설정하기

- 기본형: animation-duration: <시간>

## (5) animation-direction 속성 - 애니메이션 방향 지정하기

- 기본형: animation-direction: normal | alternate

## (6) animation-iteration-count 속성 - 반복 횟수 지정하기

- 기본형: animation-iteration-count: <숫자> | infinite

## (7) animation-timing-function 속성 - 애니메이션 속도 곡선 지정하기

- 기본형: animation-timing-function: linear | ease | ease-in | ease-out | ease-in-out | cubic-bezier(n,n,n,n)

## (8) animation 속성 - 애니메이션 관련 속성 한꺼번에 표기하기

- 기본형: animation: <animation-name> | <animation-duration> | <animation-timing-function> | <animation-delay> | <animation-iteration-count> | <animation-direction>

[13/animation.html]

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Transform</title>
<style>
  .box {
    width: 60px;
    height: 60px;
    margin: 60px;
```

```

-webkit-animation: rotate 1.5s infinite, background 1.5s infinite alternate;
-moz-animation: rotate 1.5s infinite, background 1.5s infinite alternate;
-o-animation: rotate 1.5s infinite, background 1.5s infinite alternate;
animation: rotate 1.5s infinite, background 1.5s infinite alternate;
}

@-webkit-keyframes rotate {
  from {
    transform: perspective(120px) rotateX(0deg) rotateY(0deg);
  }

  50% {
    transform: perspective(120px) rotateX(-180deg) rotateY(0deg);
  }

  to {
    transform: perspective(120px) rotateX(-180deg) rotateY(-180deg);
  }
}

@-moz-keyframes rotate {
  from {
    transform: perspective(120px) rotateX(0deg) rotateY(0deg);
  }

  50% {
    transform: perspective(120px) rotateX(-180deg) rotateY(0deg);
  }

  to {
    transform: perspective(120px) rotateX(-180deg) rotateY(-180deg);
  }
}

@-o-keyframes rotate {
  from {
    transform: perspective(120px) rotateX(0deg) rotateY(0deg);
  }

  50% {
    transform: perspective(120px) rotateX(-180deg) rotateY(0deg);
  }

  to {
    transform: perspective(120px) rotateX(-180deg) rotateY(-180deg);
  }
}

@keyframes rotate {
  from {
    transform: perspective(120px) rotateX(0deg) rotateY(0deg);
  }

  50% {
    transform: perspective(120px) rotateX(-180deg) rotateY(0deg);
  }

  to {
    transform: perspective(120px) rotateX(-180deg) rotateY(-180deg);
  }
}

@-webkit-keyframes background {
  from {
    background: red;
  }
}

```

```

        50% {
            background-color: green;
        }

        to {
            background-color: blue;
        }
    }

    @-moz-keyframes background {
        from {
            background: red;
        }

        50% {
            background-color: green;
        }

        to {
            background-color: blue;
        }
    }

    @-o-keyframes background {
        from {
            background: red;
        }

        50% {
            background-color: green;
        }

        to {
            background-color: blue;
        }
    }

    @keyframes background {
        from {
            background: red;
        }

        50% {
            background-color: green;
        }

        to {
            background-color: blue;
        }
    }
</style>
</head>

<body>
    <div class="box"> </div>
</body>
</html>

```

## [실습] 상품 페이지에 가격 떠오르게 만들기

p503 ~ p506

## [과제] 연습문제

p507 ~ p508



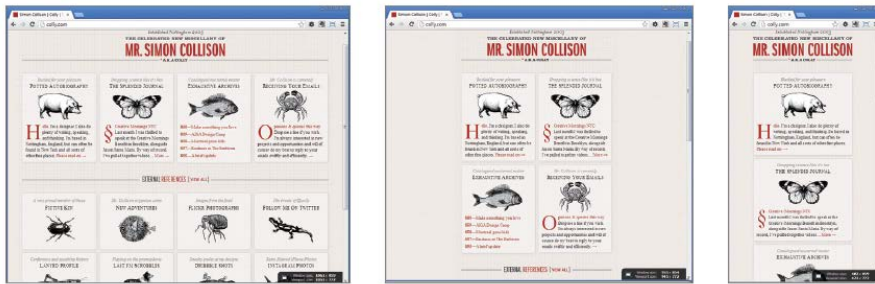


## 14. 반응형 웹이란?

### 14.1. 모바일 기기와 웹 디자인

#### (1) 반응형 웹 디자인

- 스마트폰이나 태블릿, 스마트 TV처럼 다양해지는 사용자 브라우저 환경에 따라 그때마다 웹사이트를 따로 제작하는 데는 한계가 있다.
- 여러 크기의 브라우저 창에 맞게 사이트를 따로 제작하는 일은 비효율적입니다
- 원래 웹사이트 내용을 그대로 유지하면서 다양한 화면 크기에 맞게 웹사이트를 표시하도록 해 보자 → 반응형 웹 디자인(responsive web design)
- 반응형 웹 디자인은 화면 크기에 맞게 화면 요소들을 재배치하고 각 요소의 표시 방법만 다르게 해서 사이트를 구현



반응형 웹 디자인(<http://colly.com>)

#### (2) 반응형 웹의 장단점

- 모든 스마트 기기에서 접속 가능
- 가로 모드에 맞추어 레이아웃 변경 가능
- 사이트 유지 관리 용이
- 최신 브라우저에서만 지원된다. 하위 버전의 브라우저 사용자에게 고려할 수 없다는 단점도 있다.

#### (3) 모바일 기기를 위한 기본 다지기, 뷰포트

- 뷰포트란 스마트폰 화면에서 실제 내용이 표시되는 영역이다.

#### (4) 뷰포트 지정하기

- 320px 너비로 맞춰 모바일 사이트를 제작해도 스마트폰을 보면 아주 작게 표시됨  
→ 모바일 브라우저의 기본 뷰포트 너비 980px  
→ 웹 페이지를 무조건 980px 너비로 표시하려고 하기 때문
- 해결방안: 뷰포트 크기나 배율을 조절해야 한다
- 기본형: `<meta name="viewport" content="<속성1=값>,<속성2=값>,...">`

속성	설명	사용 가능한 값	기본 값
width	뷰포트 너비	device-width 또는 크기	브라우저 기본 값
height	뷰포트 높이	device-height 또는 크기	브라우저 기본 값
user-scalable	확대/축소 가능 여부	yes 또는 no	yes
initial-scale	초기 확대/축소 값	1~10	1
minimum-scale	최소 확대/축소 값	0~10	0.25
maximum-scale	최대 확대/축소 값	0~10	1.6

예) `<meta name="viewport" content="width=device-width, initial-scale=1">`

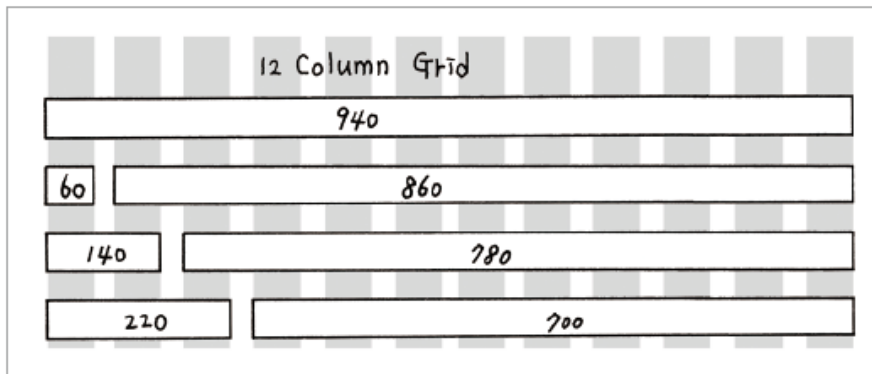
## [실습] 크롬의 디바이스 모드 활용하기

p516 ~ p519

### 14.2. 가변 그리드 레이아웃

#### (1) 고정 그리드와 가변 그리드

- 픽셀 단위를 사용했을 때의 단점을 극복하기 위해 미디어 쿼리를 이용해 기기 해상도별로 CSS 따로 정의, 그러나 기기별로 레이아웃이 달라질 수 있다.



- 어느 기기에서나 동일한 레이아웃 유지하려면 -> 가변 레이아웃(Fluid Layout)
  - 사이트 레이아웃을 픽셀 단위가 아닌 %로 지정한다.
  - 브라우저 너비 값이 바뀔 때마다 웹 요소의 너비 값도 함께 바뀐다.

#### (2) 가변 그리드 레이아웃 만들기

- 고정 그리드 레이아웃을 만들고 가변 그리드 레이아웃으로 바꾼다.
- 전체를 감싸는 div 추가하기
  - 가장 바깥에 새로운 <div> 태그를 추가하고 id 지정
  - 각 요소들의 너비를 %(백분율)로 바꿀 때 기준이 되는 너비가 필요하기 때문
  - 웹 문서 내용 전체의 크기나 배경색 등을 한꺼번에 조절할 수 있다.
  - 브라우저 화면 크기에 상관없이 웹 문서의 내용을 중앙에 배치할 수 있다.
- 예) #wrapper {  
width:96%;  
margin:0 auto;  
}  
<div id="wrapper">  
...  
</div>
- 각 요소의 너비를 백분율(%)로 바꾸기
  - 바깥의 #wrapper의 너비 값을 백분율 값으로 변환
  - #wrapper 크기를 기준으로 나머지 요소들의 너비를 %로 계산  
(요소의 너비 / 콘텐츠 전체를 감싸는 요소의 너비) \* 100  
예) article 600px → 600px / 960px \* 100 = 62.5%
  - 소수점 이하 숫자가 많다면 간단하게 소수점 이하 한두 자리로 표시
  - 약간씩의 오차를 고려하여 실제 계산한 값보다 약간 작게 지정하는 것이 좋다.

[14/fluid-grid.html] 가변 그리드 레이아웃 만들기

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Fluid Grid Layout</title>
```

```

<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
    #wrapper {
        width:96%;
        margin:0 auto;
    }
    header { /* 헤더 */
        width: 100%;
        height: 120px;
        background-color: #066cfa;
        border-bottom: 1px solid black;
    }
    .header-text{
        font-size:32px;
        color:white;
        text-align:center;
        line-height:120px;
    }
    .content { /* 본문 */
        float:left;
        width:62.5%;
        height:400px;
        padding:1.5625%;
        background-color:#ffd800;
    }
    .right-side { /* 사이드 바 */
        float:right;
        width:31.25%;
        height:400px;
        padding:1.5625%;
        background-color:#00ff90;
    }
    footer { /* 푸터 */
        clear:both;
        width:100%;
        height:120px;
        background-color:#c3590a;
    }
}
</style>
</head>

<body>
    <div id="wrapper">
        <header>
            <h1 class="header-text">가변 그리드 레이아웃</h1>
        </header>
        <section class="content">
            <h4>본문</h4>
        </section>
        <aside class="right-side">
            <h4>사이드바</h4>
        </aside>
        <footer>
            <h4>푸터</h4>
        </footer>
    </div>
</body>
</html>

```

### 14.3. 가변 레이아웃과 가변 요소

#### (1) 가변 글꼴

- em 단위

- 부모 요소에서 지정한 폰트의 대문자 M의 너비를 1em으로 지정한 것으로 1em은 16px이다. (1em = 16px)
- 텍스트 크기를 픽셀(px) 단위로 지정하면 크기가 고정되기 때문에 화면 크기가 작은 기기에서는 매우 작게 표시된다.
- em 단위를 사용하면 해당 기기에 맞춘 픽셀 크기로 계산되어 표시된다.
- rem 단위
  - em 단위는 부모 요소의 글꼴을 기준으로 하기 때문에 중첩된 부모 요소의 글자 크기의 영향을 받는다. -> em 수치가 계속 달라진다는 단점이 있다.
  - r은 루트(root)를 뜻하며 처음부터 기본 크기를 지정하기 때문에 중간에 기본 값이 바뀌지 않는다.

[14/fluid-font3.html] 가변 글꼴

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Fluid Grid Layout</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
    font-size: 16px;
}

#wrapper {
    width: 96%;
    margin: 0 auto;
    font-size: 12px;
}

.header-text {
    font-size: 2rem;
    text-align: center;
}

.fluid-text {
    font-size: 1.5rem;
}
</style>
</head>

<body>
    <div id="wrapper">
        <header class="header-text">
            <p>가변 그리드 레이아웃</p>
            <!-- 루트 요소가 16px 이므로 2em=32px -->
            <p class="fluid-text">가변 폰트</p>
            <!-- 루트요소가 16px 이므로 1.5em=24px -->
        </header>
    </div>
</body>
</html>
```

## (2) 가변 이미지

- CSS 이용하기
  - 예) .content img {
    - max-width:100%;
    - height:auto;

- <img> 태그에서 srcset 속성을 이용하면 화면 너비 값이나 픽셀 밀도에 따라 고해상도의 이미지 파일을 지정할 수 있다.  
예) 
- <picture> 태그와 <source> 태그 - 상황별로 다른 이미지 표시하기

[14/fluid-img4.html] 가변 글꼴

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Fluid Image</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
</head>

<body>
  <picture>
    <source srcset="images/shop-large.jpg" media="(min-width:1024px)">
    <source srcset="images/shop-medium.jpg" media="(min-width:768px)">
    <source srcset="images/shop-small.jpg" media="(min-width:320px)">
    
  </picture>
</body>
</html>
```

### (3) 가변 비디오

- 화면의 너비가 달라질 때마다 비디오의 너비가 늘어나거나 줄어들 수 있도록 해야 한다.  
예) video {  
    max-width: 100%;  
}

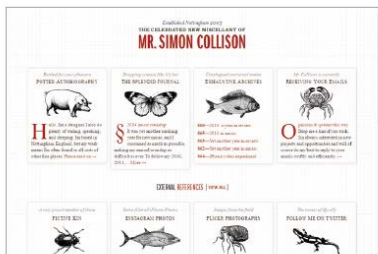
## [실습] 가변 그리드 레이아웃 구성하기

p532 ~ p539

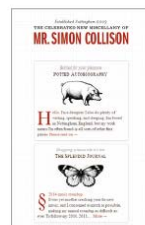
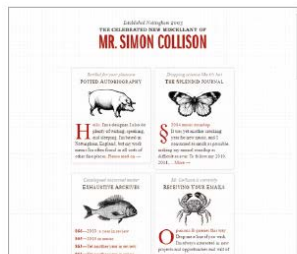
## 14.4. 미디어 쿼리

### (1) 미디어 쿼리란?

- 사이트에 접속하는 장치에 따라 특정한 CSS 스타일을 사용하도록 해 주는 기능이다.
- 브라우저 창의 너비를 조절할 때마다 화면에 표시되는 칼럼 개수가 달라짐
- PC나 태블릿, 스마트폰의 웹 브라우저 화면 크기에 따라 사이트 레이아웃이 바뀜.



colly.com



### (2) 미디어 쿼리 구문

- @media [ONLY | NOT] 미디어 유형 [AND 조건] \* [AND 조건]

연산자	설명
and	조건을 계속 추가할 수 있습니다.
,(선표)	동일한 스타일 유형을 사용할 미디어의 유형과 조건이 있다면 선표를 이용해 추가합니다.
only	미디어 쿼리를 지원하는 웹 브라우저에서만 조건을 인식하게 합니다. 이 키워드를 사용하면 미디어 쿼리를 지원하지 않는 웹 브라우저에서는 미디어 쿼리를 무시하고 실행하지 않습니다. IE에서는 미디어 쿼리를 제대로 인식하지 못하기 때문에 only 키워드를 사용하더라도 큰 의미가 없습니다.
not	not 다음에 지정하는 미디어 유형을 제외합니다. 예를 들어 'not tv' 라고 지정한다면 TV를 제외한 미디어 유형에만 적용합니다.

- 미디어 유형의 종류

미디어 유형	사용 가능한 미디어
all	모든 미디어 유형
print	인쇄 장치
screen	컴퓨터 스크린(스마트폰 스크린 포함)
tv	음성과 영상이 동시 출력되는 TV
aural	음성 합성 장치(주로 화면을 읽어 소리로 출력해 주는 장치)
braille	점자 표시 장치
handheld	패드(pad)처럼 손에 들고 다니는 장치
projection	프로젝터
tty	디스플레이 기능이 제한된 장치(픽셀(px) 단위를 사용할 수 없음)
embossed	점자 프린터

```
@media screen and (min-width:200px) and (max-width:360px) {
...
}
```

### (3) 미디어 쿼리의 조건

웹 문서의 가로 너비와 세로 높이: 뷰포트의 너비와 높이

가로, 세로 값 설정하는 속성	설명
width, height	웹 페이지의 가로 너비, 세로 높이
min-width, min-height	최소 너비, 최소 높이
max-width, max-height	최대 너비, 최대 높이

[14/mq.html]

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Media Queries</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
    background: url(images/bg0.jpg) no-repeat fixed;
    background-size: cover;
}
```

```

@media screen and (max-width:1024px) {
    body {
        background: url(images/bg1.jpg) no-repeat fixed;
        background-size: cover;
    }
}

@media screen and (max-width:768px) {
    body {
        background: url(images/bg2.jpg) no-repeat fixed;
        background-size: cover;
    }
}

@media screen and (max-width:320px) {
    body {
        background: url(images/bg3.jpg) no-repeat fixed;
        background-size: cover;
    }
}
</style>
</head>

<body>

</body>
</html>

```

## 단말기의 가로 너비와 세로 높이: 뷰포트의 너비와 높이

단말기의 가로, 세로 값을 설정하는 속성	설명
device-width, device-height	단말기의 가로 너비, 세로 높이
min-device-width, min-device-height	단말기의 최소 너비, 최소 높이
max-device-width, max-device-height	단말기의 최대 너비, 최대 높이

```

@media all and (min-device-width:320px) and (min-device-height:480px) {
    ...
}

```

## 화면 회전 : 디바이스를 세로로 또는 가로로 보기

속성	설명
orientation: portrait	단말기 세로 방향
orientation: landscape	단말기 가로 방향

[14/mq-orientation.html]

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Media Queries</title>
<meta name="viewport" content="width=device-width, initial-scale=1">

```



```

<style>
body {
    background-color: #eee;
}

@media screen and (orientation:landscape) {
    body {
        background-color: orange;
    }
}

@media screen and (orientation:portrait) {
    body {
        background-color: yellow;
    }
}
</style>
</head>

<body>
    <h1>Media Query</h1>
</body>
</html>

```

화면 비율 : 브라우저 화면 너비 값(width)을 높이 값(height)으로 나눈 것

단말기의 물리적 화면 비율 : 단말기 너비 값(device-width)을 높이로 나눈 것

색상당 비트 수 : 단말기에서 사용하는 최대 색상 비트 수

미디어 쿼리 중단점 만들기

- 중단점(breakpoint) : 서로 다른 CSS를 적용할 화면 크기
- 대부분 기기의 화면 크기 기준.
- 모든 기기를 반영할 수 없기 때문에 스마트폰과 태블릿, 데스크톱 정도로 구분
- 모바일 퍼스트(mobile first) : 모바일 기기 레이아웃을 기본으로 작성 → 태블릿 & PC 레이아웃 작성
- 미디어 쿼리 중단점은 개발자나 작업 조건에 따라 달라질 수 있다.

```

/* 스마트폰 세로 */
@media only screen and (min-width: 320px) { ..... }

/* 스마트폰 가로 */
@media only screen and (min-width: 480px) { ..... }

/* 태블릿 세로 */
@media only screen and (min-width: 768px) { ..... }

/* 태블릿 가로 / 데스크톱 */
@media only screen and (min-width: 1024px) { ..... }

```

## 14.5. 미디어 쿼리를 사용해 사이트 구성하기

### (1) 외부 CSS 파일 연결하기

- 각 조건별로 스타일시트 파일을 따로 저장한 후, <link> 태그나 @import 문을 사용해서 CSS 파일 연결한다.
- 기본형: <link href="css 파일 경로" rel="stylesheet" type="text/css" media="조건">

예) <link href="css/tablet.css" rel="stylesheet" type="text/css" media="screen and (min-width:321px) and (max-width:768px)">

## (2) 웹 문서에서 직접 정의하기

- <style> 태그 안에서 media 속성을 사용하여 조건과 그에 맞는 스타일 정의
- 기본형: <style media="미디어쿼리 조건">  
스타일 규칙들

```
</style>
```

```
<style>  
    @media 미디어쿼리 조건 {  
        스타일 규칙들  
    }  
</style>
```

```
<style media="screen and (max-width:320px)">  
    body {  
        background-color: orange;  
    }  
</style>  
  
<style>  
    @media screen and (max-width:320px) {  
        body {  
            background-color: orange;  
        }  
    }  
</style>
```

## [실습] 모바일용 문서 작성하기

p554 ~ p556

## [실습] 태블릿용 미디어 쿼리 작성하기

p556 ~ p559

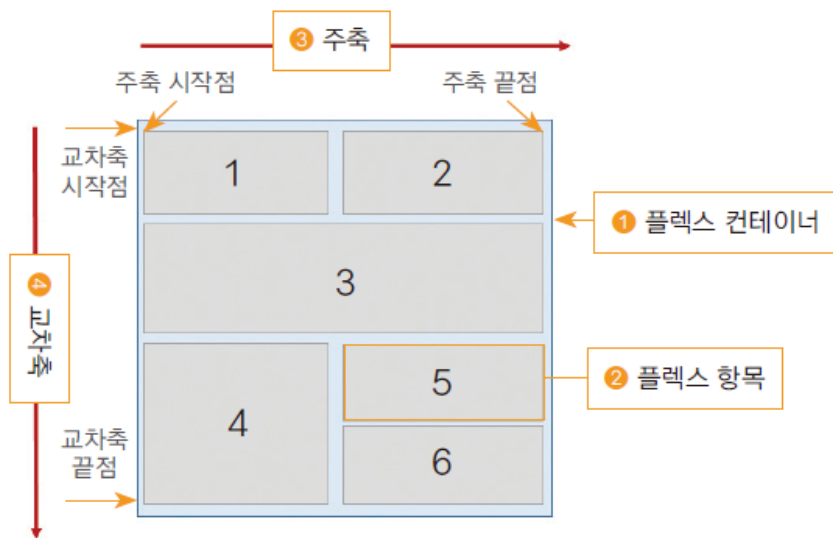
## [실습] PC용 미디어 쿼리 작성하기

p559 ~ p561

## 14.6. 플렉서블 박스 레이아웃

### (1) 플렉서블 박스 레이아웃과 새로운 용어

- 플렉서블 박스 레이아웃(flexible box layout)이란 그리드 레이아웃을 기본으로 해 플렉스 박스를 원하는 위치에 배치하는 것이다.
- 플렉스 컨테이너(flex container)
- 플렉스 항목(flex item)
- 주축(main axis)
- 교차축(cross axis)



## (2) 플렉서블 박스 레이아웃 기본 속성

### display 속성 - 플렉스 컨테이너 지정하기

- 기본형: `display: flex | inline-flex`
  - `flex`: 플렉스 박스를 박스 레벨 요소로 정의한다.
  - `inline-flex`: 플렉스 박스를 인라인 레벨 요소로 정의한다.
- 배치하려는 웹 요소들이 있다면 그 요소들을 감싸는 부모 요소를 만들고 그 부모 요소를 플렉스 컨테이너로 만든다.

예) `<style>`

```
#container {
  display: flex;
}
```

`</style>`

```
<div id="container">
```

...

```
</div>
```

### display 속성과 브라우저 접두사

- 브라우저마다 플렉스 박스를 지원하는 방법이 달라 브라우저 접두사를 붙여야 한다.

예) `.wrapper {`

```
display: -webkit-box;
```

```
display: -moz-box;
```

```
display: -ms-flexbox;
```

```
display: -webkit-flex;
```

```
display: flex;
```

```
}
```

### flex-direction 속성 - 플렉스 방향 지정하기

- 기본형: `flex-direction: row | row-inverse | column | column-inverse`

속성 값	설명
row	주축을 가로로 교차축을 세로로 지정합니다. 플렉스 항목은 주축 시작점에서 끝점으로(왼쪽에서 오른쪽으로) 배치됩니다.*
row-inverse	주축을 가로로 교차축을 세로로 지정합니다. 플렉스 항목은 주축 끝점에서 시작점으로(오른쪽에서 왼쪽으로) 배치됩니다.
column	주축을 세로로 교차축을 가로로 지정합니다. 플렉스 항목은 주축 시작점에서 끝점으로(위쪽에서 아래쪽으로) 배치됩니다.
column-inverse	주축을 세로로 교차축을 가로로 지정합니다. 플렉스 항목은 주축 끝점에서 시작점으로(아래쪽에서 위쪽으로) 배치됩니다.

[14/flex1.html] flex-direction 속성은 지정하지 않았을 경우에 기본 값인 row로 인식한다.

```
<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>플렉스 박스</title>
<style>
#container {
    width: 500px;
    height: 300px;
    margin: 0 auto;
    display: flex;
    border: 2px solid black;
}

#container div {
    width: 200px;
    border: 2px solid black;
    background: #ccc;
}

h2 {
    font-size: 20px;
    font-weight: bold;
    padding: 20px;
}
</style>
</head>
<body>
    <div id="container">
        <div id="box1">
            <h2>1</h2>
        </div>
        <div id="box2">
            <h2>2</h2>
        </div>
        <div id="box3">
            <h2>3</h2>
        </div>
    </div>
</body>
</html>
```

[14/flex2.html] 플렉스 방향을 세로로 지정한 것이다.

```
<!DOCTYPE html>

<html lang="ko">
```

```

<head>
<meta charset="utf-8">
<title>플렉스 박스</title>
<style>
#container {
    width: 500px;
    height: 300px;
    margin: 0 auto;
    display: flex;
    flex-direction: column;
    border: 2px solid black;
}

#container div {
    width: 200px;
    border: 1px solid black;
    background: #ccc;
}

h2 {
    font-size: 20px;
    font-weight: bold;
    padding: 20px;
}
</style>
</head>
<body>
    <div id="container">
        <div id="box1">
            <h2>1</h2>
        </div>
        <div id="box2">
            <h2>2</h2>
        </div>
        <div id="box3">
            <h2>3</h2>
        </div>
    </div>
</body>
</html>

```

## flex-wrap 속성 - 플렉스 항목을 한 줄 또는 여러 줄로 배치하기

- 기본형: flex-wrap: no-wrap | wrap | wrap-reverse

속성 값	설명
no-wrap	플렉스 항목들을 한 줄에 표시합니다. 기본 값입니다.★
wrap	플렉스 항목을 여러 줄에 표시합니다.
wrap-reverse	플렉스 항목을 여러 줄에 표시하되 기존 방향과 반대로 배치합니다.

[14/flex3.html] 플렉스 항목을 여러 줄에 표시한다.

```

<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>플렉스 박스</title>
<style>
#container {
    width: 500px;

```

```

        height: 300px;
        margin: 0 auto;
        display: flex;
        -ms-flex-wrap: wrap;
        -webkit-flex-wrap: wrap;
        flex-wrap: wrap;
        border: 2px solid black;
    }

    #container div {
        width: 200px;
        border: 2px solid black;
        background: #ccc;
    }

    h2 {
        font-size: 20px;
        font-weight: bold;
        padding: 20px;
    }
</style>
</head>
<body>
    <div id="container">
        <div id="box1">
            <h2>1</h2>
        </div>
        <div id="box2">
            <h2>2</h2>
        </div>
        <div id="box3">
            <h2>3</h2>
        </div>
    </div>
</body>
</html>

```

[14/flex4.html] 교차축의 끝점에서 시작점으로(아래 -> 위) 배치한다.

```

<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>플렉스 박스</title>
<style>
    #container {
        width: 500px;
        height: 300px;
        margin: 0 auto;
        display: flex;
        -ms-flex-wrap: wrap-reverse;
        -webkit-flex-wrap: wrap-reverse;
        flex-wrap: wrap-reverse;
        border: 2px solid black;
    }

    #container div {
        width: 200px;
        border: 2px solid black;
        background: #ccc;
    }

    h2 {

```

```

        font-size: 20px;
        font-weight: bold;
        padding: 20px;
    }
</style>
</head>
<body>
    <div id="container">
        <div id="box1">
            <h2>1</h2>
        </div>
        <div id="box2">
            <h2>2</h2>
        </div>
        <div id="box3">
            <h2>3</h2>
        </div>
    </div>
</body>
</html>

```

### flex-flow 속성 - 플렉스 방향과 여러 줄의 배치를 한꺼번에 지정하기

- 기본형: flex-flow: <플렉스 방향> <플렉스 줄 배치>  
예) flex-flow: column wrap; /\* 위에서 아래로, 여러 줄에 표시 \*/

### order 속성 - 플렉스 항목의 배치 순서 바꾸기

- 기본형: order: 0 | 숫자

[14/flex-order.html] 1번 박스를 두 번째, 2번 박스를 세 번째, 3번 박스를 첫 번째로 배치한다.

```

<!DOCTYPE html>

<html lang="ko">
<head>
<meta charset="utf-8">
<title>플렉스 박스</title>
<style>
#container {
    width: 500px;
    height: 300px;
    margin: 0 auto;
    display: flex;
    border: 2px solid black;
}

#container div {
    width: 200px;
    border: 2px solid black;
    background: #ccc;
}

#box1 {
    order: 2;
}

#box2 {
    order: 3;
}

#box3 {
    order: 1;
}

```

```

h2 {
    font-size: 20px;
    font-weight: bold;
    padding: 20px;
}
</style>
</head>
<body>
    <div id="container">
        <div id="box1">
            <h2>box1</h2>
        </div>
        <div id="box2">
            <h2>box2</h2>
        </div>
        <div id="box3">
            <h2>box3</h2>
        </div>
    </div>
</body>
</html>

```

### flex 속성 - 플렉스 항목 크기 조절하기

- 기본형: flex: [<flex-grow> <flex-shrink> <flex-basis>] | auto | initial

속성 값	설명
	플렉스 항목의 너비를 얼마나 늘일지 숫자로 지정합니다.
<flex-grow>	플렉스 항목의 너비를 얼마나 줄일지 숫자로 지정합니다.
<flex-shrink> <flex-basis>	플렉스 항목의 기본 크기를 지정합니다. width 속성처럼 너비 값을 지정할 수도 있고 0이나 auto를 지정할 수도 있습니다. 0일 경우, flex-grow와 flex-shrink의 인수 값을 함께 사용하고 auto일 경우, 플렉스 항목의 너비 값을 사용합니다.
initial	항목의 width/height 값에 의해 크기가 결정되는데 플렉스 컨테이너의 공간이 부족할 경우, 최소 크기까지 줄입니다.*
auto	항목의 width/height 값에 의해 크기가 결정되지만 플렉스 컨테이너의 공간에 따라 늘어거나 줄입니다.

```

<style>
    #box1 {
        flex:1 1 0; /* 늘어거나 줄이지 않음 */
    }
    #box2 {
        flex:2 2 0; /* 2배 늘어거나 2배 줄임 */
    }
</style>

```

1	2	3
---	---	---

### (3) 플렉스 항목 배치를 위한 속성들

#### justify-content 속성 - 주축 기준의 배치 방법 지정하기

- 기본형: justify-content: flex-start | flex-end | center | space-between | space-around



속성 값	설명	미리 보기
flex-start	주축의 시작점을 기준으로 배치합니다.	
flex-end	주축의 끝점을 기준으로 배치합니다.	
center	주축의 중앙을 기준으로 배치합니다.	
space-between	첫 번째 플렉스 항목과 마지막 플렉스 항목은 시작점과 끝점에 배치한 후 중앙 항목들은 같은 간격으로 배치합니다.	
space-around	모든 플렉스 항목들을 같은 간격으로 배치합니다.	

### align-items 속성, align-self 속성 - 교차축 기준의 배치 방법 지정하기

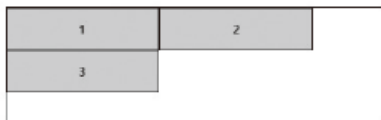
- 기본형: align-items: stretch | flex-start | flex-end | center | baseline

속성 값	설명	미리 보기
stretch	플렉스 항목을 확장해 교차축을 꽉 채웁니다. 기본 값입니다.★	
flex-start	교차축의 시작점을 기준으로 배치합니다.	
flex-end	교차축의 끝점을 기준으로 배치합니다.	
center	교차축의 중앙을 기준으로 배치합니다.	
baseline	시작점과 글자 기준선이 가장 먼 플렉스 항목(미리보기에서는 2번 항목의 글자 크기가 가장 크기 때문에 2번의 글자 기준선이 가장 멀리 떨어져 있음)을 시작점에 배치합니다. 그리고 그 글자의 기준선과 다른 항목의 기준선을 맞추어 배치합니다.	

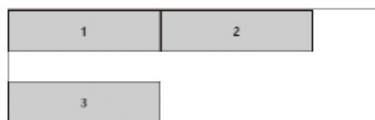
### align-content 속성 - 여러 줄일 때의 배치 방법 지정하기

- 기본형: align-content: flex-start | flex-end | center | space-between | space-around

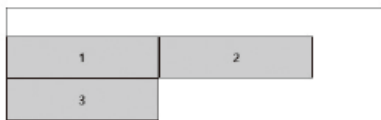
#### align-content : flex-start (기본 값)



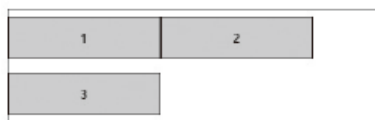
#### align-content : space-between



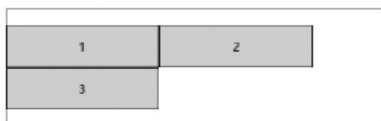
#### align-content : flex-end



#### align-content : space-around



#### align-content : center



## [실습] 플렉스 박스를 이용해 사이트 구성하기

p573 ~ p575



## 15. 반응형 웹 사이트 만들기

### 15.1. 레이아웃 구조 만들기

#### (1) 레이아웃 구상

- 구상 방법: 모바일 퍼스트(mobile first) 즉, 주로 모바일 화면부터 시작해 태블릿과 PC 화면 순서대로 구상한다.

#### (2) 미디어 쿼리 중단점 만들기

미디어 쿼리 구문의 예

```
<style>
/* 모든 상황에서 똑같이 사용할 스타일 */
...

/* 모바일 - 767px 이하 */
...

/* 태블릿 - 768px 이상 */
@media screen and (min-width:768px) {
  ...
}

/* PC - 992px 이상 */
@media screen and (min-width:992px) {
  ...
}
</style>
```

### 15.2. 각 기기용 레이아웃 만들기

#### [실습] 문서 기본 구조 만들기

p580 ~ p584

#### [실습] 모바일용 레이아웃 구조 만들기

p584 ~ p591

#### [실습] 웹 폰트 사용 및 메뉴 만들기

p592 ~ p595

#### [실습] 태블릿용 및 PC용 레이아웃 만들기

p595 ~ 600

