Title: 러닝 스쿨! 한권으로 끝내는 HTML + CSS 웹 디자인 입문

Contents

Chapter 1: 기억해 두어야 하는 웹 사이트 기본

Chapter 2: 웹 기본 구조 만들기 - HTML 기본

Chapter 3: 웹 디자인 만들기 - CSS 기본

Chapter 4: 풀 스크린 웹 사이트 제작하기

Chapter 5: 2 - 칼럼 웹 사이트 제작하기

Chapter 6: 타일 형태의 웹 사이트 제작하기

Chapter 7: 외부 미디어 사용하기

Reference

The Best Design <https://www.thebestdesigns.com> 웹 사이트 트렌드 파악에 이용

Pattern Tap <https://patterntap.com/patterntap> 웹 사이트 구성 요소 파악에 이용

Google Fonts <https://fonts.google.com> 웹 폰트 적용에 사용

Compressor <https://compressor.io/compress> 이미지 파일 용량 조절 (품질유지)

Adobe Color CC <https://color.adobe.com/ko/create> 키워드 인상에 맞는 배색 검색

Paletton <https://paletton.com> 웹 페이지 배색 미리보기

Canva [canva.com/colors/color-palette-generator](https://canva.com/colors/color-palette-generator) 이미지와 어울리는 색상 추천

Ress.css [www.github.com/filipelinhares/ress](http://www.github.com/filipelinhares/ress) 디폴트 CSS

<https://unpkg.com/ress/dist/ress.min.css> 웹 공개 디폴트 CSS

Chapter 1. 기억해 두어야 하는 웹 사이트 기본

* 1. 좋은 웹 디자인 이란

- 디자인의 목적은 상대방에게 무언가를 전달하는 것

“디자인을 통하여 무엇을 전달하고 싶은가”

“따라서 디자인은 ‘전달을 위한 수단’이며, ‘아름답게 장식하는 것’이 아니다.”

- 쉬운 사용

- 보다 좋은 생활을 제공하는 것

“디자인은 문제를 해결하는 방법이다.”

* 1. 다양한 종류의 사이트

- **기업 사이트**: 기업의 정보를 담고 있는 사이트. 회사 개요, 제품 소개, 채용 정보 등의 정보 전달

- **프로모션 사이트**: 특정 제품, 서비스, 이벤트 정보 전달 목적. 정보의 범위, 타겟층이 좁으며 주로 기간 한정

- **포트폴리오 사이트**: 자신의 작품, 실적을 소개하는 사이트.

**- 쇼핑 사이트**: 인터넷에서 상품을 판매하는 쇼핑 사이트. EC사이트, 온라인 스토어 사이트. 상품 목록 페이지, 상품 상세 페이지, 장바구니 페이지, 결제 페이지 등 다양한 페이지로 구성. 구매 유도 고려.

- **미디어 사이트**: 뉴스, 정보 공개 목적 사이트. 광고, 생각 전달.

- **SNS**: 사용자와 실시간 직접 소통. 정보 확산 유리. 짧은 글, 이미지, 동영상을 메인으로 전달

=> 하나의 웹 사이트가 여러 종류를 조합한 경우도 있다.

=> 목적에 맞는 웹사이트 구성이 중요하다.

* 1. 사용성이란?

**- 사용하기 쉬움(Usability)을 생각하기:** 요구하는 기능을 스트레스없이 사용할 수 있는 경우

**- 보다 쉽게 디자인하기:** 색 사용에 주의하기, 중요성을 명확하게 보이기, 레이아웃 통일하기

**- 읽기 쉬운 글 만들기:** 결론을 앞에 두기, 전문 용어 사용하지 않기, 간략하게 정리하기(큰 제목, 작은 제목, 줄 바꿈, 목록을 활용)

**- 사용하기 쉬운 조작성:** 예상할 수 있게 만들기, 빠른 동작, 한 번에 보고 알 수 있게 하기

* 1. 웹사이트의 구조

- **인터넷이란?** 세계의 컴퓨터들이 다양한 정보를 교환할 수 있는 시스템

- **웹이란?** 인터넷을 사용하여 정보 전송과 확인을 위한 하나의 기능

- **웹 페이지 구조**

1. 웹 서버: 컴퓨터의 일종. 정보 공개, 저장하는 기능. 직접적인 조작은 연결되지 않음. 요청에 응답.
2. 웹 클라이언트: 웹 서버에서 정보를 받는 쪽. 일반 사용자가 사용하는 컴퓨터. 정보를 서버에 요청.

- **URL(Uniform Resource Locator)이란?** 웹 사이트의 주소.

* 1. 장치의 종류
  2. 웹 브라우저의 종류

**- 웹 브라우저?**

>> 웹 서버에서 전송된 데이터를 분석하고 사람이 읽기 적합한 형태로 변환해주는 소프트웨어.

>> 웹 서버에서 전송된 웹 페이지 데이터는 알파벳과 기호로 구성된 ‘코드’로 작성’됨.

>> 웹 브라우저는 적절한 문자 크기, 이미지 배치, 색, 레이아웃 등이 사용 친화적으로 조정되어야 함.

**- 종류?**

>> 구글 크롬(Google Chrome)

>> 사파리(Safari)

>> 마이크로소프트 엣지/인터넷 익스플로러(Microsoft Edge/Internet Explorer)

>> 파이어폭스(Firefox)

**- 역할?** ‘웹 페이지 데이터를 보기 쉽게 출력’

* 1. 제작 흐름

**- 개요**

>> 기획 -> 사이트 맵 제작 -> 와이어 프레임 제작 -> 디자인 -> 코딩 -> 웹 공개

1. **기획**: 웹 페이지에는 다양한 종류가 있으며, 목적에 따라 구성이 달라진다.

**- 목적 설정**

=> 기본 목표(main goal)와 부가적인 목표(sub goal)을 정한다.

=> 목표 달성을 측정하는 기준을 설정하며 피드백을 한다.

- **타겟 사용자 결정** (방향성 설정)

=> 항목: 성별, 연령대, 직업, 취미, 고민, 수입, 국가 또는 지역

=> 범위가 너무 넓으면 적절하지 않으며, 특정 범위를 지정하는 것이 좋다.

=> 타겟 사용자를 보다 구체화한 ‘페르소나’라는 인물상을 설정하는 것도 좋다.

1. **사이트 맵 제작**: 필요한 페이지가 무엇인지, 어떻게 연결할지, 중요도에 따른 구성을 그림으로 정리

- **페이지의 우선 순위**

=> 우선 순위가 높은 페이지는 눈에 잘 띄게 네비게이션 메뉴 내부에 배치해야 한다

=> ‘사람들에게 보여줘야 하는 페이지’와 ‘부가적인 정보를 위한 준비 페이지’

- **페이지를 그룹으로 묶기**

냉장고이(가) 표시된 사진

자동 생성된 설명

=> 필요하다고 생각하는 페이지를 결정, 이후 관련 있는 페이지를 그룹화하여 계층으로 만들기

=> 필요한 페이지를 압축해서 깊이가 2계층 이내로 구성하는 것이 좋다. (최대 3 계층)

1. **와이어 프레임 제작**

>> 페이지의 레이아웃 디자인 골격 ‘와이어 프레임’을 구성

>> 색과 장식 등을 넣지 않고 텍스트, 선, 박스만으로 구성

>> 모바일 버전에서는 레이아웃이 변하는 경우가 많으니 고려해야 한다.

>> 고객과의 원활한 소통을 가능케하며, 디자인 제작 단계에서도 작업이 쉬워진다.

**- 콘텐츠 우선순위**

=> 사이트 맵과 마찬가지로, 페이지 내부에 어떤 요소가 필요한지 작성.

=> 필요 요소 중에서 우선순위를 붙이고, 이를 기반으로 레이아웃을 생각한다.

=> 순위가 높을수록 페이지 위쪽에, 출력 면적을 크게 설정한다.

=> 퍼스트 뷰(First View)에 사이트가 전달하고자 하는 바를 표현하여 구성한다.

**- 시선의 움직임을 생각하기**

=> 일반적으로 사용자의 시간은 위에서 아래로, 왼쪽에서 오른쪽으로 이동한다.

=> 우선순위가 높은 콘텐츠 일수록 페이지의 위쪽, 왼쪽에 배치한다.

**- 와이어프레임 작성 도구**

=> 온라인 도구: Casco, Moqups, Mockingbird, Wireframe.cc, InVision

=> 어플리케이션: Adobe XD, Adobe COMP CC, Adobe Illustrator, Sketch, Justinmind

1. **디자인**

>> 디자인 시안(Design Comprehensive Layout), 실제 웹 페이지에 출력되는 형태와 똑같이 그래픽 도구를 사용하여 만든다.

>> 사용자의 시점에서 이미지 배치, 배색, 글꼴, 여백 등을 고려하면서 만든다.

1. **코딩**

- HTML을 사용하여 웹 사이트의 콘텐츠 작성하기

=> 실제로 웹 페이지에 출력하고 싶은 글과 이미지 등을 HTML을 이용하여 작성한다. ‘.html’

- CSS를 사용하여 웹 사이트 장식하기

=> 색, 문자 크기, 배치와 같은 장식을 구현한다. ‘.css’

- 이미지 파일 준비하기

=> 사용할 이미지를 ‘images’ 등의 폴더에 저장한다. 웹에서 사용하는 이미지 종류는 다음과 같다. 텍스트이(가) 표시된 사진

자동 생성된 설명

1. **웹 공개**

>> 자신의 컴퓨터에 저장한 파일은 자신만 볼 수 있다.

>> 다른 사람들에게 공개하려면, 웹 서버에 파일을 전송하는 ‘업로드’를 해야한다.

- **서버 준비**: 웹 서버는 일반적으로 서버 회사에서 서버를 렌탈하는 호스팅 서비스를 이용한다.

- **도메인 준비:** 웹 사이트의 장소를 나타내는 주소

- **웹 서버 파일 업로드**

=> 일반적인 렌탈 서버는 웹에서 파일을 드래그하는 형태로 업로드 할 수 있다.

=> 하지만, 파일의 수가 많은 경우 전송 소프트웨어(FTP 소프트웨어)를 사용하는 것이 좋다.

=> 이용 시, FTP 서버 주소, 사용자 이름, 비밀번호를 입력해야 한다. (서버 계약 시 제공 정보)

**- URL을 통한 웹사이트 출력(접근)**

* 1. 제작 시작 전

- 텍스트 에디터 설치

- 브라우저 설치

- 그래픽 도구 확인

Chapter 2. 웹 기본 구조 만들기 - HTML 기본

* 1. HTML이란?

>> HTML(Hyper Text Markup Language), 웹 페이지 작성 언어.

>> <태그> 문자열을 사용하여 작성하며, 해당 페이지의 기본 구조와 기능을 지시한다.

* 1. HTML 파일 만들기

- 텍스트 에디터 실행

- 코드 작성

- 파일 저장

- 웹 브라우저 열어 보기

**- HTML 파일 이름 규칙**

=> 파일 이름에 확장자 붙이기

=> 한글 사용하지 않기

=> 사용할 수 없는 기호에 주의하기

=> 띄어쓰기 사용하지 않기. 대신, 하이픈(-)기호 사용하기

=> 기본적으로 소문자로 통일해서 파일 이름 작성하기

=> 홈페이지는 “index.html”로 지정하기. “https://example.com/index.html과 https://example.com은 동일. 같은 페이지가 출력된다.

* 1. HTML 파일 골격

- <!doctype html>: 해당 페이지가 어떤 버전의 html에 맞게 작성되었는지 선언.

- <html>~</html>: HTML 문서임을 나타냄. “lang”은 웹 페이지의 언어를 설정.

- <head>~</head>: 페이지의 제목과 설명, 사용할 외부 파일 링크 등의 페이지 정보 작성.

- <meta charset = “UTF-8”>: 문자 코드 지정. ‘UTF-8’ 포맷으로 문자를 인코딩함을 지정.

- <title>~</title>: 페이지의 제목 설정. 브라우저 탭에 출력. 북마크 페이지 제목.

- <meta name = “description” content = ““>: 페이지의 설명 입력. 검색 엔진 상, 페이지 검색 시 출력.

- <body>~</body>: HTML 문서의 본체. 콘텐츠 입력 시 실제 브라우저에 출력.

* 1. HTML 기본 작성 방법

**- HTML 기본 문법과 태그**

=> HTML은 기본적으로 문자열을 태그로 감싸며 작성한다. 태그는 다양한 종류가 있으며 해당 부분의 역할을 결정한다.

=> 시작 태그, 종료 태그의 한 쌍을 요소라 부르며, 경우에 따라 종료 태그가 없는 경우도 있다.

=> 기본적으로 태그는 대/소문자를 구분하지 않지만, 버전에 따라 소문자로 작성. 따라서 소문자 통일.

=> 태그에 따라, 시작 태그 내부에 관련 추가 정보를 작성한다. 이 정보를 속성이라고 부르며 태그 이름 뒤에 띄어쓰기 후 작성한다.

=> 추가로 이러한 정보의 내용을 값이라고 부르며, 큰따옴표(“”)로 감싸서 작성한다.

* 1. 제목 붙이기

=> 제목은 <h1> ~ <h6> 태그를 사용해서 만든다. “heading”의 약자이며 숫자가 커질수록 작아진다.

=> 태그의 크기는 내림차순으로 순서를 지켜서 사용되어야 하며, <h1>은 하나의 웹페이지에 하나만 사용하는 것이 좋다. 즉, 해당 페이지가 무엇을 나타내는가를 표현하는 문장을 나타내는데 사용한다.

=> 일반적으로 대부분의 웹페이지는 <h4>까지 사용되며. 필요시 이하 헤더를 사용한다.

* 1. 문장 출력하기

=> 단락을 출력하는 <p>태그, “paragraph”, 단락과 단락 사이에는 작은 여백이 발생한다.

=> 문단을 구성하는 문장의 줄 수가 너무 많아지면 가독성이 떨어진다. 3 ~ 5줄 정도가 적당하며, 문자 수도 한 문장에 30 ~ 45자로 조절한다.

* 1. 이미지 삽입하기

**- 이미지를 출력하는 <img> 태그**

=> 이미지에는 <img> 태그를 사용한다. 종료 태그가 없으므로 단독으로 사용하는 것이 특징이다.

=> 어떤 이미지를 출력할 것인지 src속성을 설정해야 하며, 실패 시 대체할 alt속성을 설정해야 한다.

**- 파일 경로 지정하기**

=> 파일경로란 HTML, CSS, 이미지 등의 파일을 불러오기 위해서 해당 파일이 어디 있는지 지정하는 것.

=> 상대 경로: 호출하는 페이지에서 본 대상 파일의 위치 지정. <폴더 이름>/<파일 이름>.

=> 상위 단계는 ../을 이용하여 이동.

=> 절대 경로: 호출하고 싶은 파일이 다른 웹 사이트에 공개 된 경우, 해당 (https)도메인 경로를 지정.

* 1. 링크 만들기

>> 링크는 <a>태그를 이용하며, href 속성을 통해 링크 대상이나 경로를 지정한다.

>> 링크를 넣고 싶은 부분을 <a href = “”>~</a> 태그 사이에 끼워 넣는다.

>> 이미지를 링크로 만들기 위해 <a>태그 요소 사이에 <img>태그를 조합하면 작성할 수 있다.

>> 메일 전용 링크는 mailto:를 이용한다. e.g) <a href = “mailto:info@example.com”>문의</a>

>> 링크 대상 페이지를 다른 탭에 출력하기 위해서 target 속성을 \_blank로 지정한다. 단, 사용자가 해당 페이지에서 열지에 대한 선택권을 막을 수 있다. 그러므로 이런 점을 고려해야 한다.

e.g) <a href = https://google.co.kr target = “\_blank”>Google</a>

* 1. 리스트 출력하기

- 기본적인 리스트를 작성하는 “<ul>태그 + <li>태그”

=> 기본적인 리스트를 출력 시 <ul>태그를 사용한다. “unsorted list”, 순서가 없는 리스트를 의미한다.

=> <ul>태그 내부에 <li>태그를 사용하여 항목을 추가한다. “list item”, 리스트의 항목을 의미한다.

=> <ol>태그는 순서가 있는 리스트. “ordered list”, 순서가 있는 정렬 리스트를 의미한다.

* 1. 표 만들기

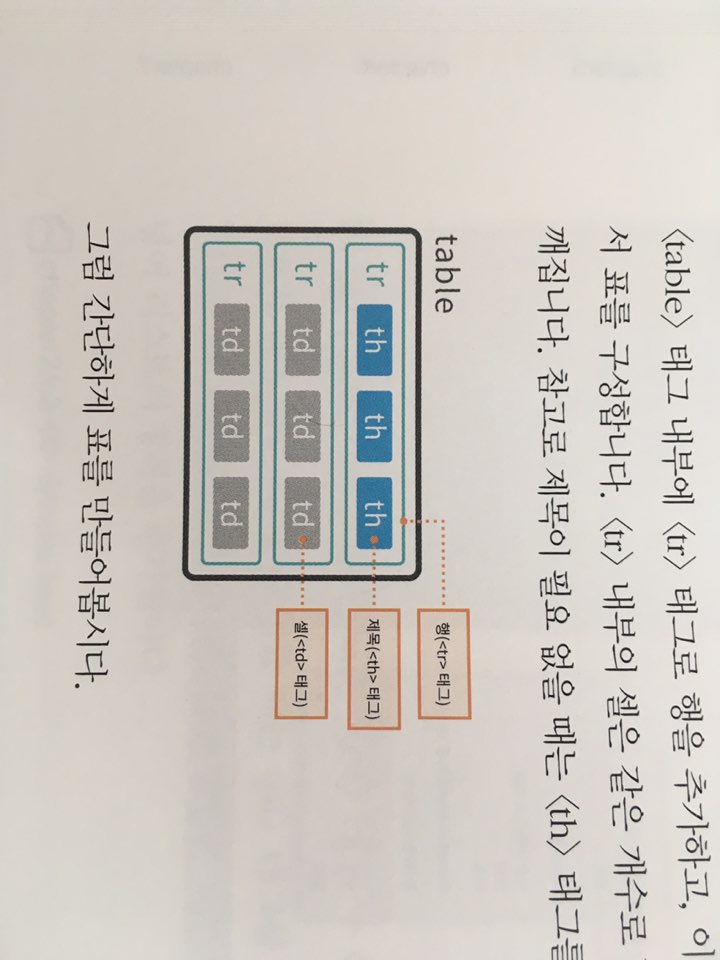
**- 표 기본**

=> <table> 표를 나타내는 태그, 표 전체를 감싼다.

=> <tr> “Table Row”의 약자로 표의 한 행을 나타낸다.

=> <th> “Table Header”의 약자로 표의 제목 셀을 나타낸다.

=> <td> “Table Data”의 약자로 표의 데이터 셀을 나타낸다.



**- 셀 병합**

=> 수평 방향으로 병합하기: <th> 또는 <td>태그에 colspan 속성을 추가. <태그 colspan=”병합개수”>.

=> 수직 방향으로 병합하기: <th> 또는 <td>태그에 rowspan 속성 <태그 rowspan=”병합개수”>.

* 1. 입력 양식 만들기

- **입력 양식 란을 만드는 <form> 태그:** <form> 태그는 입력 양식을 만들기 위한 태그이며, 입력 양식에서 사용하는 모든 부품은 <form> 태그 내부에 넣어준다. 주요 속성은 다음과 같다.

=> action 데이터의 전송 대상 페이지를 지정한다.

=> method 데이터 전송 방법을 지정한다. 주로 get 또는 post를 사용한다.

=> name 입력 양식의 이름을 지정한다.

“<form> 태그는 입력, 전송 버튼을 만들 때 사용하는 요소이지만, 그 입력 양식에 대한 처리가 HTML과 CSS에서 이뤄지는 것은 아니다. 실질 데이터 처리는 PHP등의 다른 프로그래밍 언어에서 이뤄진다.”

- **입력 양식에 사용하는 부품:** 대부분 <input> 태그를 사용하며, type 속성으로 용도를 지정해준다.

=> 한 줄 텍스트 입력란 <input type = “text”>

=> 한 줄 텍스트 입력란, 미리 출력 <input type = “text” placeholder = “입력”>

(한 줄 텍스트 입력 type의 종류) (용도)

text 기본적인 텍스트 (초기값)

search 검색 전용 텍스트

email 메일 주소

tel 전화번호

url 웹 사이트의 url

=> 여러 개중 하나만 선택, 라디오 버튼 <input type = “radio”>

=> 여러 개중 여러 개 선택, 체크 박스 <input type = “checkbox”>

(체크 박스/라디오 버튼 주요 속성) (용도)

name 버튼/박스의 이름

value 전송될 선택지의 값

checked 처음부터 선택돼 있게 하고 싶을 때 지정

=> 전송 버튼 <input type = “submit”>

- 이미지버튼 <input type = “image”>

- 선택 박스 <select> 태그 + <option> 태그

- 여러 줄 텍스트 입력란 <textarea> 태그

* 1. 사용하기 쉬운 입력 양식 만들기

**- 입력 양식의 레이블을 만드는 <label> 태그**

=> <label> 태그를 사용하면 입력 양식 부분과 레이블이 연결돼 레이블이 선택되면 입력 양식 부분에 초점이 맞춰진다.

=> 사용자에 따라서 작은 라디오 버튼, 체크 박스를 클릭하는 것이 힘들 수 있으므로, 보다 사용하기 쉬운 입력 양식을 위해 이용하는 것이 좋다.

=> 레이블 텍스트 부분을 <label> 태그로 감싸고, for 속성을 지정하여 연결하고 싶은 양식에 id 속성을 준다.

=> 식별 이름을 붙일 때는 ‘name-1’과 같이 이름을 통일하여 구별하는 것이 좋다.

* 1. 블록 요소로 그룹 나누기

1. **그룹 구분하기**

=> 웹 사이트는 여러 구성 요소들이 뭉쳐서 만들어지며, 각 구성 요소는 태그로 그룹화된다.

=> HTML만으로는 출력에 변화가 없지만, CSS를 사용하면 레이아웃 변경이 가능하다.

=> 그룹을 만들기 위한 태그에는 각각 의미가 있으며, 용도에 맞게 구분되어야 한다.

+) <h1>, <p> 태그처럼, 줄 바꿈이 들어가는 블록을 만드는 태그를 ‘블록 요소’라 부른다.

+) <a>, <img> 태그처럼, 텍스트의 일부로서 다뤄지는 태그를 ‘인라인 요소’라 부른다.

1. **페이지 상부의 헤더를 만드는 <header> 태그**

=> 페이지 상부에 있는 요소를 감싼다. 일반적으로 로고, 페이지 제목, 네비게이션 메뉴를 포함한다.

1. **네이게이션 메뉴를 만드는 <nav> 태그**

=> 기본적으로 메인 메뉴가 아닌 부분에는 사용하지 않으며, <header> 태그에 주로 포함된다.

1. **기사 부분을 만드는 <article> 태그**

=> 페이지 내부의 기사 부분, 독립적 구성이 되는 내용.

1. **테마를 가진 그룹을 만드는 <section> 태그**

=> 의미를 가진 그룹을 감싸는 태그, <article>과의 차이는, 독립적 구성이기보다는 하나의 테마

1. **페이지의 메인 콘텐츠 부분을 만드는 <main> 태그**

=> 해당 페이지의 핵심이 되는 콘텐츠 전체를 감싼다. 내부에 다양한 그룹이 존재한다.

1. **메인 콘텐츠가 아닌 추가 정보 <aside> 태그**

=> 메인 콘텐츠와는 관계성이 낮은 정보에 사용한다. 사이드바에도 사용한다.

1. **페이지 하부의 푸터 부분을 만드는 <footer> 태그**

=> 일반적으로 저작권, SNS 링크 등을 포함하게 된다.

1. **의미를 가지지 않는 블록 요소를 만드는 <div> 태그**

=> 미정, 혹은 어떠한 용도에 속하지 않거나, 단순하게 디자인을 위해 그룹을 묶는 경우 사용

* 1. 자주 사용하는 HTML 태그 목록

- 기본 구조, head 내부에 사용하는 태그

(태그) (용도)

html html 문서의 시작

head 검색 엔진을 위한 설명, CSS 파일 링크, 페이지 제목 포함

meta 언어와 설명 등 페이지 정보 작성

title 페이지 제목, 브라우저 탭, 북마크에 제목으로 출력된다.

link 참조할 외부 파일. 주로 CSS 파일 연결

body html 문서의 콘텐츠, 출력되는 부분.

- 콘텐츠로 사용하는 태그

(태그) (용도)

h1 ~ h6 제목 출력, 숫자 순서대로 작성한다.

p 문서의 문단

img 이미지 출력, src 속성으로 이미지 지정

a 링크 생성, href 속성으로 링크 대상 지정

ul 순서 없는 목록

ol 순서 있는 목록

li 목록의 항목을 나타냄

br 줄바꿈

strong 두꺼운 글자 출력, 중요 항목 표시

blockquote 인용문을 나타낸다.

small 저작권 또는 법적 설명을 나타낸다.

span 의미 없는 인라인 요소. CSS로 일부분만 장식할 때 활용

audio 음악 데이터 출력을 위해 사용

video 동영상 데이터 출력을 위해 사용

script 스크립트 데이터를 넣거나 참조할 때 사용. 주로 JS코드 사용.

- 표

(태그) (용도)

table 표를 나타내는 태그, 표 전체를 감싼다.

tr 표의 한 줄을 감싼다.

th 표의 제목 셀을 만든다.

td 표의 데이터 셀을 만든다.

- 콘텐츠에 사용하는 태그

(태그) (용도)

form 입력 양식을 만든다.

<input type = “text”> 한 줄 텍스트 입력란

<input type = “radio”> 라디오 버튼, 하나만 선택 가능

<input type = “checkbox”> 체크 박스, 여러 개 선택 가능

<input type = “submit”> 전송 버튼

select 선택 박스

option 선택 박스의 선택 항목을 만든다.

textarea 여러 줄 텍스트 입력란을 만든다.

label 입력 양식의 레이블. 입력 양식과 연결해서 사용.

- 그룹 구분 전용 블록 요소

(태그) (용도)

Header 페이지 상부 요소. 로고, 페이지 제목, 네비게이션 메뉴를 감쌈.

Nav 메인 네비게이션 메뉴

Article 페이지 기사 부분. 독립적 페이지

Section 하나의 테마를 가진 그룹

Main 페이지의 메인 콘텐츠 부분

Aside 본문이 아닌 추가 정보. 메인과 연관성이 낮은 정보에 사용

Footer 페이지 하부 요소. 저작권 SNS 링크 등을 포함

Div 의미를 가지지 않는 블록 요소

Chapter 3: 웹 디자인 만들기 - CSS 기본

3-1. CSS란?

- CSS(Cascading Style sheets): ‘연속적 스타일 시트’, 확장자 “.css”

3-2. CSS를 적용하는 방법

**- CSS 파일을 읽어 들여 적용하기**

=> 가장 일반적인 방법. 하나의 CSS 파일을 여러 HTML 파일에서 읽어 들여서 한 번에 관리.

=> HTML 파일의 <head> 내부에 <link> 태그를 사용하여 CSS 파일을 지정한다.

=> 속성으로 rel은 stylesheet를 지정하고, href로 CSS 파일을 연결한다.

‘<link rel = “stylesheet” href = “주소/.css파일”>’

**- HTML 파일의 <head> 내부에 <style> 태그를 넣어 직접 스타일을 지정하기**

=> 해당 HTML 파일에만 CSS를 적용하기 위해 사용. 특정 페이지 디자인 변경.

**- HTML 태그 내부에 style 속성 지정하기**

=> 하나 하나의 태그에 직접 적용. CSS를 적용하는 우선순위가 높음.

3-3. CSS 파일 만들기

- CSS 코드 작성하기

=> CSS 파일 첫 줄은 ‘@charset “utf-8”’을 작성하여 코드의 문자 깨짐을 방지한다.

3-4. CSS 기본 작성 방법 익히기

**- CSS 기본 문법**

=> CSS는 선택자, 속성, 값이라고 부르는 3가지 요소를 조합, 어떤 부분을 변경할지 지정.

>> 선택자

=> 선택자로 어떤 것을 장식할지 지정. HTML 태그의 이름, 클래스, ID 등을 사용해서 변경 대상 작성.

=> 중괄호 ‘{}’를 사용하여 감싸 입력한다.

>> 속성

=> 선택자로 지정한 부분의 무엇을 변경할지 작성. 문자색, 문자 크기, 배경 이미지. “:”을 써서 구분.

>> 값

=> 값으로 외관을 어떻게 변경할지 작성. 구체적인 내용을 값으로 지정. “;”을 작성해서 마무리.

**- CSS 작성 규칙**

=> 기본적으로 CSS는 대문자와 소문자를 구별하지 않지만, 소문자로 통일해서 입력한다.

=> 여러 개의 선택자에 같은 장식을 지정할 수 있다. 선택자는 쉼표(,)로 구분한다.

=> 여러 개의 장식을 지정할 때, 세미콜론(;)을 이용하여 속성을 구분한다.

=> 단위 지정하기. 절대 단위(px), 상대 단위(%, rem)

**- 요소 내부의 요소 지정하기**

여러 개의 선택자를 띄어쓰기로 구분하면, 지정한 요소 내부에 있는 요소에 장식을 지정할 수 있다.

‘div p { color : red; }’ => div 선택자 내부 p의 색상 속성 변경.

3-5. 문자와 문장 장식하기

**- 문자와 크기를 변경하는 “font-size 속성”**

=> 단위는 숫자(“px”, “rem”, “%”), 키워드(xx-small, x-small, ~, medium, ~ x-large, xx-large)

=> 본문의 문자 크기는 14~18px가 일반적이다. 타겟 사용자에 따라서 변할 수 있다.

=> 문자 크기 변화는 2~5 종류 정도만 사용한다.

=> 문자의 ‘점프율’이란 제목과 문장의 문자 크기 비율을 의미한다.

=> 점프율이 높으면 도약이 있는 즐거운 분위기를 내며, 반대로는 우아하고 조용한 분위기가 난다.

**- 폰트의 종류를 지정하는 “font-family 속성”**

=> font-family 속성을 사용해서 폰트의 종류를 지정한다. 지정한 폰트가 없으면 디폴트 폰트를 출력.

=> 폰트의 종류를 쉼표(,)로 구분하면 앞쪽부터 사용자가 가진 폰트를 찾아 지정한다.

=> 한글로 폰트 이름을 적으면 못 읽을 수도 있으므로 영어 표기를 함께 지정한다.

=> 한국어로 작성되거나 공백을 포함하는 폰트의 경우 큰따옴표(“”)로 감싸야 한다.

**- 폰트의 종류, 통일**

=> 명조체: 디자인적으로 정중하고 품격 있는 인상, 일반적으로 긴 문장에 사용하는 것이 좋다.

=> 고딕체(돋움체): 가독성이 좋으며 어떤 디자인에도 잘 어울린다. 제목, 짧은 문장에 좋다.

=> 장식 계열 폰트: 외관 장식 중심. 제목과 짧은 문장에 사용, 본문에는 사용하지 않는 것이 좋다.

=> 사이트 전체의 통일감을 위해, 디자인에 사용하는 폰트는 1~3가지로 제한하는 것이 좋다.

**- 문자의 두께를 바꾸는 “font-weight 속성”**

=> font-weight 속성으로 폰트의 두께를 지정한다. 두께 변화를 지원하지 않는 폰트는 변하지 않는다.

=> 키워드(lighter, normal, bold, bolder), 숫자(1~1000) 사이의 값.

=> 긴 문장에는 두꺼운 글자를 사용하지 않는 것이 좋다. 가독성이 떨어진다.

**- 줄 높이를 바꾸는 “line-height 속성”**

=> 줄 높이(행간)을 바꿀 때 사용한다. 줄 높이가 문자 크기보다 작으면 줄이 중첩됨을 주의한다.

=> normal(브라우저 설정 높이), 단위 없는 숫자(폰트 크기 기반 비율), 단위 있는 숫자(px, em, %)

=> 추천하는 줄 간격은 1.5 ~ 1.9이며, 전체적인 디자인에 따라 조정한다.

**- 문장 맞춤을 지정하는 “text-align 속성”**

=> 텍스트의 맞춤을 지정하며, 한국어는 디폴트로 왼쪽 맞춤이다.

=> left(왼쪽 맞춤), right(오른쪽 맞춤), center(가운데 맞춤), justify(양쪽 맞춤).

=> 중앙 맞춤의 경우, 줄의 시작 위치가 분산되기 때문에 가독성이 떨어진다. 따라서 짧은 문장에 사용.

3-6. 웹 폰트 사용하기

- 웹 폰트는 폰트 파일이 웹 서버에 있어 사용자가 별도로 설치하지 않아도 폰트를 출력할 수 있게 한다.

- Google Fonts 사용방법

=> Google Fonts 웹 사이트에 접속해서 폰트를 찾는다. Link: <http://fonts.google.com>

=> 사용하려는 폰트를 SELECT해서, Embed Font 코드를 폭사, HTML파일의 <head>에 추가한다.

=> “CSS rules” 코드를 폰트를 적용하고자 하는 요소에 적용한다.

3-7. 색 넣기

**- 색 지정 방법: 문자색과 배경색을 CSS로 지정할 때는 3가지 방법을 사용할 수 있다.**

1) 색상 코드 지정하기

=> 색상 코드라고 부르는 6자리 알파벳과 숫자를 지정한다.

=> 16진수(숫자(0 ~ 9), 알파벳(a ~ f))로 표현하며 대소문자는 구별하지 않는다.

=> 6자리 중 앞 두자리는 빨강(R), 가운데 두자리는 초록(G), 마지막 두자리는 파랑(B)을 나타낸다.

=> 0에 가까울수록 어두운 색이 되며, f에 가까울수록 밝은 색이 된다.

2) RGB값 지정하기

=> rgb(빨강색, 초록색, 파란색)을 지정하며, 각 색은 0 ~ 255사이의 숫자를 입력한다.

=> 0이 가장 어두운 색이고, 숫자가 높을수록 밝은 색이

=> 추가로 투명도를 나타내는 알파 값을 지정할 수 있다. e.g) rgba(빨강, 초록, 파랑, 투명도)

=> 투명도 알파값은 0 ~ 1 사이의 숫자를 입력하며, 0은 투명/1은 불투명이다.

=> 구글 ‘컬러 피커’ 도구 사용

3) 색상 이름으로 지정하기

=> 정해진 색 이름을 사용하여 지정한다. 그러나 한정적이어서 세세한 색 조정은 어렵다.

**- 문자 색을 지정하는 “color 속성”**

**- 배경에 색을 지정하는 “background-color 속성”**

=> 페이지 전체 배경에 색을 지정할 때는 <body> 태그에 지정한다.

=> 색상 코드: #(해시)로 시작하는 3자리 또는 6자리 수의 색상 코드를 지정한다.

=> 색상 이름: 정해진 색 이름을 지정한다.

=> RGB: rgb값 혹은 rgba로 추가로 투명도를 지정한다

- **배색 잘하기**

1) 색상, 명도, 채도

=> 색상: 색의 차이. 색상은 연결돼 있으며 이를 원형으로 나열한 것을 색상환이라고 부른다.

=> 색상환에서 반대편에 있는 색을 보색이라고 부르며 이웃한 색을 유사색상이라고 부른다.

=> 명도: 색의 밝기를 나타낸다. 명도가 높을수록 상쾌한 느낌을 주며, 낮을수록 진정되고 어둡다.

=> 채도: 색의 선명함을 나타낸다. 높을수록 선명하고 화려한 느낌, 낮을수록 차분하고 어른스럽다.

3-8. 배경 넣기

**- 배경에 이미지를 설치하는 ‘background-image 속성’**

=> 배경 이미지를 제대로 읽어 들이지 못한 경우를 고려하여 배경 이미지와 함께 비슷한 색상 배경 설정.

=> url(배경 이미지 경로 지정)[이미지파일 지정], none[배경 이미지를 사용하지 않음]

**- 배경 이미지 반복 출력 “background-repeat 속성”**

=> CSS는 디폴트 설정으로 배경 이미지를 가로세로 방향으로 반복해서 출력하여 화면을 꽉 채운다.

=> 이를 어떤 방향으로 반복할지 또는 반복하지 않을지를 결정한다.

=> repeat (가로세로 방향 모두 반복 출력), repeat-x (가로방향 반복 출력), repeat-y (세로 방향)

=> no-repeat (반복하지 않는다.)

**- 배경 이미지의 크기를 지정하는 “background-size 속성”**

=> 이미지의 크기를 지정한다. 원래 비율을 유지한 상태로 요소에 맞추거나 지정한 크기로 확대 축소.

=> 숫자 (px. rem, %), 키워드 (cover, contain)

**- 배경 이미지의 출력하는 위치를 지정하는 “background-position 속성”**

=> 배경 이미지를 출력하는 시작 위치를 지정한다.

=> 기본적으로 가로세로 방향 순서로 띄어쓰기로 구분해서 작성한다. 디폴트는 왼쪽 위(left top)

=> 숫자 (px, rem, %), 키워드 (left/center/right, top/center/bottom)

=> background-size는 background-position 바로 뒤에 슬래시(/)를 붙이고 넣어야 한다.

**- 배경 관련 속성을 한꺼번에 지정하는 background 속성**

=> background-clip / background-origin / background-size / background-color

=> background-image / background-position / background-repeat / background-attachment

3-9. 너비와 높이 지정하기

**- 크기를 지정하는 “width 속성, height 속성”**

=> 직접 텍스트를 감싸는 <a> 태그와 <span> 태그 등 인라인 요소는 변경되지 않는다.

=> 숫자 (px, rem, %), auto [관련된 속성의 값에 따라 자동으로 설정된다]

**- 너비를 auto로 지정하는 경우**

=> 블록 요소 (<div>태그, <p>태그)는 width를 지정해주지 않으면 요소의 너비는 최대 영역을 다 채운다.

=> 이는 width 속성의 초기값인 auto가 적용되기 때문이다.

=> 너비가 auto인 경우, 요소의 너비는 해당 요소를 감싸는 부모 요소보다 커질 수 없다.

**- 너비를 %로 지정하는 경우**

=> %지정 시 부모 요소의 너비에 대한 비율로 너비가 결정된다.

=> 따라서 부모 요소의 너비에 따라 너비가 결정된다.

**- 웹 사이트에서 사용하는 단위**

**1) 상대단위**

=> 브라우저의 출력 영역, 다른 요소에 지정한 크기를 기반으로 상대적으로 산출되는 단위

=> 기준이 되는 크기에 따라서 크기는 변화한다.

=> % (부모 요소를 기준으로 할 때의 비율 단위): 너비 혹은 폰트 크기

=> em (부모 요소의 크기를 기준으로 하는 단위): 부모 폰트가 16px라고 하면 16px = 1em

=> rem (루트요소(<html> 태그)의 크기를 기준으로 하는 단위): root + em

=> vw (viewport width, 뷰포트 너비 기준 비율 단위): 뷰 포트 = 브라우저 출력 영역, 100px이 1

=> vh (viewport height, 뷰포트 높이 기준 비율 단위)

**2) 절대단위**

=> 뷰 포트 또는 부모 요소의 크기에 영향을 받지 않고, 지정한 크기로 그대로 지정되는 단위

=> px (웹에서 사용되는 가장 일반적인 절대 단위)

3-10. 여백 조정하기

**- 여백의 개념**

=> 여백의 속성에는 margin (상자와 상자 사이의 거리), padding(상자와 콘텐츠 사이의 거리)

=> margin과 padding 사이의 테두리 선을 border이라 부른다.

=> 내용물의 너비는 width, 높이는 height라고 부른다.

**- 요소 밖의 여백 “margin 속성”**

=> 요소 주변에 여백을 추가한다. Margin 속성은 모든 모서리 여백을 지정할 때 사용한다.

=> margin-top(위), margin-bottom(아래), margin-left(왼쪽), margin-right(오른쪽)

=> margin 속성 만으로 위, 오른쪽, 아래, 왼쪽의 순서로 여백 크기를 지정할 수 있다.

=> 숫자 (px, rem, %), auto

**- 요소 내부의 여백 “padding 속성”**

=> 요소 내부에 여백을 설정한다.

=> padding-top(위), padding-bottom(아래), padding-left(왼쪽), padding-right(오른쪽)

=> padding 속성 만으로 시계방향 순서로 입력을 지정할 수 있다.

**- 여백을 사용해 그룹 만들기**

=> 사람은 여러 개의 요소를 가깝게 배치하면 연관된 것으로 인식한다.

=> 따라서 관련 없는 요소를 가깝게 붙여 두지 않는 것이 중요하다.

**- 테두리와 문자 사이에 여백 만들기**

=> 요소 박스의 테두리와 글자 사이의 간격을 적절히 사용하는 것은 디자인 적으로 중요하다.

=> 일반적으로 문자 크기의 1 ~ 1.5배 정도로 여백(padding) 설정을 하는 것이 가독성에 좋다.

3-11. 선 긋기

**- 선의 두께 “border-width 속성”**

=> 크기를 지정하면 모든 모서리에 두께가 일괄 적용.

=> 모서리에 따라 두께를 다르게 지정하고 싶다면, 공백으로 구분하여 시계방향 순서로 지정.

=> 키워드 (thin/medium/thick), 숫자 (px, rem, %)

**- 선의 종류 “border-style 속성”**

=> 선의 장식 방법을 지정한다.

=> none (선을 출력하지 않는다), solid (1 개의 실선), double (2 개의 실선). dashed (파선)

=> dotted (점선), groove (입체적으로 파인 선), ridge (입체적으로 튀어나온 선)

=> inset (감싼 곳이 안쪽으로 들어간 것처럼 보이는 선), outset (감싼 곳이 튀어나온 것처럼 보이는 선)

**- 선 색 “border-color 속성”**

=> 선의 색상을 지정한다.

=> 색상 코드 / 색상 이름 / RGB 값

**- 요소 주변에 선을 긋는 “border 속성”**

=> 선의 “border-width”, “border-style”, “border-color”를 한번에 지정할 수 있다.

=> “border”을 사용하면 모든 변에 적용되며 각각 “border-top” … “border-left”을 사용하면 적용 가능.

**- 선 디자인을 효과적으로 사용하기**

=> 선은 콘텐츠를 구분하고 싶을 때 많이 사용된다.

=> 문자색보다 옅은 색을 사용하면 콘텐츠를 읽는 데 방해되지 않는 선에서 깔끔해 보인다.

=> 배경색에 가까운 색으로 선을 사용한다.

=> 때에 따라 메인 색상의 선을 사용하여 디자인 강조로도 사용한다.

3-12. 리스트 장식하기

**- 리스트 마커의 종류 “list-style-type 속성”**

=> 리스트 항목 앞에 출력되는 마커를 “리스트 마커”라 부른다.

=> 따로 지정을 하지 않으면 순서 없는 리스트는 검은색 점(disc), 순서 있는 리스트는 숫자로 출력

=> none (리스트 마커를 출력하지 않음), disc (검은색 점), circle (흰색 점), square (검은색 사각형)

=> decimal (숫자), decimal-leading-zero (앞에 0이 붙어있는 숫자), lower-roman (소문자 로마 숫자)

=> upper-roman (대문자 로마 숫자), cjk-ideographic (한자 숫자), hangul (한글)

=> lower-alpha, lower-latin (소문자 알파벳), upper-alpha, upper-latin (대문자 알파벳)

=> lower-greek (소문자 그리스 문자)

**- 리스트 마커의 출력 위치 “list-style-position 속성”**

=> 리스트 마커의 위치를 항목을 기준으로 안쪽에 위치할지, 바깥쪽에 위치할지 지정한다.

=> 여백과 정렬 상태를 지정할 때 리스트 마커를 포함할지, 않을지가 달라진다.

=> 주요 값: inside, outside

**- 리스트 마커로 사용할 이미지 “list-style-image 속성”**

=> 전체적인 디자인을 맞추고 아이콘 또는 색을 적용하고 싶을 때 사용한다.

=> 너무 복잡한 이미지를 지정하면 콘텐츠에 방해될 수 있으니 주의하자

=> url(이미지 경로)

**- 리스트 마커와 관련된 장식을 한꺼번에 지정하는 “list-style 속성”**

=> “list-style-type 속성”, “list-style-position 속성”, “list-style-image 속성”을 한번에 지정한다.

=> “list-style-type 속성”과 “list-style-image 속성”이 함께 지정되면 image가 우선이다.

3-13. 클래스와 ID를 사용한 지정 방법

**- 클래스와 ID**

=> 클래스와 ID는 태그 내부에 작성할 수 있는 속성으로 모든 태그에 작성할 수 있다.

=> HTML에서 클래스와 ID를 지정하면 CSS에서 이를 사용해서 해당 부분만 디자인할 수 있다.

**- 클래스를 사용한 작성 방법**

1) HTML 파일에서 태그에 class 속성을 넣어주고 원하는 클래스 이름으로 값을 준다.

2) CSS 파일에서 마침표(.)와 클래스 이름을 붙여 작성하고, 적용하고 싶은 스타일을 작성한다.

**- ID를 사용한 작성 방법**

1) HTML 파일에서 태그에 id 속성을 넣어주고 원하는 id 이름을 작성한다.

2) CSS 파일에서 해시(#)와 id이름을 붙여 작성하고, 적용하고 싶은 스타일을 작성한다.

**- 태그 이름과 클래스/ID를 사용하는 작성방법**

=> 특정 태그를 지정하지 않으면 해당 클래스/ID로 지정된 스타일은 사용된 모든 블록에서 적용된다.

=> 특정 태그에서의 클래스/ID가 지정된 경우 적용하기 위해서는 마침표/해시 앞에 해당 태그를 넣는다.

**- 클래스 이름과 ID이름 규칙**

=> 클래스 이름과 ID이름은 원하는 대로 지정할 수 있다. 다만 몇 가지 규칙을 따라야 CSS가 반영된다.

=> 띄어쓰기(스페이스)를 넣으면 안된다.

=> 알파벳과 숫자, 하이픈(-), 언더스코어(\_)로 작성한다

=> 첫 번째 문자는 반드시 알파벳이어야 한다.

**- 한 태그에 여러 개의 ID와 클래스 붙이기**

=> 클래스 이름과 ID 이름을 작성하는 큰따옴표 내부에 띄어쓰기로 구분해서 작성하면 된다.

=> 클래스와 ID는 함께 작성할 수 있다.

**- 클래스와 ID의 차이**

1) 한 HTML 파일 내부에서 사용할 수 있는 횟수

=> ID는 페이지 내부에서 같은 ID 이름을 여러 번 사용할 수 없다.

=> 그래서 ID는 레이아웃을 잡는 등 어떠한 페이지에서도 변하지 않는 부분에 사용하는 경우가 많다.

=> 반면 클래스는 페이지 내부에서 여러 번 사용할 수 있다. 그래서 여러 번 사용하는 장식에 쓰인다.

2) CSS 우선순위는 id > class 이다.

3) 작성방법 해시(#)와 마침표(.)를 사용하는 차이가 있다.

**- ID를 사용해서 페이지 내부 링크 만들기**

=> 링크를 작성하는 <a> 태그를 사용해서 “#id\_name”이라는 링크를 적용한다.

=> 페이지 내부의 id\_name이라는 ID 속성이 있는 부분으로 점프하게 된다.

3-14. 레이아웃 만들기

**- 레이아웃이란?**

=> 다양한 콘텐츠를 사용자가 사용하기 쉽게 웹 페이지의 배치를 설계하는 것을 의미한다.

**- Flexbox로 수평 정렬하기**

=> “Flexible Box Layout Module”의 약자, 복잡한 레이아웃도 간단하게 작성할 수 있는 방법

1) Flex 컨테이너라고 부르는 부모 요소 내부에 Flex 아이템이라는 자식 요소를 넣어 HTML을 만든다.

2) 부모 요소에 “display : flex”을 추가한다

**- 자식 요소의 정렬 방향 “flex-direction 속성”**

=> 자식 요소를 어떤 방향으로 배치할지 flex-direction 속성으로 지정한다. 수평 또는 수직 정렬 가능

=> row(초기값) [자식 요소를 왼쪽에서 오른쪽으로 배치한다.]

=> row-reverse [자식 요소를 오른쪽에서 왼쪽으로 배치한다.]

=> column [자식 요소를 위에서 아래로 배치한다.]

=> column-reverse [자식 요소를 아래에서 위로 배치한다.]

**- 자식 요소를 줄 바꿈 하는 “flex-wrap 속성”**

=> 자식 요소를 한 줄 또는 여러 줄로 배치하고 싶을 때 사용한다.

=> 여러 줄로 지정할 경우, 자식 요소가 부모 요소의 너비를 넘는 경우 줄 바꿈 해서 여러 줄로 배치된다.

=> nowrap(초기값) [자식 요소를 줄 바꿈 하지 않고 한 줄로 배치한다.]

=> wrap [자식 요소를 줄 바꿈 한다. 여러 줄이 되면 위에서 아래로 배치한다.]

=> wrap-reverse [자식 요소를 줄 바꿈 한다. 여러 줄이 되면 아래에서 위로 배치한다.]

**- 수평 방향 맞춤 “justify-content 속성”**

=> 부모 요소에 빈 공간이 있는 경우 자식 요소의 수평 정렬 상태를 어떻게 할지 지정한다.

=> flex-start (초기값) [줄의 시작 지점에 배치한다. 왼쪽 맞춤]

=> flex-end [줄의 끝부분에 배치한다. 오른쪽 맞춤]

=> center [가운데 맞춤]

=> space-between [왼쪽과 오른쪽 끝에 하나씩 붙이고, 남은 부분들은 균등하게 분배하여 배치한다.]

=> space-round [왼쪽과 오른쪽 끝에 있는 공간도 포함해서 균등한 여백을 형성한 상태로 배치한다.]

**- 수직 방향 맞춤 “align-items 속성”**

=> 부모 요소에 빈 공간이 있는 경우 자식 요소의 수직 정렬 상태를 어떻게 할지 지정한다.

=> stretch (초기값) [부모 요소의 높이 또는 콘텐츠의 가장 많은 자식 요소의 높이에 맞춰 늘어난다.]

=> flex-start [부모 요소의 윗부분에 맞춰 배치한다.]

=> flex-end [부모 요소의 아랫부분에 맞춰 배치한다.]

=> center [중앙에 배치한다.]

=> baseline [베이스 라인에 배치한다.]

**- 여러 줄이 될 때의 맞춤 “align-content 속성”**

=> 자식 요소가 여러 줄이 됐을 때의 수직 방향 맞춤을 설정한다.

=> flex-wrap : nowrap의 경우 자식 요소가 한 줄이므로 align-content 속성이 의미가 없어진다.

=> stretch [부모 요소의 높이에 맞게 자식 요소의 높이를 늘여서 배치한다.]

=> flex-start [부모 요소의 윗부분에 맞춰 배치한다.]

=> flex-end [부모 요소의 아랫부분에 맞춰 배치한다.]

=> center [중앙 배치]

=> space-between [위아래 끝에 있는 요소를 끝에 붙이고, 남은 부분 균등 공간으로 나눠 배치한다.]

=> space-around [위아래 끝에 있는 여백을 포함하여 균등 공간으로 나눠 배치한다.]

**- CSS 그리드로 타일 형태로 정렬하기**

=> 타일형 레이아웃: 벽에 타일을 붙이는 것처럼 같은 크기의 박스를 균등한 간격으로 나열하는 레이아웃

=> CSS 그리드를 사용하기 위해서는 Flexbox와 같이 부모 요소와 자식 요소가 필요하다.

**- 그리드 아이템의 너비 “grid-template-columns 속성”**

=> 각 그리드 아이템의 너비를 지정해서 수평 정렬하게 만들 수 있다.

=> 한 열에 그리드 아이템을 여러 개 놓을 경우, 띄어쓰기로 구분, 필요 그리드 아이템 수만큼 너비 지정

=> 그리드 아이템 사이의 여백은 “gap 속성”을 사용한다.

**- CSS 그리드에서 사용할 수 있는 단위 “fr”**

=> fraction (비율), 부모 요소에서 자식 요소의 크기를 비율로 지정할 수 있게 해주는 단위

=> px로 너비를 지정하면, 너비가 고정되므로 화면 너비가 넓어 졌을 때 공간이 발생하는 문제발생

- 그리드 아이템의 높이 “grid-template-rows 속성”

=> 여러 줄을 만들고 싶을 때는 띄어쓰기로 구분해서 필요한 그리드 줄 수만큼 너비를 지정한다.

**- 보기 좋은 레이아웃이란?**

=> 레이아웃에 따라 전체적인 느낌이 크게 바뀌므로, 어떤 요소를 어떻게 배치할지 사전 검토가 필요하다.

=> 시선의 흐름 이해하기

1) 처음으로 방문한 웹 사이트, 이미지가 많은 웹 사이트는 “Z 법칙”로 전체적인 느낌을 파악한다.

2) 여러 번 방문한 웹 사이트, 정보량이 많은 웹 사이트는 “F 법칙”로 원하는 정보를 빠르게 찾아 간다.

=> 정보의 우선 순위 결정하기

1) 웹 페이지는 왼쪽 위부터 읽기 시작. 중요한 정보, 주목 요소를 왼쪽 위에 배치한다.

2) 반대로, 우선순위가 낮은 것은 페이지 아랫부분 또는 오른쪽에 배치한다.

=> 배치 면적 고려하기

1) 잘 보이게 하고 싶은 정보는 면적을 크게 만들어 주목도를 높인다.

2) 최상위 페이지의 메인 이미지는 크게 배치, 반대로 중요하지 않은 정보의 면적은 작게 한다.

**- 레이아웃별 참고 사이트 목록**

=> 메인 이미지를 크게 배치한 레이아웃: 보여주고 싶은 것이 확실하게 결정된 경우

=> 타일형 레이아웃: 많은 정보를 어느 정도 한 번에 출력하고 싶은 경우

=> 왼쪽 오른쪽으로 분할된 레이아웃: 같은 중요도의 요소를 나란하게 배치하는 경우

=> 대각선으로 대치하는 레이아웃: 활발함 또는 약동감을 표현하고 싶을 때 사용하는 레이아웃

3-15. 디폴트 CSS 리셋하기

**- CSS 리셋이란**

=> 브라우저는 다양한 종류가 있으며, 각 브라우저는 디폴트로 독자적인 CSS가 적용돼 있다.

=> 예를 들어 여백, 폰트 문자 크기 등은 각 브라우저에 따라서 다르게 지정돼 있다.

=> 따라서, CSS를 지정할 때 차이가 발생하는 경우가 있다. 그러므로, 디폴트 CSS를 리셋 하도록 한다.

**- 리셋 CSS 사용하기**

=> 리셋 CSS를 직접 만드는 것은 방대한 양으로 인해 어려울 수 있다.

=> 따라서, 외부 웹 사이트에 공개된 CSS 파일을 사용한다

=> “ress.css” 리셋 CSS는 모든 디폴트 CSS를 리셋 X 디폴트 스타일을 활용 브라우저간 오차를 줄인다.

**- HTML로 읽어 들이는 방법**

=> HTML 파일의 <head>에서 ress.css를 읽어 들인다.

=> ress.css 파일을 내려 받고 읽어도 되지만, 웹에 공개된 주소를 연계해서 사용한다.

=> <link rel = “stylesheet” href = “https://unpkg.com/ress/dist/ress.min.css>

=> 디폴트 CSS를 자신이 만든 CSS 아래에 작성하면, 자신이 만든 CSS를 덮어버려 없애므로 주의.

3-16. 자주 사용하는 CSS 속성 목록

- 생략