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

RealFaviconGenerator <https://realfavicongenerator.net/> 파비콘을 만들어주는 서비스

Chapter 4. 풀 스크린 웹 사이트 제작하기

4-1. 풀 스크린이란

**- 이미지 또는 동영상 등을 메인으로 화면 전체에 출력하는 레이아웃을 의미한다.**

=> 전달하고 싶은 내용을 확실하게 표현할 수 있으므로 인상적인 디자인을 만들 수 있다.

=> 풀 스크린 부분을 스크롤 하면 자세한 콘텐츠가 출력되는 패턴 구성도 있다.

4-2. 풀 스크린 페이지의 제작 흐름

**- 작성할 페이지**

=> 홈 페이지의 로고, 네비게이션 메뉴, 캐치 카피, 문장, 버튼 등의 구성을 구상한다.

**- 제작 흐름**

1) head 부분 작성하기: 웹 페이지의 정보를 작성하는 <head> 태그를 작성한다.

2) 로고, 네비게이션 메뉴 만들기: 페이지 위쪽에 배치할 로고와 네비게이션 메뉴를 만든다.

3) 콘텐츠 부분 만들기: 페이지의 중심이 되는 콘텐츠 부분을 만든다.

4) 파비콘 설정하기: 페이지가 완성되면 페이지의 탭에 출력될 파비콘을 만든다.

4-3. “head” 작성하기

**- 파일 준비하기**

**- HTML 골격 만들기**

**- “meta” 정보와 “title” 작성하기**

=> 내부 문자 코드를 지정하고, 웹 페이지의 설명을 추가한다.

=> 웹 페이지의 제목을 정한다.

**- 필요한 CSS 파일 읽어 들이기**

=> 디폴트 css, google font, 개인이 작성한 css 파일을 차례대로 연결해준다.

4-4. “header” 부분 만들기

- <body> 태그 내부에 <header> 태그를 작성한다.

=> 이어서 내부에 <h1> 태그로 로고 이미지와 <nav> 태그를 배치한다.

=> <nav> 태그 내부에는 내비게이션 메뉴로 사용할 리스트를 작성한다.

- CSS 파일 준비하기

=> 계층적으로 폴더, 파일 구성 [css, image, database …]

- 공통 부분 작성하기

=> CSS 파일의 공통 부분을 작성합니다.

1) @charset “utf-8”을 작성하여 인코딩으로 인해 발생할 수 있는 문자 깨짐을 방지한다.

2) <html> 태그에 문자 크기를 100%로 지정, 웹 브라우저의 디폴트 / 사용자 설정 문자 크기를 반영.

3) <body> <a> <img>에 추가적인 폰트 관련 설정, 문자색 설정 등을 설정한다.

- 로고 크기와 여백 조정하기

- 네비게이션 메뉴 장식하기

=> “display : flex”를 사용하여 내부의 <li> 태그를 수평으로 정렬하는 부분이 핵심이다. [flexbox]

=> “a : hober”은 링크 텍스트에 커서를 올렸을 때를 나타내는 가상 클래스이다. 해당 상황에 장식 지정

=> 로고를 왼쪽, 네비게이션 메뉴를 오른쪽 배치하기 위해, page-header에 flexbox를 설정, 수평 정렬.

=> 양쪽 끝으로 배치하기 위해 justify-content : space-between을 추가.

- 콘텐츠의 최대 출력 너비 설정하기

=> 현재 상태에서는 로고와 네비게이션 메뉴가 너무 멀리 떨어져 있다.

=> 그러므로, 콘텐츠 전체를 감싸는 박스에 최대 너비를 지정해서 큰 화면에서도 위화감 없이 출력한다.

=> <header class = “page-header wrapper”>

=> CSS에 “wrapper에 대하여 “max-width”로 최대 너비를 지정한다.

=> “margin : 0 auto”을 지정하면 박스를 화면 중앙에 배치할 수 있다.

=> “padding: : 0 4%”를 설정하면 스마토폰의 너비가 좁은 화면에서 양쪽 끝에 여백이 생겨 보기 좋다.

4-5. 콘텐츠 부분 만들기

- 문장과 버튼 설치하기

=> 페이지의 중심 부분에 배치할 문장과 버튼을 만든다.

=> HTML의 <header> 태그 아래에 <div class = “home-content wrapper>부분을 추가한다.

=> 그 안에 제목으로 사용할 <h2> 태그를 배치하고, 문장과 링크 버튼을 작성한다.

=> CSS로 “text-align : center”을 사용해서 중앙에 배치하고, 여백과 폰트 크기를 지정한다.

=> 이어서 CSS로 제목과 버튼을 지정한다. 주석을 사용하게 구별해주면 좋다.

=> “border-radius”는 테두리의 둥글기를 지정, :hover는 마우스 커서를 올렸을 때를 가상하는 클래스.

- 화면을 꽉 채우는 배경 이미지 배치하기

=> “header” 부분과 <div class = “home-content wrapper”> 부분을 <div>태그로 감싼다.

=> <div> 태그에는 “home”이라는 ID와 big-bg라는 클래스를 지정하고 여기에 배경 이미지를 설정한다.

=> CSS를 사용, “main-bg”에 background-size : cover을 지정한다. 가로 세로 비율을 유지하며 채운다.

=> 이 지정은 다른 페이지에서도 사용할 것이므로 home이라는 id가 아닌 big-bg 별도 클래스에 지정

=> “home” id에 홈에서 출력하고 싶은 배경 이미지를 지정한다.

=> “min-height : 100vh”를 지정해서 이미지의 높이를 화면의 높이 만큼 지정한다.

4-6. 파비콘 설정하기

- 파비콘이란

=> 웹사이트를 북마크 하거나, 탭이 출력됐을 때 사이트 이름 옆에 출력되는 아이콘을 말한다.

=> Favorite(즐겨찾기) + Icon(아이콘) = Favicon(파비콘)

=> 사용자가 봤을 때, 한 눈에 어떤 사이트인지 구분할 수 있게 해준다.

- 파비콘 디자인

=> 브랜드를 연상시킬 수 있는 아이콘을 만들어 파비콘으로 사용하면 웹 사이트 디자인에 활용할 수 있다.

=> 파비콘은 작고, 사이즈 제한이 있으므로 테마 컬러를 사용하거나 간단한 도형을 사용하여 쉽게 디자인

- 파비콘 전용 이미지 준비하기

=> 일러스트레이터 또는 포토샵 등의 그래픽 도구를 사용해서 파비콘 전용 파일을 만든다. (PNG)

=> 사이즈는 고해상도 장치(2K이상의 해상도)에서도 잘 보일 수 있게 가로세로 32픽셀 이상의 사각형

- HTML “head” 내부에서 파비콘 읽어 들이기

=> <link rel = “icon” type = “image/png” href = “../images/favicon.png”>

- 파비콘을 만들어주는 서비스

4-7. 풀스크린 페이지 변경해보기

- 블랜드 모드로 배경 이미지에 색 겹치기

=> CSS 기능으로 여러 개의 이미지 또는 색을 다양한 방법으로 중첩해서 독특한 효과를 만들 수 있다.

=> background-color을 지정하고, 이어서 background-blend-mode로 블렌드 모드를 지정한다.

multiply (곱하기), screen (스크린), overlay(오버레이). darken(어둡게), lighten (밝게)

color-dodge (색상 닷지), color-burn (색상 번), hard-light (하드 라이트)

soft-light (소프트 라이트), difference (차이), exclusion (제외), hue (색조)

saturation (채도), color (색상), luminosity (광도)

- 배경을 그레이디언트 색상으로 설정하기

=> 화면 전체를 예쁜 그레이디언트 색상으로 출력하는 웹사이트도 많습니다.

=> 테마 색상을 잘 선택하면 예쁜 그레이디언트를 구현할 수 있다.

=> background-image 속성에 linear-gradient 값을 사용하고 쉼표(,)로 구분해서 다른 색을 지정

=> 원형 그레이디언트를 구현할 때는 radial-gradient를 사용한다.

- 배경 이미지에 블랜드 모드를 사용해서 그레이디언트 색상 섞기

=> background-image 속성으로 배경 이미지와 그레이디언트 색상을 콤마(,)로 구분, 두 개 지정

Chapter 5. 2-칼럼 웹 사이트 제작하기

5-1. 2-칼럼 레이아웃이란

- 2-칼럼 레이아웃의 장점과 구성 요소

=> 2-칼럼 레이아웃이란 “2-열로 나누어진 레이아웃”을 의미한다.

=> 칼럼 레이아웃은 콘텐츠의 양이 많은 뉴스 또는 블로그 등의 웹 사이트에서 사용하기 좋다.

=> 범용성이 높으므로 어떠한 형태로 만드는지 기억하자

=> 기본 구성: 페이지 상단의 이미지, 제목, 메인 콘텐츠 영역, 사이드바

- 2-칼럼의 너비 비율

=> 대부분은 메인 콘텐츠 영역과 사이드바를 “2:1” 또는 “3:1”로 분할해서 구성한다.

=> 디자인 또는 내용에 따라서 반으로 분할한다. 이를 “스플릿 스크린 (Split screen)이라고 한다.

5-2. 2-칼럼 페이지의 제작 흐름

- 제작 흐름

1) 페이지 전체의 제목 만들기: 페이지 상단에 이미지와 함께 페이지 제목을 출력한다.

2) “footer 만들기”: 화면 가장 아랫부분에 출력할 푸터를 생성한다.

3) 수평 정렬 설정하기: 2-칼럼 레이아웃 구성.

4) 메인 콘텐츠 영역 만들기: 메인 콘텐츠 내용을 구성한다. 날짜 부분 장식도 포함한다.

5) 사이드바 만들기: 카테고리 목록 리스트, 소개문 등의 추가 설명을 배치

6) 반응형 웹 지원하기: 스마트폰에서 보기 쉬운 형태로 출력하도록 필요한 기능을 추가한다.

5-3. 페이지 전체의 제목 만들기

- 파일 준비하기

=> 로고, 네비게이션 메뉴가 작성된 메인 페이지와 같으면 복제해서 저장한다.

- HTML 편집하기

=> 공통 부분을 남기고, 필요한 부분을 편집한다.

=> head 내부의 title 편집하기

=> 불필요한 콘텐츠 제거하기

=> ID 이름 변경하기

=> 제목 부분 추가하기

=> CSS로 장식하기

5-4. 푸터 부분 만들기

- 푸터 콘텐츠 작성하기

=> <body>태그에 푸터를 작성한다.

=> <small> 태그는 책임 사항, 저작권자 등의 주석을 나타낼 때 사용하는 태그, “&copy;”은 (c)로 출력.

5-5. 수평 정렬 설정하기

- 레이아웃 골격 만들기

=> Flexbox를 사용해서 메인 콘텐츠 영역과 사이드바를 수평 정렬한다.

=> 2개의 박스를 부모 요소로 감싸준다.

=> 메인 콘텐츠를 위한 <div> 안에 <article> 영역과 <aside> 영역을 작성하여 레이아웃 골격을 작성.

- 레이아웃 구성하기

=> 너비 지정하기: <article>과 <aside> 태그의 너비를 % 단위로 지정한다.

=> Flexbox로 수평 정렬하기: “.news-contents”에 display:flex를 지정해서 자식 요소를 수평 정렬한다.

=> justify-content : space-between을 추가하면 자식 요소가 양쪽 끝에 붙어 사이 여백이 생긴다.

5-6. 메인 콘텐츠 영역 만들기

- 기사 정보 부분 만들기

=> HTML로 기사 상단에 출력할 제목, 날짜, 카테고리 부분을 만든다.

=> 제목, 날짜, 카테고리 장식

=> 이미지와 본문 작성하기

5-7. 사이드바 만들기

- 제목과 문장 장식하기

=> <subtitle>, <sub-menu>, <ul>를 이용하여 카테고리 제목과 리스트를 작성한다.

=> CSS로 제목의 아랫부분에 <border-bottom>으로 선을 추가하고, 문자 크기와 여백을 조정한다.

5-8. 반응형 웹 지원하기

- 반응형 웹 디자인 (Responsive Web) 이란?

=> 출력 영역의 너비에 따라서 변화하게 디자인된 웹 사이트

=> 예를 들어, 데스크톱에서 볼 때와 스마트폰에서 볼 때는 너비가 크게 다르다.

=> 콘텐츠 내용을 따로 변경하지 않고, 장치의 크기에 따라 CSS 만으로 외관을 변경할 수 있다.

=> 작은 화면에서 보기 쉽게 하려면

(1) 칼럼 수 줄이기: 스마트폰 출력 시에는 1-칼럼으로 출력하는 방법을 사용한다.

(2) 네비게이션 메뉴 펼치기: 네비게이션 메뉴를 보이지 않게 하고, 메뉴 전용 아이콘을 통해 출력.

- 브레이크 포인트

=> 미디어 쿼리를 사용하면 장치의 화면 크기에 따라 스타일을 변경할 수 있다.

=> 이때 변경의 기준이 되는 화면의 크기를 “브레이크 포인트”라 부른다.

=> @media (max-width : 600px)의 경우, 브레이크 포인트는 600px 이다.

=> 단말기는 수직을 길게 두고 사용할 때의 너비 (= 장치의 짧은 부분의 너비)를 기준으로 포인트를 설정.

- 반응형 웹 지원하는 방법

=> “viewport” 설정하기

(1) 뷰포트란 다양한 장치에서의 출력 영역을 의미한다.

(2) 별도로 지정하지 않으면 스마트폰에서 출력했을 때도 데스크톱과 같은 너비에 맞춰서 출력

(3) 그 결과 글자가 너무 작아서 읽기 힘들어지며 가독성이 매우 떨어진다.

(4) 따라서 HTML의 <head>내부에 다음의 <meta>태그를 작성해서 출력 영역의 너비를 맞춘다.

<meta name = “viewport” content = “width=device-width,initial-scale=1”>

(5) 문자의 크기를 확인하며 추가적인 조정을 한다.

=> 미디어 쿼리 기본

(1) 미디어 쿼리는 출력하는 화면 크기에 따라서 다른 CSS를 적용하게 해주는 기능이다.

(2) 활용 예로, “화면의 너비가 600px”보다 작으면 문자의 크기를 작게 만든다”

(3) 사용자가 보는 환경에 맞게 스타일을 변경할 수 있다.

(4) @media “미디어 쿼리를 사용하겠다 선언”, 활용 예는 CSS 파일 참고.

- 문자 크기와 여백 조정하기

=> 미디어 쿼리를 이용하여 모든 페이지의 공통 부분을 CSS파일에 추가한다.

=> 600px 이하의 화면일 때 캐치 카피와 네비게이션 메뉴의 문자 크기, 콘텐츠 여백 등을 조절한다.

- 콘텐츠 수직 정렬하기

=> Flexbox를 사용해서 수평 정렬된 부분을 스마트폰에서는 수직 정렬이 되게 한다.

=> flex-direction을 사용하면 어떤 방향으로 정렬할지 지정할 수 있다.

=> 콘텐츠 영역과 사이드바를 수직으로 정렬한다. 추가로 각각의 너비를 100%로 변경, 화면을 채운다.

=> 문자의 크기와 여백을 조정한다.

5-9. 칼럼 페이지를 변경해보기

- 3-칼럼 레이아웃으로 설정하기

=> 실습에서는 기존 2-칼럼 레이아웃에 세로로 긴 배너 이미지를 추가하여 3-칼럼으로 변경한다.

=> 메인 컨텐츠를 구성하는 <div>에 추가한다.

=> CSS를 사용해서 “article”의 너비를 조금 줄여 균형을 맞춘다.

- 출력할 칼럼의 순서 변경하기

=> 출력할 칼럼의 순서를 변경하기 위해 CSS의 order 속성을 활용한다.

=> order 속성은 “display : flex” 가 지정된 요소의 자식 요소에만 동작을 한다.