Quest 01. HTML과 웹의 기초

## **Checklist**

* HTML 표준의 역사는 어떻게 될까요?
  + HTML 표준을 지키는 것은 왜 중요할까요?   
    1. 수정 및 운영관리 용이  
     콘텐츠의 올바른 구조화와 CSS로 시각표현을 통일하여 HTML, CSS를 따로 분리하여 제작하면 CSS 수정 만으로도 손쉽게 디자인의 수정이 가능하고 페이지 제작 시 부담 및 시간, 비용의 감소 효과가 있으며 관리에 용이하다.  
    2. 접근성 향상  
     웹 표준을 이용하여 만든 문서는 다양한 브라우징 환경에 접근이 가능하고 일반인 뿐만 아니라 장애인, 노인 등의 다양한 사용자가 어려움 없이 접근 가능한 페이지를 만들 수 있다.  
    3. 검색 엔진의 최적화(SEO)  
     meta 요소를 이용하여 정확한 문서 정보를 제공할 수 있고 의미에 맞는 마크업은 검색 시 효율성을 높일 수 있다.  
    4. File size의 축소, 서버 저장 공간의 절약  
     구조(HTML)과 표현(CSS)를 분리해서 개발하면 해당 페이지를 불러오게 될 경우, 브라우저의 캐시에 CSS가 저장되고 메모리에 상주하게 되어 동일한 CSS는 다시 불러오지 않기 때문에 HTML코드가 가벼워지게 된다.  
    5. 효율적인 마크업  
     CSS와 HTML 문서를 분리하여 제작할 경우, 불필요한 마크업이 최소화 되어 페이지 로딩 속도를 향상 시킨다.  
    6. 호환성 기능  
     기존 IE 브라우저에서만 작동이 가능했던 요소들이 웹 표준을 준수함으로써 다양한 브라우저(크롬, 파이어폭스, 사파리, 오페라 등)에서도 작동
  + XHTML 2.0은 왜 세상에 나오지 못하게 되었을까요?  
     XML과 HTML을 합성하여 더 확장된 표현으로 나왔지만 하위 호환성을 지원하지 않아서 이전의 태그들로 작성된 것들이 사용되지 않을 수 있다는 문제점이 있었고, 문법이 엄격하여 사용자들이 습득에 어려움을 겪어 점자 차용자에게 외면받게 되었다.
  + HTML5 표준은 어떤 과정을 통해 정해질까요?  
     W3C(world wide web Consotium)에서 정해져 공표된다.
* 브라우저의 역사는 어떻게 될까요?  
  + Internet Explorer가 브라우저 시장을 독점하면서 어떤 문제가 일어났고, 이 문제는 어떻게 해결되었을까요?  
     독점 시장에서 기술 발전은 이루어지지 않는다. 5년간 2번의 서비스 팩 업데이트 이외에는 기능 개선이 거의 이루어지지 않았기 때문에 느리고, 웹표준을 어기며, 기능 확장을 Active X라는 프로그램을 통해서 해야하기에 호환성이 떨어지는 등 문제점이 계속해서 대두되게 된다. 해당 문제는 계속 되다가 크롬, 파이어폭스 등 새로운 브라우저가 부상하면서 독점은 해소되고 있는 중이다.
  + 현재 시점에 브라우저별 점유율은 어떻게 될까요? 이 브라우저별 점유율을 알아보는 것은 왜 중요할까요  
     웹 브라우저 시장이 중요한 이유는 사용자들의 인터넷 사용정보를 수집할 수 있기 때문, 또한 웹 브라우저 시장을 장악하면 이를 활용한 다양한 비즈니스 모델을 만들어 낼 수 도 있다.  
    사용자 환경을 파악하고 크로스 브라우징에 대응하기 위해서 개발자는 항상 점검하고 호환성과 폴리필을 진행해야 한다.  
    (폴리필(polyfil) : 변경된 표준을 준수할 수 있게 기존 함수의 동작 방식을 수정하거나, 새롭게 구현한 함수의 스크립트)

- 2023 상반기 전세계 웹브라우저 점유율 Top 5

| 브라우저 이름 | 점유율(%) |
| --- | --- |
| Chrome | 64 |
| Safari | 19.86 |
| Edge | 4.92 |
| Firefox | 2.86 |
| Opera | 2.6 |

- 2023 상반기 국내 웹브라우저 점유율 Top 5

| 브라우저 이름 | 점유율(%) |
| --- | --- |
| Chrome | 49.86 |
| Safari | 17.97 |
| Samsung Internet | 15.18 |
| Whale Browser | 8.71 |
| Edge | 6.19 |

* + 브라우저 엔진(렌더링 엔진)이란 무엇일까요? 어떤 브라우저들이 어떤 엔진을 쓸까요?  
    - 렌더링 엔진의 역할은 요청된 컨텐츠들을 브라우저 화면에 표시한다. 기본적으로 HTML 및 XML 문서와 이미지를 표시할 수 있다. plug-in 이나 확장기능을 통해 다른 타입의 데이터들도 표시할 수 있다. (ex. PDF Viewer를 통한 PDF표시) 브라우저들은 각기 다른 렌더링 엔진을 사용한다. 오픈 소스 렌더링 엔진인 Webkit이 가장 보편적으로 사용되고 있다.
  + 모바일 시대 이후, 최근에 출시된 브라우저들은 어떤 특징을 가지고 있을까요?  
    - 웹 브라우저를 보다 소형화하면서 빠른 속도&적은 데이터 용량을 기반으로 하면서 디바이스별로 반응형 뷰를 제공가능한 브라우저들이 대두되고 있다. 또한, 기기 제조사들이 직접 브라우저 구현에 뛰어들면서 사용자 편의성을 고려한 애드 블록 등 확장 프로그램에 대한 고려와 모바일의 유저 인터페이스에 적합한 블루 투스 컨트롤, 자이로 센서 기능들을 브라우저 내에서도 가능하도록 연결지어 보다 확장성을 넓히는 데 주력하고 있다.  
     참고>> 반응형 웹은 하나의 웹사이트에서 PC, 스마트폰, 태블릿 PC등 접속하는 디스플레이 종류에 따라서 화면 크기가 자동으로 변하게 해준다.   
    뷰 포트(View Port)는 작은 화면의 스마트폰에서 PC의 웹 페이지 표시할 때 전체적인 페이지의 배율을 조정해준다. 반응형 웹에서 페이지의 배율이 아닌 CSS 스타일을 변경하고 싶다면 - 미디어 쿼리(Media Query)  
    >> 뷰 포트 설정 ex.  
    <meta name=”viewport” content=’width=device-width, initial=scale=1.0”>
* HTML 문서는 어떤 구조로 이루어져 있나요?  
   - 웹 문서는 보통 <!DOCTYPE html>로 시작해 <html>, <head>, <body>라는 3개의 영역으로 구성되어 있다.
  + <head>에 자주 들어가는 엘리먼트들은 어떤 것이 있고, 어떤 역할을 할까요?  
    - 웹 브라우저의 정보를 읽어주는 태그, <meta>, <title>태그를 사용한다.  
    - <meta>   
     웹 브라우저에는 보이지 않지만 웹 문서와 관련된 정보를 지정할 때 사용.  
     문서의 character 인코딩을 특정, 저자 및 설명 추가  
    - <title>  
     title 태그에서 지정하는 내용은 거의 모든 웹 브라우저의 제목 표시줄에 표시된다. 또한 해당 페이지의 방문자나 검색엔진은 제목 표시줄의 제묵을 보고 페이지의 전체 내용을 추측할 수 있다. 웹 사이트의 즐겨찾기를 지정할 때도 웹 문서 제목으로 추가된다.  
    - <link>  
     파비콘(웹 사이트에 포함된 작은 아이콘) 및 css 파일 불러오기 등
  + 시맨틱 태그는 무엇일까요?  
    Semantic은 ‘의미의, 의미론적인'이라는 뜻, 즉 시맨틱 태그란 의미가 있는 태그를 말한다.
    - 시맨틱 엘리먼트를 사용하면 어떤 점이 좋을까요?  
      - 웹 브라우저가 HTML의 소스 코드만 보고도 어느부분이 제목, 본문의 내용인지 파악할 수 있다. 또한 화면 장독기와 같은 보조 기기에서 사이트의 구조를 제대로 이해할 수 있다. 즉 웹 사이트 사용자에게 보다 더 정확한 내용을 전달할 수 있다.  
      - 문서의 구조가 정확히 나눠지므로 PC나 모바일의 웹 브라우저와 여러 스마트 기기의 다양한 화면에서 웹 문서를 표현하기가 쉽다.  
      - 인터넷에서 웹 사이트를 검색할 때 필요한 내용을 정확히 찾을 수 있다.(웹사이트의 본문 내용을 검색해야 한다면 메뉴,푸터 영역이 아니라 본문 영역에서만 검색할것이다)
    - <section>과 <div>, <header>, <footer>, <article> 엘리먼트의 차이점은 무엇인가요?  
      - <section> 웹 문서의 콘텐츠 영역을 나타낸다. <article>태그와 비슷해 보이기도 하지만, <section> 태그는 몇 개의 콘텐츠를 묶는 용도로 사용하고 <article>은 블로그의 포스트처럼 독립된 콘텐츠로 쓴다.  
      - <div> 는 id나 class속성을 사용해서 문서 구조를 만들거나 스타일을 적용할때 사용. 즉, 영역을 구별하거나 스타일로 문서를 꾸미는 것이다.  
      이때, 단순히 스타일을 적용하려고 콘텐츠를 묶으면 <section>대신 <div> 태그를 사용한다.  
      - <header> 사이트의 머리부분, 사이트 전체의 헤더도 있지만 특정 영역의 헤더도 있다. 사이트에서 헤더는 주로 맨 위쪽이나 왼쪽에 있으며, 검색 창이나 사이트 메뉴를 삽입한다.  
      - <footer> 웹문서 맨 아래쪽 영역. 사이트 제작정보나 저작권 정보, 연락처 등을 넣는다. 푸터 영역에는 <header> 태그를 비롯하여 <section>, <artical>등 다른 시맨틱 태그를 모두 사용할 수 있다.
  + 블록 레벨 엘리먼트와 인라인 엘리먼트는 어떤 차이가 있을까요?  
     - 블록 레벨 엘리먼트 : 무조건 한 줄을 점유하고 있고, 다음 태그는 무조건 줄바꿈이 적용된다.  
     - 인라인 : text 크기만큼만 공간을 점유하고 줄바꿈을 하지 않는다.

## **Advanced**

* XML은 어떤 표준일까요? 어떤 식으로 발전해 왔을까요?
* YML, Markdown 등 다른 마크업 언어들은 어떤 특징을 가지고 있고, 어떤 용도로 쓰일까요?  
  XML(eXtensible Markup Language)은 HTML처럼 데이터를 보여주는 목적이 아닌, 데이터를 저장하고 전달할 목적으로만 만들어졌다.  
  XML은 웹 문서를 구조화하는 표준형식으로 1996년 W3C가 제안하였다.  
  가장 먼저 나온 마크업 언어인 SGML(Standard Generalized Markup Language)은 1969년 미국 IBM이 사내 주요 문서를 쉽게 교환하고 관리하고자 만든 것으로 1986년 ISO 표준으로 제정되었다. SGML은 특정 문서를 정의하는 데 필요한 태그를 임의로 생성하여 문서 구조를 정의할 수 있으나 너무 복잡하여 활용하기 어렵고 이를 지원하는 소프트웨어를 개발하기가 어렵다는 문제가 있다. 한편 HTML은 웹 브라우저에서 웹 문서의 각 부분이 보여질 형식, 특히 하이퍼링크를 표시하는데 사용된다.  
  하지만 단순성, 사용의 용이성이라는 장점을 가지면서도 제한된 몇 개의 태그만을 가짐으로써 문헌을 충분히 표현할 수 없다는 단점이 있다.  
  즉, HTML 태그는 문자의 크기나 색깔 등 표현방식에만 치중할 뿐 문서의 구조 정보를 표현하기 어렵다.   
  XML은 디자인에만 제한된 HTML과는 달리 독자적인 애플리케이션, 즉 RDF(Resource Description Format)을 갖는다.  
  즉, HTML과 SGML의 장점을 모두 살려 표준화 작업이 이루어진 웹 페이지 기술언어이다.  
  XML은 웹에서 구조화된 문서를 전송 가능하도록 설계되었기 때문에 문서를 구성하는 각 요소들의 독립성을 보장함으로써 문서의 호환성, 내용의 독립성, 요소 변경의 용이성 등의 특성을 제공한다.  
  즉, XML은 데이터를 보여주지 않고, 데이터를 전달하고 저장하는 것만을 목적으로 한다.