01

01-4. 용어

Overview

플랫폼

- 플랫폼이라는 용어는 광범위하게 사용되는 용어입니다.
- 소프트웨어 개발자 입장에서 플랫폼 용어를 한정 지어 보면 우리가 만드는 애플리케이션을 실행시켜 주는 환경을 플랫폼이라고 합니다.

- MSWord 도 어플리케이션이죠. 그리고 윈도우 OS에서 실행되죠. 그러면 MSWord 라는 애플리케이션을 실행시켜 주는 플랫폼은 윈도우 OS다 라고 이야기 할 수 있습니다.

- 우리가 주 목적으로 하는 자바스크립트의 경우 자바스크립트로 만든 애플리케이션이 브라우저에서 실행된다면 브라우저는 우리 애플리케이션을 실행시켜 주는 플랫폼입니다.

- 자바스크립트로 백앤드 웹 애플리케이션을 만들 수 있는데 그렇다면 우리의 애플리케이션은 Node.js 에서 실행됩니다. 그래서 이 경우에는 Node.js 가 우리 애플리케이션의 플랫폼입니다.

네트워킹

- 전세계의 수많은 컴퓨터(개인 컴퓨터일 수도 있고 서버일 수도 있습니다.)들이 인터넷으로 연결되어 있고 인터넷에 연결된 컴퓨터끼리 상호 데이터를 주고 받는 것을 네트워킹이라고 합니다.

- 카카오톡 애플리케이션을 통해 친구와 채팅 데이터를 주고 받는 것은 인터넷 망을 통해 채팅 데이터를 상호 주고 받는 것입니다. 즉 인터넷에 연결되어 있는 친구의 카카오톡 애플리케이션과 나의 애플리케이션이 네트워킹을 통해 데이터를 주고 받는 것입니다.

W3C

- World Wide Web 의 약어로 웹 표준화 단체입니다. 웹의 표준을 책정하여 다양한 브라우저에서도 동일하게 작성 혹은 동작하는 웹 애플리케이션을 개발할 수 있게 하기 위한 단체입니다.
- W3C 의 웹 표준화 관련된 문서등은 https://www.w3.org/ 에서 확인할 수 있습니다.

웹 디자이너 vs 웹 퍼블리셔 vs 웹 개발자

- 프런트 웹 애플리케이션이 만들어지기 위한 직업군들 입니다.

- 웹 디자이너는 말 그대로 화면을디자인 하는 역할이며 웹 퍼블리셔는 디자이너가 구상한 디자인을 실제 HTML, CSS 조합으로 만들어 내는 역할을 합니다.
- 그리고 웹 개발자들은 웹 퍼블리셔가 만든 HTML 문서에 자바스크립트로 애플리케이션적인 동작이 가능하도록 프로그램을 작성하는 역할을 합니다.
- 그런데 프로젝트 환경에 따라 이 역할이 명확히 구분되어 있기도 하고 웹 디자인과 퍼블리싱을 같이 하기도 하고 개발자가 퍼블리싱과 개발을 같이하기도 합니다.

프로토콜

- 프로토콜(Protocol) 이란 통신 규약입니다.

- 클라이언트 서버 구조에서 애플리케이션끼리 네트워킹을 하며 각종의 데이터를 주고 받습니다.
- 데이터는 어떻게 구성되어 있고, 어느 부분의 데이터는 어떤 의미의 데이터인지에 대한 상호 규칙이 있어야 그 규칙에 맞게 데이터를 전송하고 받는 곳에서는 그 규칙대로 데이터를 해석해서 이용할 수 있는 것입니다.
- 이런 프로토콜은 여러가지가 있는데 가장 많이 알려져 있고 가장 많이 이용하는 프로토콜이 HTTP 이며 이 이외에도 TCP/IP, FTP 등 다양한 프로토콜 등이 있습니다.

HTTPS

- HTTPS 는 HTTP 프로토콜에 보안을 강화한 프로토콜입니다.
- HyperText Transfer Protocol over Secure Socket Layer 의 약어이며 일반 HTTP 프로토콜처럼 프런트와 백앤드 웹 애플리케이션이 상호 데이터를 주고 받는 것은 동일한데 HTTPS 는 네트웍 통신의 인증과 데이터 암호화가 추가된 프로토콜입니다.

도메인과 IP

- 네트워킹을 하려면 대상이 되는 컴퓨터의 네트워크 상에서의 주소를 알아야 합니다.
- 이 네트워크 상에서의 주소가 IP 입니다.
- 일반적으로 4개의 숫자가 dot(.) 으로 연결되어 111.222.333.444 형태로 구성되는 주소입니다.
- 그런데 이 IP 를 사람이 외우기는 힘듭니다.
- 그래서 IP 에 일종의 이름을 붙이고 그 이름으로 IP 가 이용되게 하는데 그 이름에 해당되는 것이 도메인입니다.

- 사용자들이 google.com 이라고 입력하게 되면 구글 서버와 네트워킹 하기 위한 도메인을 명시한 것이고 이 도메인에 해당되는 IP 주소를 DNS(Domain Name Server)에서 얻어와 네트워킹을 하게 됩니다.
- 결국 네트워킹을 위한 주소는 IP 이며 사용자 편의성을 위해 IP 에 이름을 붙인 것이 도메인입니다.

Node.js

- 쉽게 백 앤드 웹 애플리케이션을 개발하기 위한 기술 중 하나라로 정리해 주시면 좋습니다.
- 물론 정확하게 노드를 이야기 하자면 "자바스크립트 런타임" 이라고 이야기 해야하고, 백 앤드 웹 애플리케이션 뿐만 아니라 다양한 애플리케이션을 자바스크립트로 개발할 수 있게 하는 기술이라고 이야기 해야 합니다.
- 하지만 현 시점, 노드가 사용되는 대표적인 곳이 백앤드 웹 애플리케이션임으로 백앤드 웹 애플리케이션 개발 기술 중 하나라고 정리해도 틀린 말은 아닐 것 같습니다.

DOM Node

- HTML 문서에서 HTML 태그를 자바스크립트에서는 흔히 DOM 노드라고 부릅니다.
- 물론 태그(Tag), 요소(Element) 라고도 불리우지만 웹 문서를 구성하는 하나의 객체라는 의미로 자바스크립트에서는 흔히 DOM 노드라고 부릅니다.

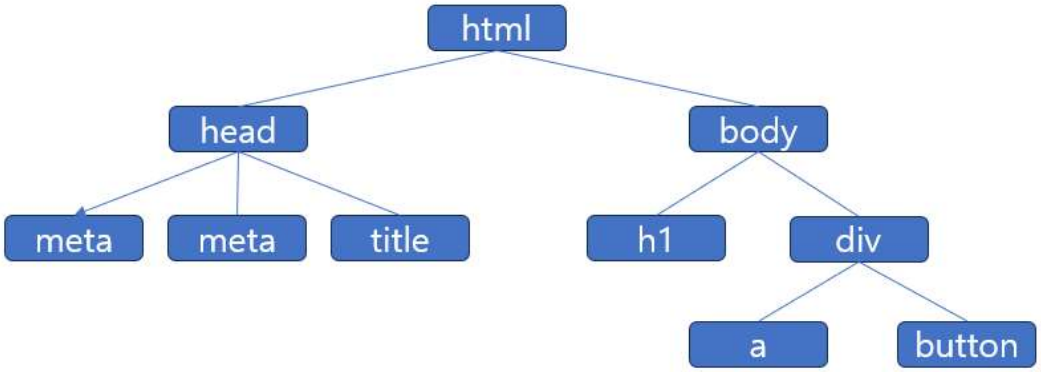
- 노드는 태그를 지칭하는 용어인데 앞에 DOM 은 무엇일까?
- 왜 흔히 DOM 노드라고 부르는 것일까?
- 그건 자바스크립트에서 HTML 문서를 흔히 DOM 이라고 칭하기 때문입니다. DOM 은 정확히 이야기 하자면 Document Object Model 의 약어이며 자바스크립트 혹은 HTML 에서만 사용하는 용어가 아닌 소프트웨어 개발 여러곳에서 범용적으로 사용되는 용어입니다.

- Document Object Model 을 어렵게 생각할 필요는 없고요, 쉽게 무언가 구성요소가 계층구조를 작성된 것을 지칭하는 용어라고 보시면 됩니다. HTML 문서도 계층 구조로 작성이 되죠.

DOM Node

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Hello World</h1>
  <div>
    <a>google</a>
    <button>button</button>
  </div>
</body>
</html>
```

DOM Node



- 계층 구조 혹은 트리(Tree) 구조로 전체 HTML 문서가 구성되어 있으며 이런 구조를 DOM(Document Object Model) 이라고 부릅니다.
- 그리고 이 DOM 구조의 하나하나의 구성요소(여기서는 태그들)를 노드라고 부릅니다.
- 그래서 흔히 자바스크립트에서 태그를 지칭할 때 DOM Node 라고 부르는 것입니다.
- a DOM 노드, button DOM 노드, 이런 식으로 용어를 사용합니다.

디버깅

- 소프트웨어 개발 공통 용어, 소프트웨어 개발 단계에서 생기는 오류를 찾아 수정하는 행위를 디버깅이라고 합니다.
- 오류가 발생했다는 것은 무언가 문제가 있다는 것이며 이 문제를 흔히 버그라고 부릅니다.
- 그리고 그 버그를 해결한다는 의미에서 오류를 해결하는 행위를 디버깅이라고 합니다.

IDE

- Integrated Development Environment 의 약어로 통합 개발 환경을 제공해주는 개발자 툴을 통칭하는 용어 입니다.
- IDE 에서 제공하는 대표적인 기능이 애플리케이션을 작성하는 코드 에디터 기능이고 거의 대부분 IDE 에서 코드 입력해서 개발함으로 개발할 때 이용하는 에디터가 IDE 라고 보면 됩니다.
- 그런데 Integrated 라는 단어가 있듯이 코드 에디터 이외의 다양한 기능이 추가된 도구라는 의미입니다.

API

- API 는 Application Programming Interface 의 약어이며 소프트웨어 개발 전반에 걸쳐서 사용되는 용어입니다.
- "애플리케이션 개발을 위한 인터페이스" 라는 의미입니다.
- 흔히들 어떤 API 를 이용해 무언가를 구현한다는 표현을 많이 합니다.
- 예로 들면 "자바스크립트 window API 를 이용해서 " 혹은 "공공 데이터 API 를 이용해서" 라는 표현을 합니다.

- 인터페이스를 이용하면 원하는 특정 기능 혹은 데이터등을 이용할 수 있다는 개념입니다.
- 애플리케이션 개발에 "API 를 이용해서" 라는 부분이 2가지 경우로 나누어 볼 수 있습니다.
-
- 애플리케이션 코드를 작성하기 위한 변수, 함수, 클래스, 객체
- 애플리케이션에서 필요한 데이터, 서비스등을 위한 네트웍 URL

API

- 우리가 코드를 작성해야 하는데 원하는 코드가 누군가에 의해 이미 만들어 져 있다면 우리는 그 코드를 이용만 하면 원하는 로직을 구현할 수 있습니다.
- 애플리케이션을 작성할대 이미 누군가에 의해 작성된 코드의 변수, 함수, 클래스, 객체등을 이용하는 것을 API 라고 하며 이 경우의 API 는 이용자 입장에서 이름을 알고 그 이름으로 이용하는 것이 됩니다.

- API 는 네트워킹 서비스와 관련된 용어로 사용됩니다.
- 서비스 혹은 데이터를 제공하는 업체의 URL 만 알고 있으면 되고 그 URL 로 요청을 하면 우리가 원하는 서비스 혹은 데이터를 제공 받을 수 있게 된다는 개념입니다.
- 흔히 "카카오 API 를 이용해서" , "기상청 API를 이용해서" 라는 표현을 합니다.

라이브러리

- 이미 만들어 놓은 것을 라이브러리(Library) 라고 부릅니다. 줄여서 Lib라고도 부르지요.

- 라이브러리는 또 크게 내장 라이브러리와 외부 라이브러리로 구분합니다.
- 내장 라이브러리란 우리 애플리케이션을 실행시켜 주는 플랫폼에서 제공하는 라이브러리입니다.
- 자바스크립트로 만든 애플리케이션이 브라우저의 자바스크립트 엔진에 의해 실행된다면 이 자바스크립트 엔진에서 기본으로 제공되는 라이브러리가 내장 라이브러리가 됩니다.
- 내장 라이브러리는 우리가 이용하기 위해서 별도의 준비 작업이 필요 없습니다.

- 내장 라이브러리는 대부분 어떤 소프트웨어 언어 혹은 애플리케이션을 위한 핵심적이고 기본적인 기능만 제공하며 애플리케이션은 그보다 더 많은 다양한 것들이 구현이 되어야 합니다.
- 다수의 외부 라이브러리를 이용해서 개발합니다.
- 물론 외부 라이브러리는 플랫폼에서 제공하는 것이 아님으로 이용하려면 애플리케이션에서 사용할 수 있도록 다운로드 및 설정이 되어야 합니다.

감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.

윌리엄 셰익스피어
William Shakespeare