프론트엔드 개발 수업 소개

**목차**

[1. 수업 계획 2](#_Toc156778737)

[1) 선수과목 2](#_Toc156778738)

[2) 수업 내용 2](#_Toc156778739)

[2. 개발도구 설치 3](#_Toc156778740)

[3. 웹프로그래밍 기술의 개요 4](#_Toc156778741)

[1) 웹서버 프로그래밍 4](#_Toc156778742)

[2) 프론트엔드 개발 기술 4](#_Toc156778743)

[3) 백엔드 개발 기술 4](#_Toc156778744)

[4. 프론트엔드 개발 기술의 종류 5](#_Toc156778745)

[1) jquery 5](#_Toc156778746)

[2) react.js 5](#_Toc156778747)

[3) vue.js 5](#_Toc156778748)

[5. 기타 6](#_Toc156778749)

[1) Javascript 언어 버전 6](#_Toc156778750)

[2) 풀스택 개발 7](#_Toc156778751)

[3) 하이브리드 앱 (hybrid app) 7](#_Toc156778752)

[4) 하이브리드 앱 개발 기술의 종류 8](#_Toc156778753)

[6. 요약 8](#_Toc156778754)

# 수업 계획

## 선수과목

### Python, C++, Java, Javascript 언어들 중 하나 필수

프로그래밍 언어를 모르면 진도를 따라갈 수 없다.

1학년 때 C 언어 하나 배운 것으로는 따라가기 힘들다.

### 웹개발 입문, 웹프로그래밍 중 하나 필수

위 과목 중 하나를 수강하지 않았으면 진도를 따라갈 수 없다.

위 두 과목 중 하나를 먼저 수강하라.

### 자료구조 선 수강 매우 권장

외국어를 배우는 것에 비유를 하자면,

Java 언어를 한 학기 배운 것은 영어 중학교 문법 정도를 배운 것.

자료구조를 한 학기 배운 것은 영어로 기본적인 문장 영작 훈련을 한 것.

프론트엔드 개발 수업은 영어로 논술 쓰기 훈련에 해당함.

영어 문법에 익숙하지 않거나 영어 영작 훈련이 안된 학생이,

영어 논술 쓰기 과목을 수강하면 아무리 열심히 공부해도 진도를 따라가기 매우 어려움.

## 수업 내용

### javascript 언어

웹브라우저에서 실행 가능한 유일한 언어인 javascript 언어를 한달 동안 빠르게 배운다.

### Vue.js 프레임웍

javascript 언어로 웹사이트의 UI를 구현하기 위한 객체지향 프레임웍

# 개발도구 설치

M1 맥북 개발도구 설치.docx

Windows 개발도구 설치.docx

위 문서를 참고하여 아래 SW들을 설치하라.

- node.js

- Visual Studio Code

- git 클라이언트

# 웹프로그래밍 기술의 개요

## 웹서버 프로그래밍

웹사이트의 대부분의 기능을, 서버에서 실행되는 코드로 구현하는 방식.

웹브라우저 창에 그려질 화면도, 서버에서 만들어서 출력한다.

버튼이나 메뉴를 눌렀을 때, 서버에서 어떤 코드가 실행되어야 하고,

그 실행 결과로 서버에서 출력되는 화면 내용이, 웹브라우저에 전달되어 웹브라우저 창에 그려져야 한다.

앱에 비해서 응답 속도가 크게 느릴 수 밖에 없다.

웹프로그래밍 기술의 시작은 바로 이 방식이었음.

JSP도 이 방식임.

100% 서버 프로그래밍 기술로만 웹사이트를 개발하는 것은 아니고,

웹브라우저에서 실행되는 약간의 코드는 javascript 언어로 구현하기도 함.

이때 사용하는 기술이 jQuery

## 프론트엔드 개발 기술

서버에서 실행될 수 밖에 없는 DB 관련 기능을 제외한 대부분의 기능을

웹브라우저에서 실행되는 javascript 코드로 구현하는 방식.

웹브라우저 창에 그려질 화면을, 웹브라우저에서 실행되는 javascript 코드가 만들고 제어하기 때문에,

화면 반응 속도가 빠르고 매끄럽다.

모바일 앱처럼 느껴지는, 빠르고 매끄러운 UI 구현이 가능하다.

실시간 게임, 워드프로세서, 동영상 플레이어 등 PC 앱과 같은 기능 대부분 구현 가능하다.

## 백엔드 개발 기술

프론트엔드가 담당하는 부분을 제외하고, 순수하게 서버가 담당해야하는 부분만 서버에 구현하는 방식.

UI 관련 기능을 프론트엔드가 담당하기 때문에, 서버의 코드는 DB 관련 기능만 담당하면 된다.

프론트엔드 개발 기술이 대세이기 때문에,

백엔드 개발 기술로 구현하는 것도 같이 대세가 되었다.

# 프론트엔드 개발 기술의 종류

프론트엔드 개발 기술의 구현 언어는 전부 javascript 이다.

웹브라우저에서는 javascript 코드만 실행되기 때문이다.

## jquery

### 장점

대부분의 기능이 서버에서 실행되는 웹서버 프로그래밍 방식으로 개발할 때

웹브라우저에서 실행될 간단한 javascript 코드를 구현하기에 좋다.

웹서버 프로그래밍에서 간단한 UI 기능을 javascript로 구현할 때 jquery가 가장 대표적인 기술이다.

### 단점

UI 전체를 개발하는 프론트엔드 개발에는 무리다.

### 요약

웹서버 프로그래밍의 보조 수단으로 간단한 기능만 구현하기에 좋지만,

프론트엔드 개발 기술로는 부족하다.

## react.js

### 장점

가장 많이 사용되는 완성도 높은 프론트엔드 개발 기술

하이브리드 앱 개발 기술(react native)의 완성도가 높다.

### 단점

난이도가 약간 높다.

### 요약

프론트엔드 개발자라면 react.js

## vue.js

### 장점

react.js 보다 약간 쉽고 편하다.

백엔드 개발자가 프론트엔드 개발 기술을 배우기에 적당하다.

### 단점

react.js 보다 점유율이 낮다.

### 요약

백엔드 개발자인데 프론트엔드 기술을 공부하고 싶다면 vue.js

# 기타

## Javascript 언어 버전

Javascript 문법 표준을 ECMAScript 라고 부른다. 줄여서 ES.

### ES5 (ECMAScript5)

Javascript 문법 표준의 5번째 버전을, ECMAScript5,줄여서 ES5 라고 부른다.

2009년 발표됨.

가장 많은 웹브라우저가 지원하는 보편적인 표준.

객체지향 문법이 빠져있다.

### ES6 (ECMAScript6)

Javascript 문법 표준의 6번째 버전

문법 표준 발표 연도를 붙여서 ES**2015**, ECMAScript**2015** 라고 부르기도 한다.

객체지향 문법 추가됨.

ES5 문법에 비해서 크게 변함.

프론트엔드 개발자라면 최소한 ES6 문법으로 개발해야 한다.

### 자잘한 버전 업

ES6 문법에서 자잘한 버전 업.

ES7, ES2016

ES8, ES2017

ES9, ES2018

ES10, ES2019

ES11, ES2020

ES12, ES2021

ES6에서 자잘한 버전업들이기 때문에,

위 표준들을 전부 뭉뚱그려서 ES6 라고 부른다.

### Typescript

마이크로소프트가 만든 언어.

Java 언어에는 자료형(type)이 있지만, Javascript에는 없다.

(예: int, double, String, ArrayList)

Java 언어에 있는 자료형과 객체지향 문법과 유사한 것을

Javascipt에 추가한 것이 Typescript 이다.

그래서, Java 개발자에게는 순수 Javascript 문법보다 Typescript 문법이 더 친근하다.

프론트엔드 개발자라면 Typescript 문법으로 개발하는 것이 바람직하다.

## 풀스택 개발

소규모 백엔드 서버 구현은 아주 쉽고 간단하기 때문에

프론트엔드 개발자가, 프론트엔드 뿐만 아리나 백엔드 서버도 같이 구현하는 경우가 많다.

이렇게 프론트엔드 백엔드 양쪽 다 개발할 수 있는 개발자를 풀스택(fullstack) 개발자라고 한다.

### 프론트엔드 출신 풀스택 개발자

frontend: Typescript + React.js

backend: Typescript + Node.js

### 백엔드 출신 풀스택 개발자.

frontend: ES6 + Vue.js

backend: Java + Spring

## 하이브리드 앱 (hybrid app)

프론트엔드 개발 기술을 사용하여 모바일 앱도 구현할 수 있다.

이런 앱들을 하이브리드 앱이라고 한다.

(예: 네이버 앱, facebook 앱)

### 장점

프론트엔드 개발 기술을 사용하여,

모바일 웹사이트와 안드로이드 앱, 아이폰 앱을 한 번에 구현할 수 있다.

### 단점

하이브리드 앱이 네이티브 앱보다 약간 느리다.

## 하이브리드 앱 개발 기술의 종류

### Phonegap

Adobe가 개발함.

ES5 + jQuery로 개발.

React Native 전에는 대세 기술이었음.

### React Native

facebook이 개발함.

ES6 or Typescript 언어 + React.js로 하이브리드 앱 개발.

현재 가장 널리 사용됨

### Xamarin

Microsoft가 개발함.

C# 언어로 개발.

Unity 3D 게임 엔진과 더불어, 가장 대표적인 모바일 게임 개발 기술.

모바일 게임 분야는 꽉잡고 있지만, 범용 앱 분야까지 대세가 될 확률은 낮다.

### Electron

프론트엔드 개발 기술을 사용하여 PC 데스크탑 앱을 만든다. (Windows, Mac, Linux)

ES6 or Typescript 언어 + React.js or Vue.js

Visual Studio Code가 Electron으로 개발된 하이브리드 앱이다.

### Flutter

Google이 개발함.

Dart 언어로 개발.

React Native 보다 빠르다.

모바일 앱뿐만 아니라, PC 데스트탑 앱도 만들 수 있다. (Windows, Mac, Linux)

React Native의 유력한 대안으로 기대되지만 아직은 아님.

# 요약

세상은 넓고 개발 기술의 종류는 많다.

그리고 빠르게 발전한다.

ES6 문법을 조금 살펴보고, Vue.js 기초를 배운다.