

코딩교육 열풍과 현주소

글. 김승욱 Rloha 대표



인공지능 스피커가 우리 삶에 깊숙하게 들어오고, 2020년이 코앞인 지금 그 어느 때보다도 사람들은 삶을 풍족하게 해주는 IT 기술과 밀접한 생활환경을 누리고 있다. 이에 따라 관련 기술을 배우고 싶어 하는 사람들은 늘어나기 마련인데 최근에는 코딩을 배우기 위한 학원이나 교육 프로그램이 꽤 많이 생겼다. 심지어 코딩지도사, 코딩교육 지도사 같은 민간 자격증도 생기다 보니 자녀들의 선행학습에 예민한 학부모들은 코딩을 미리 배워야 하는 것은 아닌지 걱정하기도 한다. 상황이 이렇다 보니 특수 아닌 특수를 타고 코딩학원이 우후죽순 생겨나는 것에 대해 우려의 목소리를 내는 뉴스 기사 또한 등장하였다.

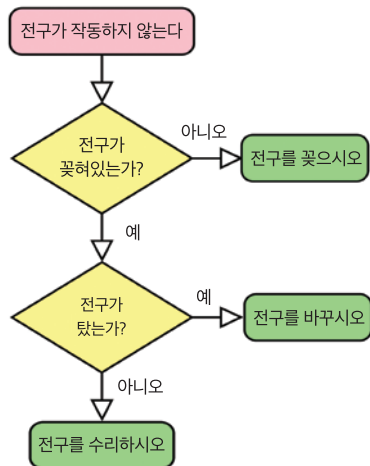
도대체 코딩은 무엇이고 왜 이렇게 화두가 되는 것일까?

컴퓨터, 스마트폰 등에서 동작하는 프로그램과 같은 것을 만들기 위하여 코드를 작성하는 것을 코딩이라 하는데, 좀 더 쉽게 말하자면, 컴퓨터가 특정 업무를 수행할 수 있도록 그 순서와 방법을 숫자, 영문자, 특수기호 등을 사용하여 지시 또는 명령서를 작성하는 것이라고도 할 수 있다.

앞에서도 비슷한 말을 했지만 스마트폰, 노트북, 태블릿, 스마트워치 같은 전자기기를 제어하거나 해당 기기에서 동작하는 프로그램을 만들고자 하는 사람들이 자연스럽게 이런 코딩을 접하고자 하는 사람들이 늘어날 수 있긴 하겠는데 몇 년 전부터 수요가 급증한 연령층은 다름 아닌 초/중/고 학생들이다. 국영수 주요 과목 공부하기도 바쁜 학생들, 그리고 그 부모들은 왜 이렇게 코딩에 관심을 가지고 여러 학교에서는 방과 후 교과목으로 코딩 교육을 실시하는 것일까? 그 중심에는 2015년에 교육부에서 발표한 'SW 중심사회를 위한 인재양성 추진계획'이라는 문서가 있다. 해당 문서에서 가장 중요한 표를 다음에서 살펴해보도록 하자.

구분	현행	개편안	주요 개편 방향
초등학교 (‘19년~)	실과 內 ICT 단원 (12시간)	실과 內 SW 기초 교육 실시 (17시간 이상)	· 문제해결과정, 알고리즘, 프로그래밍 체험 · 정보윤리의식 함양
중학교 (‘18년~)	‘정보’ 과목 (선택교과)	‘정보’ 과목 34시간 이상 (필수교과)	· 컴퓨팅사고 기반 문제 해결 실시 · 간단한 알고리즘, 프로그래밍 개발
고등학교 (‘18년~)	‘정보’ 과목 (심화선택 과목)	‘정보’ 과목 (일반선택 과목)	· 다양한 분야와 융합하여 알고리즘, 프로그램 설계

초·중·고 2015 교육과정 개편 내용 / 출처 : SW 중심사회를 위한 인재양성 추진계획. 2015.7.21.(교육부, 前 미래창조과학부)



중학교 교육에서 선택교과가 필수교과로 바뀐 것이 핵심이다. 아직 고등학교는 필수교과가 아니라고는 하지만 심화선택에서 일반선택으로 바뀐 것도 시사하는 바가 크다. 다음 개편에서는 필수교과로 개편될 가능성이 조금은 생겼다는 뜻이기도 하다. 이것이 호들갑이라고 하는 사람도 있을 수 있겠으나, 점점 IT가 생활 깊숙이 들어오고 있는 상황에서 해당 교과와 중요도가 증가하면 증가했지 떨어지지 않는 것이라고 본다.

다시 교육과정 개편 관련 표를 보도록 하자. 또 주목해야 할 단어는 알고리즘, 프로그래밍이 되겠다. 즉, 프로그램을 만들기 위해서 일련의 과정을 거쳐야 한다는 것이다. 물론 이전에도 알고리즘 기반 문제 해결을 위해서 좌측과 같은 순서도 또한 교육과정에 포함되어 있었다.



이전에는 논리적인 사고와 절차를 도형과 화살표를 활용한 순서도로 그려가면서 했지만 이를 엔트리(Entry), 스크래치(Scratch) 같은 블록 프로그래밍 소프트웨어를 활용하여 구현하거나 프로그래밍 언어로 많이 사용되는 C, Java, Python을 사용하는 등 난이도가 이전보다 올라갔다고 보면 되겠다. 2018년에 과외를 받던 한 초등학생에게서 입수한 가정통신문에 코딩교육을 소개하고 있길래 그중 일부를 스캔한 것이 좌측 그림이다.

이렇게 요즘 초등학교에서도 방과 후 과정으로 코딩교육이 운영되고 있고 다른 학교의 사례도 보고 싶다면 구글 같은 검색엔진에서 '코딩교육 가정통신문' 이 정도로 검색을 하면 관련 이미지나 문서가 줄줄이 나오니 찾아보면 되겠다. 앞의 사례처럼 방과 후 교육을 하는 것은 좋은데 학부모의 말을 직접 들어보니 실상은 썩 좋지 않았다. 블록 프로그래밍을 끝낸 다음에는 분명 실제 프로그램 개발에 사용되는 C, Java, Python 같은 언어를 배우는데, 이것을 지도할 교사가 부족하다 보니 그나마 가르칠 수 있는 사람이 할 줄 아는 언어로 진행한다는 것이다. 어떤 학교에서는 C, 어떤 학교에서는 Java 등 일관된 교육이 이루어지지 않는다는 제보였다. 사실 이것은 초·중등 교육에 국한된 것이 아니라 대학도 크게 다르지 않다. 물론 기존 컴퓨터공학과나 SW(소프트웨어) 관련 전공은 정착된 지 제법 되었기 때문에 좀 낫지만 데이터 분석 분야는 안타까운 경우가 굉장히 많다. 국내에서 손가락 안에 드는 학교조차도 교내 연구실에서 돌고 도는 코드나 조교 또는 교수님이 나누어 준 예제 코드를 보면 상당히 조악하게 작성되어 있는 경우가 제법 있다. 이런 상황이 벌어진 이유는 깔끔하고 효율적으로 작성된 코드가 아니라 코드로 만들어내는 결과물에만 초점을 맞췄기 때문이고 비IT 관련 학과는 아직도 체계적인 코딩교육이 이루어지지 않는 경우가 많기 때문이다.

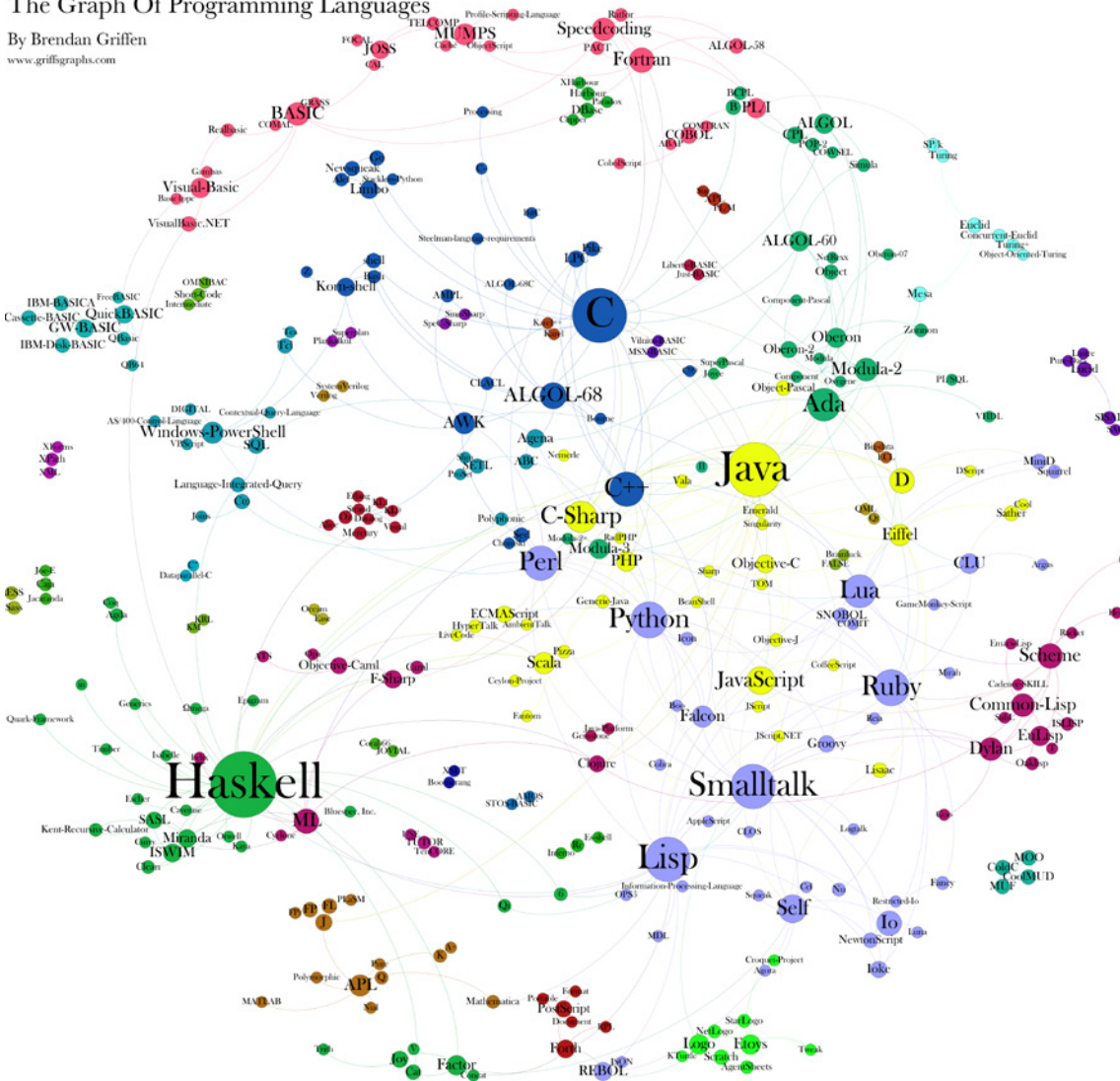
이 시점에서 이런저런 궁금증이 또 생길 수 있다. 도대체 프로그래밍 언어라고 언급된 C는 무엇이고 Python은 무엇인가? 학문도 여러 갈래가 있듯 프로그래밍 언어도 그 특징과 쓰임새가 제각기 다르다. 아무래도 모든 언어를 전부 다룰 수는 없으니 다음의 그림을 보며 프로그래밍 언어의 다양성을 간접적으로 체험해보도록 하자.



<https://neo4j.com/graphist/the-graph-of-programming-languages>

The Graph Of Programming Languages

By Brendan Griffithen
www.griffithgraphs.com



어떤 언어 간 긴밀한 연결을 보이는지 눈을 찡그려가면서 고생하지 마시고 인공지능 4차산업 융합과 혁명의 시대에 맞게 QR코드를 활용해보도록 하자. 해당 페이지로 이동하면 보다 자세히 볼 수 있겠지만 이 정보는 단순히 “이렇게 많은 언어가 존재하는구나!” 정도까지만 생각하셨으면 한다. 가끔 이런 자료를 보여주면 “어머!? Haskell이라는 것이 엄청 글자도 크고 중요한 것 같은데 우리 아이도 이제 이거 배우면 되나요???” 라고 말씀하시는 분이 있는데 조금만 더 알아보고 해도 늦지 않다.

프로그래밍 언어의 발전과 그 유래를 생각하자면 앞의 네트워크 그래프의 의미는 크겠지만, 이렇게 빠르게 변하는 시대에 길을 한 번 잘못 들면 되돌리기 매우 힘들다. 한 언어를 배우는 것은 결과적으로 보면 다른 언어를 배우는 데 도움은 되겠지만, 기왕이면 본인이 목표로 하는 직군에 가장 많이 사용되거나 본인의 취향에 맞는 언어를 선택하는 것이 보다 빠른 시간 내에 목표를 이룰 수 있도록 도와줄 것이다. 물론 해당 그래프에 적혀 있는 언어가 중요하지 않은 것은 아니다. 나 자신 또는 내 자식이 배울 언어를 선택함에 있어서 조금은 분별력을 가지는 것이 중요하다. 아무래도 학원이나 관련 교육기관의 담당자가 상황에 맞는 언어를 추천해주긴 하겠지만 아무래도 그 추천의 범위는 그 교육기관에서 유료로 제공하는 교과목 내에서 추천해주는 경우가 대부분이다. 상담을

받는 사람의 진로를 진심으로 생각해주면서 조언을 해주는 경우는 드물기에 다음의 내용을 보고 최소한의 배경 지식을 쌓아 놓도록 하자.

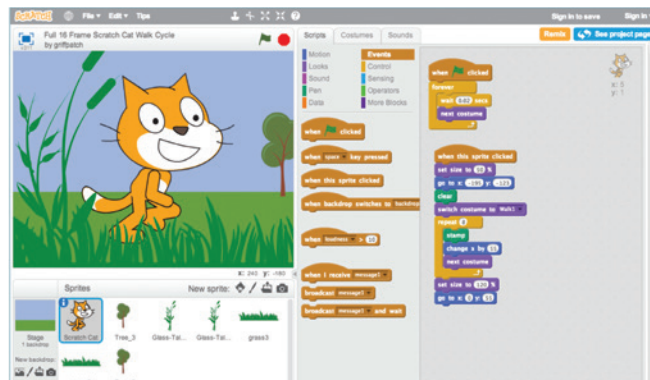
분야	언어
입문	Scratch, Entry
모바일 앱	Android - Kotlin, Java, iOS - Swift
데스크톱 앱	C++, C#, Java, JavaScript, Python
웹 개발	HTML, CSS, JavaScript, PHP
서버 / 인프라	go, Ruby, Python, Java, SQL, Scala
과학 기술	R, Python, C++, Fortran, Julia
디자인	Python, C++, C#

참고로 좌측의 표는 분야별로 주로 사용되는 언어를 기준을 몇 개 나열하였을 뿐 기재하지 못한 언어가 꽤 많으니 혹시 본인이 알고 있거나 사용하지 않은 언어가 있다고 해도 실망하지 않으셨으면 한다. 각 언어는 나름의 쓰임새가 있고 시대에 따라서 그 흐름이 바뀔 뿐이기 때문이다. 그럼 입문부터 하나씩 살펴해보도록 하자.

입문

요즘 어디 초등학생 코딩교육 입문이라고 하면 죄다 스크래치(Scratch), 엔트리(Entry) 라는 것을 한다. 이것은 프로그래밍 언어라고 하기에는 조금 애매한 위치에 있지만 사용자가 직접 영어/숫자/특수문자를 많이 입력하는 것이 아니라 블록을 조립하여 특정 기능을 수행하도록 하기 때문에 초보자로 하여금 쉽고 재미있게 코딩이라는 것을 배울 수 있도록 한다.

이런 블록프로그래밍의 대표 격이라고 할 수 있는 스크래치는 MIT 미디어 연구소에서 제작된 것으로 전 세계적으로 교육용으로 활발하게 사용되고 있다. 특히 초보자 코딩 교육 관련 자료를 보았다면 생소한 단어들이 많아 특별한 내용이 기억나지 않을 수 있지만 아마 노란 고양이라는 어쩐지 익숙하게 느껴질 수 있다.



옆 화면에서 스크래치로 코딩하는 것을 볼 수 있으며 원하는 동작에 따라 그에 맞는 블록과 특정 조건을 입력하여 고양이를 움직이게 하는 등 이런저런 명령을 내릴 수 있다. 특히 아이들에게는 쉬운 게임을 하는 듯한 인상을 주어 비교적 친숙하고 거부감 없이 코딩이라는 것을 접할 수 있기에 초보자에게 추천을 하는 것이고 최근에는 초등학생들이 해당 교육을 많이 듣는다.

※ 스크래치와 비슷하게 한국에서 만들어진 엔트리도 있다. 이 역시 블록 프로그래밍으로 쉽게 코딩을 익힐 수 있다.

스크래치와 엔트리 비슷한 블록 프로그래밍을 통해서 기초 교육을 끝냈다면 블록 프로그래밍을 활용한 장난감 로봇 제어에 도전할 수 있다. 아이들뿐만 아니라 성인도 마찬가지로 직접 조종기를 사용하여 자동차나 드론을 움직이는 것이 아니라 직접 입력한 값을 기반으로 로봇을 움직이게 만드는 것은 상당히 특별한 경험이다. 수년간 코딩교육과 로봇캠프를 진행해본 결과 특히 로봇을 활용한 교육을 할 때 사람들의 눈이 반짝반짝 빛이 나는 것을 몸소 느낄 수 있다. 블록 프로그래밍은 초보자로 하여금 흥미를 유발하기에 참 좋는데 전문적으로 하려면 언젠가는 다른 언어로 넘어가야 한다. 특히 아이들의 경우 로봇 교육을 받고 나면 굉장한 관심을 보이고 로봇공학자가 되고 싶어 하기도 하고, 게임을 만들어 보고 싶다고 하기도 한다. 둘 다 장래 희망으로 삼기에는 좋긴 하지만, 정말 잘하기 위해서는 수학 공부를 꽤 많이 해야 하기에 마냥 해맑은 아이들의 의지를 꺾기보다는 아이의 적성에 맞는 다른 길도 제시해 주는 것도 방법이다.

모바일 앱

앞의 블록 프로그래밍과 다르게 이제 본격적으로 코드를 작성하는 일반 프로그래밍 언어를 언급하고자 한다. 모바일 앱(Application)이라고 하면 크게 안드로이드(Android)와 iOS로 나누어진다. 앱을 구동하는 운영체제에 따라 사용하는 언어 또한 크게 갈리는데 iOS(아이폰, 맥)에서 구동하는 앱을 개발하려면 바로 Swift를 선택하면 되는데 안드로이드의 경우 Java와 Kotlin으로 나뉜다. Java는 발표된 지 약 25년이 지났으며 C언어와 같이 정말 많은 곳에서 사용되는 언어이다. 그렇다 보니 안드로이드 앱 개발뿐만 아니라 다른 분야까지 생각한다면 적극적으로 추천하는 언어이다. 그렇다고 쉬운 언어는 아니니 주의해야 한다. 그리고 Kotlin이라는 언어는 2017년에 google에서 정식으로 안드로이드 개발 언어로 지정하면서 급부상한 언어이다. Java와 상호운용이 잘 지원되며 안드로이드 앱만을 위해서라면 보다 간결한 Kotlin을 해보는 것도 좋은 선택이다.

이전에는 초보자가 프로그래밍 언어를 배워 무언가를 만들어보고 싶다고 하면 대부분 데스크톱 PC를 기준으로 이런저런 언어를 추천했지만 모바일 앱을 만들기 위한 생태계(커뮤니티, 유튜브, 관련 서적 등)도 잘 되어있고, 특히 아이디어가 좋다면 1인 개발자로 부업을 올릴 수도 있기에 개인적으로 앱 개발에 우선 도전하는 것을 추천한다. 그리고 갤럭시 워치나 애플 워치 같이 새로운 휴대용 단말이 등장함에 따라 새로운 시장이 형성되었기에 지금부터 공부하더라도 충분히 도전해볼 가치가 있다.

데스크톱 앱

사용자가 컴퓨터 모니터나 휴대용 단말의 화면을 통해서 눈으로 볼 수 있는 영역을 프론트 엔드(Front-End)라고 한다. 일반 데스크톱 PC에서 사용되는 프로그램을 개발하는 것은 프론트 엔드 개발이라고 하기에는 조금 무리가 있지 않을까 한다. 보통 데스크톱 앱은 디자인과 사용자 경험(UX)을 고려하는 것보다는 최대한 기능 구현이 최우선인 경우가 많기 때문이다. 아쉽게도 이 분야의 수요는 모바일 기기의 등장에 따라 점차 줄어들고 있는데 그렇다고 유망하지 않은 것은 아니다. 왜냐하면 대부분의 사람이 최신 기술에 몰리다 보니 윈도우 응용프로그램(앱) 개발자 같은 경우는 의외로 기술자가 없어서 상대적으로 귀한 존재가 되었는데 이전에 관련해서 개발자를 채용할 때 좋은 사람을 찾는 것이 문제가 아니라 기술을 가진 사람 자체가 없어서 꽤 애를 먹었던 기억이 있다.

이 분야에서는 보통 C언어 계열(C++, C#)을 하게 되는데, 요즘엔 웹 개발에서 사용되던 언어인 JavaScript 기반인 Electron이라는 것으로 이를 대체하기도 한다. 이 Electron으로 Mac, Windows, Linux 운영체제에 맞게 앱을 개발(정확하게는 패키징)할 수 있어 C언어보다 JavaScript를 하는 것이 더 효율적일 수 있다. Electron으로 만들어진 대표적인 크로스 플랫폼 데스크톱 앱은 Slack, Atom, Skype 등이 있으니 한 번 알아보는 것도 좋다.

웹 개발

스마트폰의 등장으로 앱 개발에 사람들이 몰리긴 했지만 2014년에 HTML5의 등장으로 또 한 번 주목을 받고 있다. 이전에는 스마트폰 앱을 Java 또는 Swift로 제작했다면 이제 HTML을 활용하여 앱의 콘텐츠 영역을 제작하여 배포하는 하이브리드 앱도 등장했기 때문이다. 이와 더불어 앱 개발에서 주목을 받는 PWA(Progressive Web App) 또한 HTML/CSS/JavaScript를 기반으로 제작하기 때문이다. 그리고 HTML5 기반의 게임도 무시할 수 없다. 기존에 게임은 PC 게임과 모바일 게임으로 양분되어 각 운영체제에 맞는 언어로 개발되었는데 이제 웹 개발에 사용되는 언어(W3C의 웹 표준을 따르는 프로그래밍 언어)로도 게임을 개발할 수 있고, 특히 설치 없이 웹 브라우저만 있으면 되니 사용자의 접근성도 보장하기 때문에 각광받고 있다.

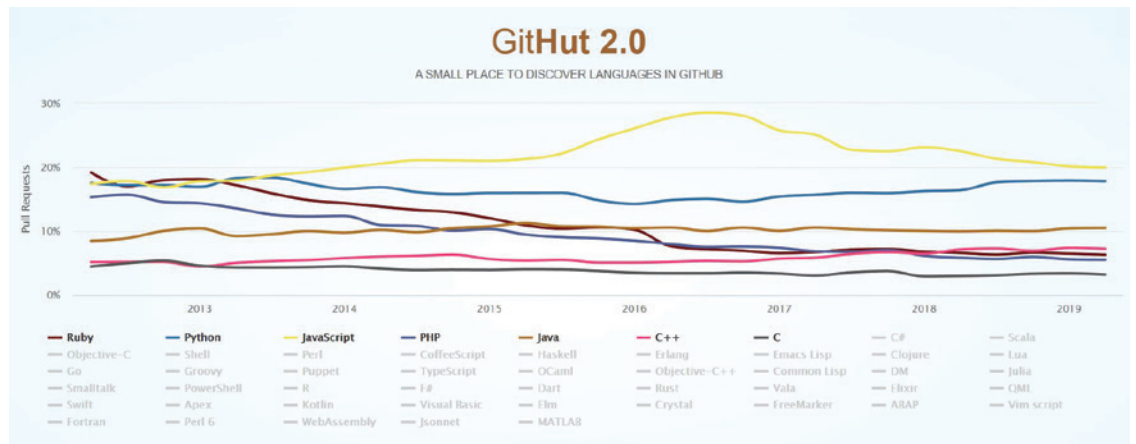
보통 웹 언어는 HTML/CSS/JavaScript를 세트로 묶어서 공부를 시작하며 HTML과 CSS는 다른 언어에 비해 난이도가 낮은 편이기 때문에 많은 사람이 웹 디자이너 또는 웹 퍼블리셔로 도전하기도 한다. 웹 부분은 아무래도 심미적인 요소가 많이 들어가기 때문에 디자인 감각보다는 기술적 구현에 자신 있는 분은 Front-End 개발자로 가는 경우가 있기도 하다. 이런 경우 PHP, 데이터베이스, 네트워크 등 기술 관련 공부량이 기하급수적으로 증가한다.

아직 소개하지 않은 Python이라는 언어가 있긴 하지만 필자는 스크래치나 엔트리 학습이 끝난 학생이라면 웹 개발 관련 언어를 도전하는 것이 괜찮지 않을까 한다. 다른 프로그래밍 언어는 학습 초반에 실용성이 의심되는 여러 가지 기호와 자료형을 외워야 하는 반면 HTML은 문법이 아주 간단하며, 코드를 작성함에 따라 그 내용을 바로바로 웹 브라우저를 통해 확인할 수 있기에 처음 코딩을 하는 사람도 흥미를 잃지 않고 꾸준히 할 수 있기 때문이다. 게다가 기술의 발전으로 좀 더 깊게 공부하는 경우 스마트폰 앱 개발과 웹 게임 개발에도 활용 가능하니 충분히 매력적이라고 할 수 있다.

그리고 JavaScript는 최근에 다양한 프레임 워크가 등장하면서 웹뿐만 아니라 서버 관련 부분인 back-end로의 확장, 인공지능을 학습하고 구현할 수 있도록 해주는 Tensorflow까지 사용이 가능하게 되면서 관련 분야의 사람들에게는 절대 무시할 수 없는 언어로 자리매김했기 때문에 기회가 되면 한 번 공부해보는 것은 어떨까 싶다. 프로그래밍 언어 순위 관련해서 정말 많은 자료가 있지만 JavaScript의 위용을 확인해볼 수 있는 자료를 준비했다.



※ QR코드를 통한 상세 내용 확인 가능



이제 서버/인프라, 과학 기술, 디자인 분야에서 사용되는 프로그래밍 언어를 다뤄야 하는데 과학 기술만 하더라도 그 양이 방대하기에 이번엔 모두 소개하기 어려워 다음으로 미루고자 한다. 다음 글을 기다리기 어려운 독자는 ‘프로그래밍 언어도감’이라는 책을 한 번 읽어보는 것을 권장한다. 해당 도서에서는 언어별로 간략한 소개와 함께 이해를 돕기 위한 삽화가 충분히 있어 출퇴근길에 가볍게 읽기에도 딱 좋은 서적이다. 📖