# 2020년 09월 07일 - 예상 질문

### Q. Block Coding이라고 해도, 개발에 익숙하지 않은 사람들은 여전히 개발하기 어려울 것 같은데 어떻게 보완하실 예정이신가요?

A. 먼저 시작하기 전, 개발에 도움이 될 튜토리얼을 만들어 둘 예정입니다. 또한 SNS 기능의 Q&A를 통해 개발에 중요한 팁들도 얻을 수 있으며, 조금 더 개발에 익숙해 진다면, 모듈에 적혀있는 함수나 변수, 상수들의 주석을 최대한 활용해서 직관성 있게 보여줘서 개발에 도움이 되게 할 예정입니다.

### Q. React Code 변경의 경우 시간이 굉장히 오래 걸릴 것 같은데 해결 가능한 문제인가요?

A. React 의 경우는 굉장히 제한된 범위의 기능만 지원할 예정입니다. 기본적인 버튼 배치나, 사진, 색상과 그림자, 기본적인 Navigation 등은 지원하겠지만, Timer, Camera 등의 어렵고 복잡한 이벤트 들은 지원하지 않을 예정입니다. 순전히 실제 기기에서 확인 가능한 UI Test로써의 기능을 지원 할 예정입니다.

### Q. 외부 모듈의 분석 방법은 어떻게 되나요?

A. Python의 경우, 원하는 모듈이 이미 저희가 만든 Block에 있는지 검색 하는 기능을 만들 예정입니다. 만약 원하는 모듈이 없다면, 해당 Python Module의 분석을 위해 Module 이름을 등록합니다.

등록 한 이후, 서버에서 자동으로 pip를 통해 다운로드 받은 뒤, Python 모듈의 소스 Tree를 생성하고, 가장 상위의 모듈부터 Python언어 Parsing 을 통해 함수 / 변수 / 상수 / Class 등을 구분해 냅니다.

구분 된 함수, 변수, 상수, Class 등은 각각 Block으로 다시 만들어서 서버에 등록되게 되고, 사용자는 추가된 모듈을 사용하면 됩니다.

### Q. 에러 분석이나 디버깅 시스템의 경우는 지원 되는 내용인가요?

A. 코드 실행 결과만 간단하게 보여줄 예정으로 생각하고 있으나, 시간이 남게 된다면 에러 분석창도 코딩 입문자들도 더욱 쉽게 확인 가능하도록 수정할 예정입니다.

## Q. 비슷한 Python Block Coding인 엔트리 파이썬과의 차이점은 무엇인가요?

A. 엔트리 파이썬의 장점은 다음 2개 라고 생각됩니다.

- 기존에 만들어진 코드들이 아닌 entry 라는 블록 코딩 라이브러리를 따로 만들어서 캐릭터가 실제로 움직이면서 흥미를 유발
- print 나 다소 딱딱한 함수 대신 몇번 말하기, 움직이기 등의 흥미 유발 함수들이 있음.

저희가 엔트리와 다르게 할 점은

1. 모든 "외부 모듈" 에 작동하는 블록 자동 생성 기능.
2. 실제 외부 프로그램에 바로 입력 가능한 파이썬 코드로의 변환
이 있을 것 같습니다.

즉, 흥미 유발을 통한 코딩을 학습하게 하는게 목적이 아닌, 바로 쓰고 싶은데 코딩을 모르는 디자이너를 타겟으로 한다고 하면 되겠네요~

<엔트리와 새싹 디발자 키우미의 차별성을 얘기하면서 추가적으로 이야기할 수 있는 것들>

앤트리와 차별성이 있음을 우선적으로 이야기 하고, 앤트리를 사용하는 유저들이 불편했던 점들 또한 추가적으로 개선하여 프로젝트 개발에 반영할 계획임도 알려주기.

→ 사용 연령이 대부분 10대 초반의 학생이다. 그렇기 때문에 정보공유, Q&A 게시판에서 원하는 정보나 대답을 찾기 어려움. 커뮤니티는 친목 도모 정도로 사용되고 있음.

커뮤니티 의견게시판의 내용을 확인해 본 결과, 기존 앤트리 사용자들은 UI가 깔끔하지 않아 보기 불편하다는 의견이 압도적으로 많았음. 이러한 부분들도 수정하여 사용하기 편한 개발 환경을 제공하기 위해 노력할 것임.

## Q.변환 과정은 어떻게 진행되는가?

A. 사용하는 블록에 따라 사용되는 용어가 할당되어 있고 이를 코드로 변환할 때 인지하여 작성된다.
들여쓰기와 같은 파이썬만의 문법을 이용하여 코드 변환에 활용할 수 있을 것으로 생각된다.
외부 프로그램을 활용하여 바로 입력 가능한 파이썬 코드로의 변환이 가능하다.

## Q. 기존의 스크래치, 파이썬 엔트리와 같은 블록형 코딩 툴이 있음에도 불구하고 이 프로그램을 필요로 하는 이유(차별점)?

A. 실질적으로 코딩에 도움이 될 수 있도록 프로그램을 만드는 것이 목적이므로 흥미 유발과 체험에 그치는 기존 프로그램과 차별점을 갖는다.

## Q. 앱을 지원하지 않는다 하셨는데 그럴 경우 sns기능 등은 오히려 제약을 받게 되는 사항이 아닌지?

A. SNS 기능은 사실 질문과 질문 답변을 받는 용도로만 사용될 것이기 때문에, 앱이 없다고 해서 정보 검색이 힘들어 지거나 정보 교류에 문제가 생기지는 않을 것 같다고 생각됩니다.,

## Q. 개발 기간이 생각보다 더 짧을 경우 이에 대한 해결책?

A. Python Editor가 지원하는 기능들을 하나씩 더 완벽하게 만들 예정입니다. 자동완성, 문맥 예측을 통한 자동 검색, 에러 문구를 더 쉽게 이해 할 수 있는 Block Code에서 에러를 찾아주는 기능, 출력 결과를 바로 바로 볼 수 있는 기능,~~ 또한 React Code 생성기 ( React Drawer 정도로 이야기 해도 좋을 것 같습니다. ) 에서도 간단한 버튼 이벤트 들은 추가 할 수 있을 것 같다고 생각됩니다.~~

## Q. 그 정도의 기능만 지원한다면 사실상 파이썬 강좌를 듣고 해결하는 것이 더 빠른 공부법이 아닌가?(공격적 질문)

A. 물론 파이썬 강좌를 듣고 해결하는 방법도 좋을 수 있습니다. Block Coding의 단점은 줄로 Code를 작성하는 것에 비해서 매우 느리다는 단점이 분명히 있습니다.

다만 저희의 궁극적인 목표는 언어에 신경쓰지 않고 소설을 쓰듯 블럭을 연결하고 이야기를 만드는 과정에서 Code가 생성되는 것에 목표를 두고 있기 때문에 파이썬 강좌를 듣고 줄로 Code를 작성하는 것과는 차이가 있다고 생각합니다.

## Q. 그래서 어떻게 파이썬 학습하는 시간이 줄어드는거지? 블럭형 코딩 학습 후 파이썬 재 학습의 경우 학습시간이 더 느는게 아닌가?

A. 물론 재학습의 경우 학습 시간이 더 늦는게 맞습니다. 그러나 저희 프로젝트의 목표는 비개발자들이 개발( 코딩 )에는 신경쓰지 않고 마치 이야기를 쓰듯 Block을 조립해서 원하는 목표 ( Code )를 가져가는 것에 목표로 하고 있습니다.

Python이 아무리 쉬운 언어라고 하지만, List에 추가하는 함수 이름이나 리스트 내부의 값들을 정렬하는 방법, 심지어는 변수 두개를 바꾸는 방법도 비개발자들에게는 어디서 부

터 찾아봐야 하는지 어려운 것도 맞습니다. 저희의 목표는 이런 비개발자들이 검색과 학습을 하지 않더라도, 클릭을 통해서 원하는 기능 수행하는 걸 목표로 하고있습니다.

## Q. 인터프리터를 서버에서 돌릴건지, 클라이언트에서 돌릴건지?

## Q. 서버에서 돌린다면 여러가지 이슈(보안이나 해킹 그리고 처리량. 지난 학기 코코딩 팀의 경우에는 인터프리터 머신이 하나 존재하여 이를 가상화하여 배포함.)를 해결할 방안등

A. 인터페이스 서버를 돌려야 하는 상황이 개발 하다보면 올 것 이라고 생각합니다. 아직은 공부가 더 필요하겠지만, Docker Container를 통해 가상 환경을 구현하고, 그 가상 환경에서 완성된 Python Code를 실행하여 Standard Output 만 읽어서 결과를 보여준다면 서버에 대한 공격 시도나, 서버 자원을 잡아먹는 코드에서 어느정도 자유로울 것이라고 생각합니다.

https://www.youtube.com/watch?v=sL_syMmRkoU&feature=youtu.be


중앙대학교 SW·AI TECH FAIR

중앙대학교 테크페어에 여러분을 초대합니다.

http://www.causw.com/capstone/109


## Q. 학습용이라면 학습에 맞게 프로젝트를 진행해야할 것 같음.

A. 물론 Q&A 게시판 등의 SNS 기능은 학습용으로 진행되는 기능이 맞습니다. 하지만 저희 프로젝트의 목표는 "개발을 신경쓰지 않는 디자이너들의 도구" 로의 기능을 더 추구하고 있습니다.

## Q. react로 한정한 이유가 있는지? 사실 파이썬 관련으로 교육을 진행하는 프로젝트를 만들면서 js 변환 툴이 섞이니 이질감이 든다.( Python 이야기

## 가 나오다가 React 가 나오는 것이 너무 뜬금 없다는 느낌이 든다. )

A. 원래 저희의 목표는 Python Block Coding의 개발이 끝이었으나, 이것만으론 조금 디자이너들에게 도움이 되기 어렵다고 생각했습니다. 또한 Python은 쉽고 좋은 언어이지만, GUI Tool의 기능이 조금 약한것은 사실입니다. 이를 보완하기 위해서 다른 언어의 Block Coding을 생각 해 보았으나 딱히 개발자가 아닌 사람들이 쓸만한 언어가 없었습니다.

다른 방향으로 생각하여, 디자이너들이 쓸만한 기능이 뭐가 있을지 생각하던 도중, 그림을 그려 그것을 바로 UI로 바꿔주는 기능을 생각 했습니다. 그러나 그림을 그리면 그것을 HTML, CSS로 바꿔주는 프로그램들은 이미 시중에 많은 좋은 프로그램들이 많이 있었습니다. 따라서 그것에서 더 나아간 GUI를 그리면 자동으로 프로젝트까지 만들어주는 기능을 생각 해 보았습니다.

## Q. 이 기능이 정말 잘 쓰일 수 있는 프로세스를 생각해보면서 기능을 생각해 보는것도? 일러스트레이터나 기타 디자인 툴과 연동이 되는거라면 정말 쓰일 수 있을 것 같다.

A. 현재는 C4D 라는 3D 모델링 프로그램에서 Python을 사용하는 것을 확인 했습니다. 최대한 그 프로그램에서 쓰기 편하게 바꿔보도록 하겠습니다.

## Q. 같이 협력하는 Tool은 쓰지 않는지?

## Q. 그래서 프로젝트가 추구하는 방향이 무엇인지?

A. 원래는 교육용으로 생각을 했었지만, 저희가 만든 아이디어를 모아보니 교육용 보다는 개발 툴로써 쓰이는게 좋을 것 같아 이쪽으로 방향을 내고 싶었다.

디자이너들이 코딩에는 신경쓰지 않고 디자인에만 신경 쓸 수 있게 해 주는 프로그램 혹은 개발을 배우지 않고 개발을 할 수 있게 해 주는 프로그램.