개발환경 세팅

12일 공부 후 9월 5일부터 단위 프로젝트, 기업체와 협약하여 만든 과정임.

기업체에서 원하는 주제에 맞게 선택하여 진행

Python + SQL, 크롤러(데이터 수집) 🡪 DB 적재

그 후 분석 및 인공지능

ZoomIt(판서 프로그램)

책 1권 당 1주일 정도

초거대 자연어 LLM 관련 집중

뒤로 갈수록 최신기술 사용, 논문에 나온 것 구현하기, IEEE논문 찾아서 진행

최종 프로젝트는 서버 등 전반적 활용하여 1달 쉬지 않고 일해도 부족함

인공지능 하는 경우 주피터 노트북, 미니콘다(가상환경), 코랩, VScode, IDE(spyder, pycharm 등은 안함)

가상환경과 colab의 차이 - 리소스를 구글을 사용하느냐 여부

인공지능 필요한 라이브러리르 사용, 크롤링을 위한 리소스를 가상 환경에 설치, 딥러닝을 위한 리소스를 다른 가상환경에 설치하여 버전 충돌 방지

동적크롤링(브라우저를 자동으로(매크로))은 로컬에서 해야함

코랩

요즘 대세, 페어 코딩(인공지능과 같이 코딩하기

Gpu 세팅

셀의 가장 마지막 결과만 보여줌

함수

받아서 돌려줌, 받기만 함, 주기만 함, 안 뚫려있음

코딩은 쉽게 해야 한다(가독성)

가독성이 높은 코드를 짜는 방법, 실수를 덜 할 수 있음

버그: 프로그램이 의도대로 되지 않음 – 로직이 잘못됨

에러: 프로그램이 원치 않고 죽는 것

버그의 3가지 종류

1. 항상 재현되는 버그(재현 경로를 알아 코드 수정 가능)
2. 간헐적으로 발생하는 버그(재현 경로가 있음, 찾기 어려움)
3. 한번 발생하는 버그

애니콜 개발자 출신이심(양산검증, stress test, 100대 1주일)

옛날삼성 – 문제 발생하면 stop

ToDo: 구글 계정 새로 파기

#, ##, ### 텍스트

프로그램: 명령어들의 집합

파이썬을 하는 이유

백엔드 – 자바, 리엑트

파이썬 – 수학적 표현에 능함, 행렬 연산, 데이터 사이언스에 적합

소스코드, 인터프리터(1줄씩 단위별 실행), 컴파일러(한번에 코드 전체 번역)

기술이 새롭게 나옴 🡪 검증 기간 (5~10년) 소요

파이썬 30년 정도 (최신 기술임)

7년된 모델 최신임

리눅스 명령어 cd\

기본 가상환경 base

Conda create -n mypjt1

Conda info –envs

\*는 활성화 된 것

Conda activate mypjt1

지금은 pc를 하나 새로 받았다고 볼 수 있음

리눅스 기반의 환경 새로 만들었음(virtual box 같음)

Conda install jupyterlab

정리

Miniconda 설치(다운로드)

Conda create -n <프로젝트명>

Conda info –envs

Conda activate <프로젝트명>

Conda install jupyterlab

Jupyter lab

* 노트북 클릭

생성한 폴더를 root로 잡음

코랩에서는 세션이 끊기는 문제가 있지만 여기선 파일 넣으면 살아있는 장점이 있음

Localhost 링크

Ctrl + / 🡪 주석 토글

코드 테스트를 위해 – 잠시 실행을 막음(디버깅)

가상환경은 디버깅이 어려움(IDE에서 더 수월)

이력서 업데이트 등 무엇을 했는지 적어놓지 않으면 막상 적으려 했을 때 몇 줄 쓰지 못함

노력에 대한 기록이 필수적임 – 쓸 것이 없음

어필할 수 있어야 함

작게 알게된 사실 적기

기술이력서\*\* 중요~~

습관들이기

언어마다 궁합이 맞는 툴이 있음

삼성 C#, MSSQL

키워드(Keyword)

파이썬에 이미 의미를 가지고 이는 예약어(예약석)

단어들이 이미 지정되어 있음

파이썬은 키워드를 바꾸는 것을 막아놓지 않음

다른 언어는 막아줌

바꾼 경우 초기화 해야함 – 파이썬의 최대 단점

Import keyword

Print(keyword.kwlist)

Ctrl + space 🡪 키워드 주어짐

함수: 미리 정해진 기능의 집합

호출(call)을 함

‘int’ object is not callable

함수 호출 불가능

Numpy, pandas

리스트 안에 원하는 데이터가 있는지 없는지 알고 싶음 🡪 in

Keyword.kwlist 🡪 집합

.iskeyword 🡪 기능

식별자 – 변수나 함수 이름

7up은 안 되고, sevenup은 된다

언더바는 특수한 변수 나타낼 때 사용됨

네이밍 룰 – 스네이크 케이스, 캐멀 케이스

Java – 캐멀 케이스(첫글자 소문자)

클래스 – 대문자로 시작(캐멀 케이스), 문법은 아님(권장사항)

함수 - 뒤에 괄호가 있다

변수 - 괄호가 없는 단어

지적 받는 경우

주석달기, 규칙(날짜, 문제 원인), 규칙, 내 스타일로 코딩 했을 때 (대형 시스템에 적용시 영향 미칠 수 있음; side effect)

공부할 때 – 새로 만드는 것

회사가면 – 기존 코드를 보고 분석/개선 🡪 더 어려움

입사해서 6개월~1년 버티면 크게 성장함

class Animal: 🡪 클래스 만들 때

Beautifulsoup() 🡪 클래스 사용할 때

변수 이름을 못 알아보면 의미가 없음

주특기

전달만 되면 됨

엑셀로 로마자 변환

한글 변수는 잘 사용하지 않음

문자열 String

숫자 number

불 boolean – True False

명령어, 데이터

자료형 type()

이스케이프 문자 – 다음 문자와 결합하여 function 진행

문자열 연산 🡪 + 만 가능

Print(“안녕하세요”+1) 🡪 python에서 불가, Java에서 가능

누구나 error나 버그는 남

IT에서 사용하는 에러코드는 정해져 있음

어떤 상황에서 어떤 에러가 나오고 해결하는지 고치며 실력 향상

구글, chatGPT에 대입

문자열 반복 연산자 \*

문자열 🡪 문자끼리 더하기

list라는 집합을 이루는 자료형에 대입

Java, python에서 쉽게 가능

인덱스

Score

|  |  |  |
| --- | --- | --- |
| 60 | 70 | 80 |

[-1]

마지막 인덱스

슬라이싱[:]

실행해보고 유추하기

[:] copy 효과가 있음

깊은 복사 – 값은 같지만 다른 참조

얕은 복사 – 값이 같고 같은 참조

개발자 만으로는 부족함, 제안서, 기획 등 설계하는 DB를 잡고 퀄리티 높이는 쪽에 종사하게 됨

점점 시야를 넓힐 수 있어야 롱런 함

B=A라고 하면 주소를 넘겨줌

Indexing, slicing(원본유지)

코딩에 집중하는 것보다 설계를 할 줄 아는 사람

나머지 연산자

모든 자연수 % 3 🡪 0, 1, 2 이런 case가 필요할 때

어떤 숫자 // 2

data가

수 // %

0 0 0

1 0 1

2 1 0

3 1 1

4 2 0

5 2 1

행 열

0 1

0

1

2

행열 구조의 data 시각화

0부터 5까지 연속적인 data가 주어졌을 때

Concatenate – 합치다

키보드로 입력 받으면 문자열로 받음

Input() – 사용자로부터 무언가 받아야 넘어감

Int()

Memory map – 변수가 메모리 위치 가리킴

= 할당 연산자

A = 10

B = a

Literal 리터럴 – 변수가 아닌 상수

Type cast – 자료형 변환

ValueError

Bool – 0만 False고 나머지 전부 True

조건식 – 참/거짓

문자열의 format() 함수 🡪 f 더 많이 사용

Upper(), lower() 🡪 왜 사용하는가? 어디 적용?

계속하시겠습니까? 🡪 입력받은 소문자, 대문자 y를 모두 같이 변경하여 처리 등

공백제거하기 strip() 🡪 양 옆의 공백 제거

“AB”

“AB “

Elastic search – 검색엔진(백엔드) 탑재 오픈소스

데이터에서 공백을 제거하고 찾아야 정확히 찾을 수 있으므로

문자열의 구성 파악하기 isOO() 🡪 참/거짓 판단

Find() rfind()

왼쪽, 오른쪽 찾기

Split() – 분리하기

In 연산자 – 있는지

박스모양 🡪 함수 의미함

자릿수 맞추기는 UI가 있기 때문에 콘솔창에 맞출 일은 거의 없음

Isdecimal() 조건문에 활용

Isalnum()

.isdigit()

Find()

Rfind()

Slicing 후 반복하여 -1 나올 때까지

다양한 환경 예제 풀어보기

빨리 적용하여 결과를 내는 사람이 우대 받음

In 연산자

12월 9일~ final project 시작 예정

프로그램 다운 냄새가 날 수 있도록

단위변환 조건문 반복문