

2025-04-23(수)

수업 오리엔테이션

cnn - 이미지, 병렬처리 rnn - 자연어 처리 Imm -  
SW 개발자 학습 방법

- Programming Language vs Language?
- S/W 개발자는 언제 많이 배울게 될까요?
- S/W 개발자는 뭐하는 사람인가요?
  - 문제를 해결하는 사람...
  - 문제 해결을 어떻게 할 것인가?
    - 논리적인 해결력..

## 1차 프로젝트 및 포트폴리오 자료 작성

- 데이터 분석 & 준비

## 서로가 지켜야 할 사항들

-

## 럼\*\*\*병에 적합한 생성형 AI개발

- 텍스트 생성, 번역, RAG
- 벡터 DB
- lanchain
- TensorFlow
- docker, aws
- 머신러닝, 딥러닝, 파이썬
- Spring Framework, SpringBoot, myBatis, Restful API
- Flask (chatbot\_views.py) - 생성형 AI, RAG
- Html5, Bootstrap5
- Database
- MSA (Microservice architecture)
- UML

## \*\*\* \*\* 인구예측 및 생성형 AI서비스 개발

- 머신러닝
- flask
- spring framework
- Hugginf Face I/F

- 텍스트 기반 생성형 AI
- TensorFlow, Java, jsp, html5, css3, javascript

## 지역소멸 및 대응방안 예측 서비스

- Lasso
- Flask
- scikit-learn
- Spring Framework

## AI 교육의 가치?

- 우수한 포트폴리오 완성

---

## Python 강의

---

- 파이썬
  - 1991년 네덜란드 출신 개발자 귀도 반 로섬이 개발한 언어로, 처음엔 C언어 기반으로 개발,
  - 이후 다양한 기능이 추가됨
  - 플랫폼 독립적인 언어
  - 플랫폼
    - 흔히 말하는 운영체제를 뜻함
    - windows, mac os , linux등이 이에 속함
    - virtual vox :
  - 파이썬은 어떤 운영체제든 상관없이 사용할 수 있는 언어
  - 인터프리터 언어
  - 객체 지향 언어(Object oriented language)
    - 클래스란?
      - 사회 현상중에서 데이터
  - 동적 타이핑 언어(Dynamic typing language)
    - 프로그램의 실행 시점에서 각 변수의 타입(type)을 결정 (Java ? No)
    - 프로그램의 실행 시점에 변수들의 메모리 공간을 자유롭게 할당 받을 수 있음(stack & heap 메모리)
    - 속도는 느리지만 메모리 공간을 할당 받고 사용한 것이 가능해짐
- ORM: 오브젝트를 그대로 DB에 저장이 가능
- Stack / Heap 메모리
  - [파이썬 메모리구조](#)
- 2. 변수의 이해
  - stack memory
    - local 변수(지역변수): 함수 호출이 끝나면 사라지는 변수
    - global 변수: 모듈 전체가 끝날때까지 유지되는 변수

- heap 도 stack 도 아닌 global namespace 라는 별도의 딕셔너리 구조에 저장
- 딕셔너리: key, value 쌍으로 이루어진 자료구조
  - 모두 단위로 'global namespace'가 생성됨
  - 이 네임스페이스는 내부적으로 dict타입이며, 변수이름을 key로 객체의 참조를 value로 저장
  - 전역변수는 이 네임스페이스에 등록되고, 이 정보는 global() 함수로 확인할 수 있음
- 모듈이란 파이썬 코드가 들어 있는 하나의 .py파일로, 다른 파이썬 코드에서 불러와서(import) 사용 할 수 있는 재사용 가능한 코드 단위임
  - 모듈은 네임스페이스를 제공하는 단위이며, 해당 모듈 안의 모든 이름은 그 모듈만의 네임스페이스에 존재
  - 네임스페이스: 이름(name)과 객체(object)간의 매핑(사전적 연결)을 저장하는 공간임
  - [\[python\] from, import, as 사용법](#)
    - 모듈(module)
      - 특정 기능을 .py 파일 단위로 작성한 것
    - 패키지(package)
      - 특정 기능과 관련된 여러 모듈을 묶은 것
    - 파이썬 표준 라이브러리(library)
      - 파이썬에 기본으로 설치된 모듈과 패키지
- 컴퓨터 구조
  - 폰 노이만 Architecture:
- 메모리(Memory: RAM(Random Access Memory))
  - 메모리는 실행되는 프로그램의 명령어와 데이터를 저장함
  - 메모리에 저장된 값의 위치는 주소로 알 수 있음

### 3. 함수

- 함수란 ?
  - 어떤 일을 수행하는 코드의 덩어리, 또는 코드의 묶음
  - 수학적으로  $f(x) = x + 1$
  - 함수의 장점
    - 논리적인 단위로 분할 가능
    - 코드의 캡슐화(encapsulation)
- 파이썬 함수의 구조
  - def - 함수를 만들때 사용하는 keyword
  - 매개변수는 함수에 입력으로 전달되는 값을 받는 변수
  - return - 함수의 결과값(리턴값)을 돌려주는 명령어? keyword

- 호출 한 함수로 제어권을 넘겨주는 역할
  - 파이썬 함수의 구조
    - 함수명은 소문자로 시작함
    - 함수명을 가능하면 짧게
    - 함수명이 길어지면 snake 방식
    - 함수는 들어온 입력값을 받은 후
  - implicit(암묵적)
  - explicit(명시적) - 머신러닝
  - 위치 인자
    - def add\_many(\*args):
      - \*args는 함수에 전달된 임의의 개수의 위치 인자들을 하나의 튜플로 처리할 수 있게함
      - 함수 호출 시 인자의 개수를 미리 알 수 없을 때 유용함
  - 키워드 매개변수
    - 매개변수에 초기값 미리 설정하기
      - 초기화하고 싶은 매개변수는 항상 뒤쪽에 놓아야 함
        - def say\_myself(name, man=True, age) x
        - def say\_myself(name, age, man=True) o
  - 함수 안에서 함수 밖의 변수를 변경하는 것은 프로그램 설계상 문제임
    - 코드의 의도 파악이 어려움
    - 디버깅 및 유지보수의 어려움(side effect)
    - 재사용성과 모듈화가 저하됨
    - 병렬 처리와 동시성 문제(race condition)
  - lambda 예약어
    - 함수를 한 줄로 간결하게 만들 때 사용
    - def와 동일한 역할
  - Type of Machine Learning
    - Supervised learning
    - Unsupervised learning
- #### 4. 클래스
- 클래스와 객체
    - 클래스(Class)
      - 사물과 추상화를 표현하기 위해 변수와 메서드로 이루어진 틀
    - 객체(Object)
      - 클래스로 부터 생성된 실체
      - 클래스로부터 생성된 것
      - 객체가 생성되면 그 객체는 메모리에 할당되어 실제로 사용됨

- 인스턴스(Instance)
  - 특정 클래스에서 생성된 객체, A클래스와 a인스턴스의 관계위주
  - 그 객체가 어떤 클래스의 실체인지를 명확히하고,
  - 특정 클래스에서 생성된 객체임을 강조할 때 사용함.
- 객체와 인스턴스의 차이점?
  - [python]-1 클래스와 객체 - 클래스의 필요성, 객체 vs 인스턴스 ★