|  |  |
| --- | --- |
| 교육 제목 | **Anaconda installation and Python** |
| 교육 일시 | 20211028 09:00 ~ 18:00 |
| 교육 장소 | 강의실 (오프라인 수업 참석) |
| **교육 내용** | |
| 오전 | 파이썬 기본  Program = pro + gram  (작성된것). (미리). (작성된 것)  순차처리의 정의  이진숫자 (binary digit)  파이썬(인터프리터 언어)의 장단점  - 장점 :  ; 대부분의 운영체제에서 동일하게 사용됨. (Java 도 유사)  ; 다양한 분야에서 활용 가능.  - 단점 :  ; 느리다. 단 최근 컴퓨터 성능이 좋아져서 연산이 많지 않다면 차이가 크지 않다.  (컴파일러 언어 ;; 먼저 번역 및 해석이 된 것을 실행을 진행/ 인터프리터 언어 ;; 실행을 하면서 번역 and 해석 동시)  Editing -> interpreting 과정을 거친다.  프로그램 구조  - 식별자 > 표현식 > 문장 > 프로그램  식별자 활용의 기준/룰  Snake\_case / CamelCase  Class ==> CamelCase  () 유무에 따라 변수, 함수로 구분  #, ‘ ‘ 은 주석 처리  ‘ ‘ 은 다중 레이어의 주석 처리 시 활용  다중레이어 선택 후, Cmd + / 를 입력하면 다중레이어 주석으로 표현된다. |
| 오후 | 파이썬 (Anaconda installation) 설치  Jupyter notebook 활용  Jupyter 신규 명령어 및 기본 활용법  / enter : 실행 및 cell 추가  cmd enter : only 실행  위아래 화살표 아이콘 = 명령 셀을 위아래로 순서 변경을 할 수 있다.  kernel 항목 내 restart 내역있음.  Run 시, 최종 명령만 결과를 보여준다.  Toggle in view : line number 를 보이게, 안보이게 할 수 있음.  Cel delete : 가위 모양의 아이콘  Background 실행 중인 terminal 은 실행기 이므로 닫아서는 안된다.  Terminal 에서 python을 입력하면 python을 통해 실행할 수 있도록 “>>>” 이 표기된다.  Python terminal을 빠져나올 시, exit () 를 입력  Cell 의 속성을 markdown으로 할 경우 설명을 입력할 수 있다.  Cell 선택 전 a 를 누르면 cell 이 뒷단에 생성  Cell 선택 전 b 를 누르면 cell 이 앞단에 생성  커서 없이 cell 선택 후 m 누를 시, markdown type의 cell 로 변경  커서 없이 cell 선택 후 y 누를 시, code type의 cell 로 변경  Running 에서는 실행되고 있는 내역을 볼 수 있으며, 불필요할 시 shotdown을 시킬 수 있다.  Markdown cell 도 ctr+enter 시 실행으로 주석처럼 볼 수 있게 된다.  파이썬은 대소문자를 구별하여 다르게 인식한다.  Python 신규 명령어 및 기본 활용 방법  Practice1)  #키워드 출력  import keyword  print(keyword.kwlist)  # . 이후에 tab키를 누르면 연관이 있는 명령어, 식별자가 완성될 수 있게 목록화한다.  명령어 위에 마우스포인터를 올리고, shift+tab을 누르면 함수 사용법을 보여준다.    python의 전반적인 정보가 필요하다면, Help 아이콘을 눌러서 확인이 가능하다.  Practice2) Snake & Camel  hello\_coding // HelloCoding  hello\_python // HelloPython  we\_are\_the\_world // WeAreTheWorld  create\_output // CreateOutput  init\_server // InitServer  init\_matrix // InitMatrix  —————  Str : 문자열  Int : 숫자  \ : escape 문자  \n : 줄바꿈의 의미  \t : 탭의 의미  #문자열 연산자 '+', '\*'  #문자열 + 문자열 => 문자열 이어 붙이기  #문자열 \* 숫자 => 문자열 숫자만큼 반복  print("안녕"+"하세요")  print(3\*"안녕하세요")  print("안녕하세요"\*3)  Zero index : start from 0  One index : start from 1  연산자  // 몫을 출력  % 나머지 값 출력  \*\* 거듭제곱  # 숫자 : 정수, 실수  # 연산자 : +, -, \*, /, //(몫), %(나머지), \*\*(거듭제곱)  # 연산 시, ()를 이용해서 우선순위 결정 가능  # 다음 레이어 대신 ; 을 이용해서 동일한 결과 출력 가능  print("10 + 5 : ", 10 + 5)  print("10 - 5 : ", 10 - 5)  print("10 \* 5 : ", 10 \* 5)  print("10 / 5 : ", 10 / 5); print("10 // 5 : ", 10 // 5)  print("10 % 5 : ", 10 % 5); print("10 \*\* 5 : ", 10 \*\* 5)  변수 만들어 사용하기  구조  변수 선언 >변수 할당 > 변수 참조 > input(), int(), float(), str()  - 변수는 data type이 존재하지 않는다.  - 복합대입연산자 a += 10 의 경우 더 속도가 빠를 수 있기에 자주 쓰인다.  - 복합대입연산자 숫자 문자 모두 가능하다.  -input function  - 입력될 때까지 잠시 프로그램 대기 == block  - 입력전 cell 번호에 \*이 보이는데, 이는 입력을 대기하는 중임을 표시함.  - 입력을 하지 않으면 \*상태로 걸려 있는데, 정지 버튼을 누르면 해당 cell이 더이상 대기하지 않고 멈춘다. |