|  |  |
| --- | --- |
| 교육 제목 | AI 프로젝트 기반 S/W 전문가 양성과정 |
| 교육 일시 | 21.11.03 |
| 교육 장소 | 비대면(집) |
| **교육 내용** | |
| 오전 | #복습  자료형  문자열, 숫자, bool  자료를 모아둔 타입  list, dictionary, tuple  자료형 변수 type 확인  type(변수)  문자열  문자열 정의  “” 또는 ‘’ 으로 묶음  문자열 연산  + : 문자열을 이어 붙인다.  / : 문자열을 반복  문자열 인덱싱  기본적으로 0부터 문자열 길이의 -1까지 인덱싱 가능  반대로 –(문자열길이) ~ -1까지로 인덱싱 가능하다.  문자열 슬라이싱 [a:b]  a<=index<b  a=0이면 처음 0번째 인덱스부터 b=0이면 마지막 인덱스까지 범위  문자열 길이 구하는 함수  Len(문자열변수)  숫자  숫자 자료형의 종류  정수, 소수  숫자 연산자  +,-,\*,/ (res=float), // (res=int), % (나머지 연산자), \*\*(제곱 연산자)    변수  파이썬 변수는 다른 언어 와 다르게 특정 자료형만 지원 하지 않고 모든  자료형을 저장 할 수 있다  복합 대입 연산자  += : 덧셈 후 대입  -= : 뺄셈 후 대입  \*= : 곱셈 후 대입  /= : 나눗셈 후 대입  %= : 나머지 구한 후 대입  \*\*= 제곱 후 대입  사용자 입력  <저장할 문자열 변수> = Input(“출력할 메세지”)  Int(변수) : 변수를 정수 값으로 변환  float(변수) : 변수를 소수 값으로 변환  str(변수) : 변수를 문자열로 변환  문자열 format함수  Print(“{} {} … {}”.format(변수, 변수, … 변수))  Format 함수는 문자든 숫자든 문자열로 변환하여 대괄호에 매칭시켜 넣는다  format함수 다양한 기능  {: (기호표기)(표현할 자리수).(소수점자리 표기시 자리수)(int d, float f)}    대소문자 바꾸기  <문자열>.upper() -> 전수 대문자로 변환  <문자열>.lower() -> 전수 소문자로 변환    문자열 양 공백 제거  <문자열>.strip() -> 문자열 양 옆 공백 제거  <문자열>.rstrip() -> 문자열 오른쪽 공백 제거  <문자열>.lstrip() -> 문자열 왼쪽 공백 제거  문자열 구성 파악하기 isXXXX()  <문자열>.함수()  문자열 구성 파악하기 함수 리스트 (p.101)  isalnum() : 문자열이 알파벳 또는 숫자로만 구성되어 있는지  isalpha() : 문자열이 알파벳으로만 이루어져 있는지  isdentifier() : 문자열이 식별자로 사용 할 수 있는지  isdecimal(): 문자열이 정수 형태인지  isdigit() : 문자열이 숫자로 인식 될 수 있는건지  isspace() : 문자열이 공백으로만 구성 되어 있는지  islower() : 문자열이 소문자로만 구성 되어 있는지  isupper() : 문자열이 대문자로만 구성 되어 있는지  문자열 찾기  <문자열>.함수(“찾으려는 문자열”)  반환값=찾은 인덱스  find() : 왼쪽부터 검색  rfind() : 오른쪽부터 검색  문자열과 in 연산자  문자열 내부에 찾으려는 문자열이 있냐 없냐 판단  <찾으려는 문자열> in <원본 문자열>  문자열 자르기  Split() : <문자열>. Split(“자를 려는 문자열”)  불 자료형  불 자료형 연산자  A == B : A와 B가 같으면 true  A != B : A와 B가 틀리면 true  A < B , A > B : 작고, 크면  A <= B , A >= B : 작거나 같거나, 크거나 같거나  논리연산자  Not  Or  And  If 조건문  If <조건문>:  <처리>  elif <조건문>:  <처리>  else:  <처리>  리스트  리스트 선언  <리스트 변수명> = [요소 , 요소 ,… 요소]    리스트 접근하거나 수정하기  접근 : <변수> = <리스트 변수명>[인덱스]  수정 : <리스트 변수명>[인덱스] = <변수>  리스트 연산자  1. 문자열 연산자 기능과 동일  2. len(리스트변수) > 리스트 요소 개수를 출력함  리스트 관련 함수  <리스트변수>.extand (<추가할요소들>) : 리스트에 여러 요소 붙임  <리스트변수>.append(<추가할요소>) : 리스트 마지막 요소 뒤에 추가  <리스트변수>.insert(index,<추가할요소>) : 리스트 특정 부분에 추가    리스트 요소 제거  인덱스로 제거  del : del <리스트변수>[인덱스]  pop() : <리스트변수>.pop(인덱스)  값으로 제거하기  Remove() : <리스트변수>.remove(값) : 값만 제거  clear() : <리스트변수>.clear() > 모두 제거  리스트 내부에 값이 있는지 없는지 확인  <값> in <리스트변수> |
| 오후 | For 반복문  for <반복자> in <반복조건>:  <반복 할 내용 >  반복 조건에 리스트 자료형을 넣으면 요소가 하나씩 출력 된다  딕셔너리  딕셔너리는 리스트와 비슷해 보이지만 결정적으로 키와 값으로 이루어진 차이점이 있다  <딕셔너리변수명> = { <키>:<값>,<키>:<값>, … , <키>:<값> }  딕셔너리 접근 및 수정  접근 : <변수> = <딕셔너리>[키]  키값 추가: <딕셔너리>[새로운키] = <새로운값>  값 추가 : <딕셔너리>[기존키] = <추가할값>  요소제거 : del <딕셔너리>[제거할키]    딕셔너리 내부에 키가 있는지 확인  in 키워드  <찾으려는키> in <딕셔너리>  get()함수  <리턴값>=<딕셔너리>. get(<찾으려는키>)  키값이 존재하지 않으면 “None” 리턴  for문과 함께 사용  for<키변수>in<딕셔너리>:  <처리할 내용>  range() : 범위 지정 함수  range(A) : 0~A까지 1씩 증가  range(A,B) : A~B까지 1씩 증가  range(A,B,C): A~B까지 C씩 증가  for반복문에 범위와 같이 사용  for <변수>in <범위>:  <처리내용>  While 반복문  While <불표현식>:  <처리내용>  <중간저장. P.190부터 재개>  #모듈  모듈 사용 : Import <모듈명>  모듈 종류 :  표준모듈: 파이썬에 기본적으로 지원하는 모듈  외부모듈: 다른 유저들이 만들어 공개한 모듈  모듈을 식별자로 선언도 가능 하다  Import <모듈명> as <사용자 지정 식별자>  내부 모듈 (자주 사용)  random  sys  os  datetime  time  urllib  외부모듈  설치(파이썬 내부에서)  !Pip install <사용하려는 외부 모듈> |