|  |  |
| --- | --- |
| 교육 제목 | **파이썬 2일차 교육** |
| 교육 일시 | 2021년 10월 29일 |
| 교육 장소 | 영우글로벌 러닝 |
| **교육 내용** | |
| 내용 | 2021.10.29 5일차     1. 함수    1. 함수는 영어로 function, 즉 사람 또는 사물의 기능이라는 뜻을 가진 단어와 동음이의어다. 지금까지 살펴본 숫자나 문자열과 같은 자료도 컴퓨터에서는 하나의 사물처럼 취급되기에 내부적으로 여러 기능을 가지고 있다.    2. 문자열 뒤에 마침표를 입력해 보면 자동완성기능으로 다양한 자체 기능들이 제시됨. 2. 문자열의 format() 함수    1. 중괄호 포함한 문자열 뒤에 마침표 찍고 format()함수 사용하되, 중괄호 개수와 format() 함수 안 매개변수의 개수는 반드시 같아야함.    2. 문자열의 중괄호 기호가 format()함수 괄호 안의 매개변수로 차례로 대치되면서 숫자가 문자열이 됨.    3. “{}”.format(10) -> “10” type = string   “{} {}”.format(10, 20) -> “10 20”   1. format() 함수의 다양한 기능    1. “{:d}.format = int 자료형 정수 출력한다는 것을 지정    2. {:5d}.format = 특정칸에 맞춰서 숫자를 출력하는 형태    3. {:05d}.format = 빈칸을 0으로 채우는 형태    5. {:+d} = 양수든 음수든 부호를 붙임.    6. {: d} = 양수를 부호를 붙이고, 음수를 부호를 안붙임.    8. 부동소수 점 출력 {:f}    9. 소수점 지정 : {:15.3f} 전체 표현 자릿수는 15, 소수점 미만 3자리    10. 의미없는 소수점 제거하기 {:g} 2. 대소문자 바꾸기 : upper()와 lower() / # 사용하는 문법 : [문자열].[함수()] 3. 문자열 양 옆의 공백 제거하기 : strip() / # 사용하는 문법 : [문자열].[함수()] 5. \*\*\*함수와 메소드의 차이    1. 함수 : 함수 이름을 통해 함수를 사용할 수 있다.    2. 함수 예) print(), type(), str(), int(), bool(),    3. 함수의 값을 변수에 대입할 수 있다.    4. 👉🏻 output = function\_name(input)    5. 메소드 : object(객체)와 연관되어 사용된다. → 사용하고자 하는 대상이 . 으로 연결되어있어야함    6. str,float,list와 같은 자료형은 모두 객체이므로 이러한 자료형과 연관되어 사용되는 것은 메소드로 볼 수 있다.    7. 메소드 예) .split() , .append() 등    8. 'class and its Object'의 개념.    9. 함수는 독립적으로 정의되므로 이름으로만 호출이 가능함.    10. 그러나 메소드는 이름으로만 호출되지 않음. 정의된 클래스의 참조에 의해 클래스를 호출해야한다.    11. 메소드는 클래스 내에서 정의되므로 해당 클래스에 종속된다.    12. 함수가 메소드보다 더 포괄적인 의미를 가진다.    13. **# 함수명() -> len(), print(), type()**    14. **# 오브젝트.함수명() -> 클래스 ( 맴버변수, 매서드(함수) ) -> str, upper,...** 6. 문자열의 구성 파악하기 : is00() 7. 문자열 찾기 : find()와 rfind()    1. find() : 왼쪽부터 찾아서 처음 등장하는 위치를 찾음    2. rfind() : 오른쪽부터 찾아서 처음 등장하는 위치를 찾음. 8. 문자열 자르기 : split()      1. 조건문    1. boolean :       1. True, False 값만 가질 수 있음.       2. 비교연산자를 통해 만들 수 있음.          1. == : 같다          2. != : 다르다          3. <, >, <=, >= : 크기비교          5. 문자열에도 비교연산자 적용가능함.             1. 영어 A ~ Z 인것처럼 글자에 대한 순서가 있으므로, 순서에 대한 크기비교연산이 가능하다. (<, >, <=, >=)          6. 불끼리 논리 연산자 사용가능 (not , and, or)       3. not 연산자       4. in 연산자 : 해당 문자열이 포함되어 있는지 확인함.          1. ex. “1” in “1, 2, 3, 4”  = True 2. IF 조건문    1. : 조건에 따라 코드를 실행하거나 실행하지 않게 할때 사용하는 구문    2. if 불 값이 나오는 표현식 : -> if의 조건문 뒤에는 반드시 콜론(:)을 붙여줘야하고, 그 이후 문장에서는 반드시 4칸 들여쓰기를 해야한다.    3. # 숫자인 경우 0이면 False / 문자열인 경우 none 이면 False    5. 날짜/시간 활용하기     예제  예제       1. **리스트란?**    1. **여러가지 자료를 저장할 수 있는 자료**    2. **대괄호 내부에 자료를 넣어 선언함.**       1. **eg. [ 273, 32, 103, ‘문자열’, True, False]** 2. **리스트 선언하고 요소에 접근하기**    1. **대괄호로 선언함. : [a, b, c, d]**    2. **인덱스 : 왼쪽부터 0, 1, 2, 3, 4 … 이된다.**    3. **리스트옆에 대괄호를 씌어 요소에 접근한다.**       1. **ex. list[0] : 첫번째 요소 접근, list[1:3] : 리스트 두번째부터 세번째 요소 접근**       2. **리스트안에 여러 리스트가 있을 경우**          1. **List = [ [1, 2, 3,], 2, 3 ]**          2. **List[0][1] = 2**       3. **리스트에서의 IndexError 예외**          1. **리스트의 길이 넘는 인덱스로 요소에 접근하려는 경우 발생함** 3. **리스트 연산자**    1. **+ : 리스트를 연결하는 연산**    2. **\* : 숫자만큼 반복하여 연결하는 연산**    3. **len()  : 리스트의 데이터 개수** 4. **리스트에 요소 추가 / 제거하기**    1. **.append() 함수 : 리스트뒤에 요소로서 추가**       1. **ex. list\_a.append(‘요소’) : list\_a 에 ‘요소’가 추가됨.**    2. **.insert() 함수 : 리스트 중간에 요소로서 추가**       1. **ex. list\_a.insert(위치,요소) : list\_a의 위치에서 요소가 추가됨.**    3. **.extend() 함수 : 리스트와 리스트를 연결** |
| 주요 명령어 | * **“~ {} ~”.format(“a”) -> a는 첫번째 괄호에 대응됨.** * **{:d}.format = int 자료형 정수 출력한다는 것을 지정** * **{:5d}.format = 특정칸에 맞춰서 숫자를 출력하는 형태** * **{:05d}.format = 빈칸을 0으로 채우는 형태** * **{:d}.format = int 자료형 정수 출력한다는 것을 지정** * **{:5d}.format = 특정칸에 맞춰서 숫자를 출력하는 형태** * **{:05d}.format = 빈칸을 0으로 채우는 형태** * **{:d}.format = int 자료형 정수 출력한다는 것을 지정** * **{:5d}.format = 특정칸에 맞춰서 숫자를 출력하는 형태** * **{:05d}.format = 빈칸을 0으로 채우는 형태** * **# 함수명() -> len(), print(), type()** * **# 오브젝트.함수명() -> 클래스 ( 맴버변수, 매서드(함수) ) -> str, upper,...** * find() : 왼쪽부터 찾아서 처음 등장하는 위치를 찾음 * rfind() : 오른쪽부터 찾아서 처음 등장하는 위치를 찾음 * 문자열 자르기 : split(). * **.append() 함수 : 리스트뒤에 요소로서 추가** * **ex. list\_a.append(‘요소’) : list\_a 에 ‘요소’가 추가됨.** * **.insert() 함수 : 리스트 중간에 요소로서 추가** * **ex. list\_a.insert(위치,요소) : list\_a의 위치에서 요소가 추가됨.** * **.extend() 함수 : 리스트와 리스트를 연결** |