

2주차 수업

단국대학교 알고리즘 튜터링

<https://github.com/m1nnh/DKU-Algorithm-Tutoring>

파이썬 이란?

- 1990년 암스테르담의 귀도 반 로섬이 개발한 인터프리터 언어이다.
- 파이썬은 컴퓨터 프로그래밍 교육을 위해 많이 사용하지만, 기업의 실무를 위해서도 많이 사용하는 언어이다.
- 또한 파이썬 프로그램은 공동 작업과 유지 보수가 매우 쉽고 편하다.



파이썬의 특징

- 파이썬은 인간다운 언어이다.
Ex) if 4 in [1, 2, 3, 4] : print('4가 있습니다.')
- 파이썬은 문법이 쉬워 빠르게 배울 수 있다.
- 파이썬은 간결하다.
- 파이썬은 개발 속도가 빠르다.

파이썬으로 할 수 있는 일

- 시스템 유틸리티 제작
- GUI 프로그래밍
- C/C++와의 결합
- 웹 프로그래밍
- 수치 연산 프로그래밍
- 데이터베이스 프로그래밍
- 데이터 분석, 사물 인터넷
- 시스템과 밀접한 프로그래밍영역, 모바일 프로그래밍은 사용할 수 없다.

숫자 자료형

- 숫자형 (정수형, 실수형, 8진수와 16진수)
- 사칙 연산의 종류 ($A = 5$, $B = 3$)

+	$A + B \rightarrow \text{result} = 8$
-	$A - B \rightarrow \text{result} = 2$
*	$A * B \rightarrow \text{result} = 15$
// or /	$A // B \rightarrow \text{result} = 1$, $A / B \rightarrow \text{result} = 1.666$
%	$A \% B \rightarrow \text{result} = 2$
**	$A ** 2 \rightarrow \text{result} = 25$, $B ** 2 \rightarrow \text{result} = 9$

문자열 자료형

- 문자열은 작은 따옴표, 혹은 큰 따옴표로 둘러 쌓인 문자의 집합들이다. (문자와 문자열은 엄연히 틀림)
- 문자열 안에 작은 따옴표나, 큰 따옴표를 포함시키고 싶을 때는 (""" """) or ("' '") 로 감싸주면 된다.

Ex) `print("""Python's favorite food is perl""")`

- 백슬래시를 사용해서 작은 따옴표나 큰 따옴표도 포함 시킬 수 있다.

Ex) `print('Python\'s favorite food is perl')`

- 문자열 연산(A = Park, B = MinHyeok)

+	A + B -> result = 'ParkMinHyeok'
*	A * 2 -> result = 'ParkPark'
len	len(A) -> result = 4

인덱싱과 슬라이싱

- 인덱싱이란 무엇인가를 '가리킨다'는 의미이고, 슬라이싱은 무엇인가를 '잘라낸다'는 의미이다.
- 파이썬에서 인덱스의 시작은 0번부터 시작한다.
- `A = 'ParkMinHyeok'`
Ex) `A[2] = 'r'`, `A[6] = 'n'`, `A[-1] = 'k'`
- 한번에 단어나 원하는 인덱스 까지만 뽑아내는 기술을 슬라이싱이라 한다.
- `A = 'Dankook University'`
Ex) `A[:] = 'Dankook University'`, `A[3:7] = 'kook'`, `A[::-1] = 'ytisrevinU kooknaD'`
- 슬라이싱의 중요한 점은 `A[start:end]`가 있으면 `start`는 포함을 시키고 `end`는 포함을 시키지 않는다.
Ex) `A[2:5] -> index : 2 <= index < index : 5`

문자열 관련 함수

- `count()` : 문자 개수 세기

Ex) `A = 'hobby' A.count('b') -> result = 2`

- `find()` : 위치 알려 주기

Ex) `A.find('b') -> result = 2` ('b'가 처음 나온 위치의 인덱스를 반환)

- `Index()` : 위치 알려 주기

Ex) `A.index('b') -> result = 2` (위에 설명과 동일)

- `join()` : 문자열 삽입

Ex) `>>> ','.join('abcd') -> result = 'a, b, c, d'` ('abcd' ,를 기준으로 'abcd' 문자열 삽입)

- `upper()` : 소문자를 대문자로 변환, `lower()` : 대문자를 소문자로 변환

Ex) `A.upper() -> result = 'HOBBY'`

- `split()` : 문자열 나누기

리스트 자료형

- 리스트 = 배열 []
- 리스트 사칙연산은 앞에 설명한 것과 같음
- `append()` : 리스트에 요소 추가
Ex) `A = [1, 2, 3, 4]` `A.append(5)` -> result `A = [1, 2, 3, 4, 5]`
- `sort()`, `sorted()` : 리스트 정렬
Ex) `A = [3, 6, 9, 1]` `A.sort()` -> result `A = [1, 3, 6, 9]`, `B = sorted(A)` result `B = [1, 3, 6, 9]`
- `Reverse()` : 리스트 뒤집기 (이것 보다 슬라이싱에서 배운 `[::-1]`을 더 자주 이용)
- `index()` : 위치 반환
Ex) `A=[1, 5, 6, 3]` `A.index(6)` -> result = 2

튜플 자료형

- 리스트는 []으로 둘러 싸지만 튜플은 ()로 둘러싼다.
- 리스트와 달리 튜플은 값 변경이 불가능하다.

딕셔너리 자료형

- 딕셔너리는 key : value 가 한쌍으로 이루어진 자료형이다. 이를 연관 배열 또는 해시라고 한다.
- {key1 : value1, key2 : value2, key3 : value3, ...} 기본적인 딕셔너리 형태이다.
- 딕셔너리 값 추가하기
Ex) A = {1 : 'a'} A[1] = 'b' result A = {1 : 'a', 2 : 'b'}
- A.keys() -> 딕셔너리의 키 값만 출력
- A.values() -> 딕셔너리의 밸류 값만 출력
- A.items() -> 딕셔너리 키, 밸류 값 모두 출력

집합 자료형

- 집합 자료형은 set 키워드를 사용한다.
- 집합 자료형은 중복을 허용하지 않는다.
- 집합 자료형은 순서가 없다.
- add() : 값 1개 추가
Ex) A = set() A.add(1) result A = {1}
- update() : 값 여러개 추가
Ex) A.update([4, 5, 6, 6]) result A = {4, 5, 6}
- remove() : 특정 값 제거
Ex) A.remove(4) result A = {5, 6}

사용자 입력과 출력

- `input()` : 사용자 입력
- `print()` : 출력 (문자열 띄어쓰기는 콤마로 하고, 끝 부분에 `end` 속성값을 줄 수 있음)
- `map()` : 여러 개의 문자를 입력 받을 때 사용
Ex) `A, B = map(int, input().split())`
- `map()`을 이용하여 리스트도 한 줄에 입력 가능
Ex) `A= list(map(int, input().split()))`

if 문

- if문의 기본 구조
- if 조건문 :
 수행할 문장
- ...
- elif 조건문:
 수행할 문장
- ...
- else:
 수행할 문장
- ...
- 들여쓰기 필수!!

조건문 비교 연산자

- 비교 연산자

$X < Y$	X 가 Y보다 작다
$X > Y$	X 가 Y보다 크다
$X == Y$	X와 Y가 같다
$X != Y$	X와 Y가 같지 않다
$X \geq Y$	X가 Y보다 크거나 같다
$X \leq Y$	X가 Y보다 작거나 같다

조건문

- 파이썬 특징 연산자

X or Y	X와 Y 둘 중에 하나만 참이어도 참
X and Y	X와 Y 둘다 참이어야 참
not X	X가 거짓이면 참
X in Y	(Y는 집합) X가 Y안에 있으면 참
X not in Y	(Y는 집합) X가 Y안에 없으면 참

while 문

- while 조건: (조건이 참이면 반복문 수행)
수행할 문장
수행할 문장
...
- while문을 강제로 빠져 나갈 때 break문을 사용하면 됨

for 문

- while문과 같이 똑같은 반복문
- for 변수 in (리스트, 튜플, 문자열):
 수행할 문장
 수행할 문장
 ...
- for문과 함께 자주 사용하는 range 함수가 있다.
Ex) range(start, end, jump) start는 시작 숫자, end 끝 숫자
- range함수의 주의 사항은 **end는 포함하지 않는다.**

함수

- def 함수이름(매개 변수):
 수행할 문장
 수행할 문장
 ...
- 함수는 코드의 가독성을 높여준다.
- 함수는 코드의 재사용성을 높여준다.
- 함수의 프로세스 과정
- 입력값 -> 함수 -> 결과값
- 함수의 결과값은 언제나 하나임!!