

\n : 탭간격

\\ : \ 표현

\' : ' 표현

\": " 표현

★ 파이썬은 문자열을 더하거나 곱하기 가능 ★

인덱싱: 무엇을 가리킨다 → 0부터 시작

★ 문자열 뒤에서부터 뒤를 때는 음수 사용 → -1 부터 시작

단어 뽑으려면 한글자씩 뽑아서 더하기
or

슬라이싱 ★ 슬라이싱 [] 대괄호 사용

★ 끝번호는 슬라이싱 하고 싶은 인덱스 + 1 잊지말기

끝번호 생략시 문자열 끝까지 출력

인덱싱처럼 음수 사용가능 but, [-7] 뺀다면 [-6] 써야함

슬라이싱 사용시 문자열 변경도 가능

★ 문자열 포매팅 ★

%d 넣고, 뒤에 %숫자
↙ 변수대입도 가능

↳ 문자열 포맷코드

%s 넣고, 뒤에 % 'string'

↳ 괄호일때 2개 이상 갖 가능

%d 정수

%s 문자열

%c 문자대

%f float type

‘%’ 넣으려면 %% ← 두번써야! (포매팅의 경우)



"%.10s" % 'hi'

→ 8칸의 공백과 hi (오른쪽정렬)

"% -10s" % 'hi'

→ hi와 8칸의 공백 (왼쪽정렬)

* 소숫점 표시 "%.4f" % 3.141592

→ 소숫점 4번째까지만 표시

"%.10,4f" % 3.141592

→ 앞에 4글자 공백 + 뒤에 소숫점 4자리까지 표현

⇒ format 함수 사용가능

"I eat {} apples".format(3)

→ {} 부분이 3으로 바뀜

↙ 문자열,
변수,
2개이상값
모두가능

"I ate {} number {} apples. so I was sick
for {} day {} days.".format(number=10, day=3)

⇒ 가능!

인덱스와 이름 혼용도 가능

오른쪽정렬 "{0:>10}".format("hi")

"{0:<10}".format("hi") → 왼쪽

* 가운데

"{0:^10}".format("hi") → 가운데 정렬

"{0:=^10}".format("hi") → == == hi == ==

"{0:!!<10}".format("hi") → hi!!!!!!!



소식표현

$$y = 3.141592$$

"{0:0.4f}.format(y) → 3.1415

"%0:10.4f".format(y) → 3.1415
4개공백

문자표현

"{{ and }}" format() → '{ and }'

↑ 연속사용가능

★ f 문자열 포매팅

name = 'ကျွန်း'

age = 27

f'나의 이름은 {name} 입니다. 나이는 {age} 입니다.'

→ 면소현 $\begin{matrix} \nearrow 27 \\ \nwarrow \end{matrix}$ \Rightarrow 총력
면산도가능

독서너리 이용도 가능

$d = \{ \text{'name': '홍길동', 'age': 30} \}$

f' 도의 이름은 `{d["name"]}`입니다. 나이는 `{d["age"]}`입니다.

: < 왼쪽 정렬 : > 오른쪽 정렬 : ^ 가운데 정렬

$\therefore = 1$ 가운데 정렬후 \Rightarrow 로 공백채우기

! < 왼쪽 정렬 후 !로 공백채우기

* 문자열 앞 ← 부호 전체 길이 두

Count → 문자개수세기

find → 문자 처음나온 인덱스 반환
(없으면 -1 반환)

index → find와 기능동일
but, 문자없으면 오류발생

join * → " ".join('abcd')
→ a, b, c, d] 문자열 삽입가능

리스트, 튜플에서도 이용가능

upper() 소문자 → 대문자) → 실행하면 변수 자체가
lower() 대문자 → 소문자 바뀌는거 x ∴ 대입 다시 해줘야

lstrip() 왼 공백지우기

rstrip() 오 공백지우기

strip() 양쪽 공백지우기

replace(바꿀문자열, 바꿀문자열) ← 치환가능

split() ← 공백기준 문자열 분리

split(':') ← : 기준 문자열 분리

* 리스트는 [] 로 표현, , 로 요소구분

↑ 비어있는 리스트도 가능

리스트 ← 어떠한 자료형도 포함가능

리스트 안 리스트 사용시에는 a[3][0] (다중중첩가능)

↑ 활용가능



: 숫자 \rightarrow 처음부터 숫자-1 까지

숫자: \rightarrow 숫자부터 끝까지

리스트 더하기, 곱하기 (반복) 가능

$\text{len}(\text{리스트}) \rightarrow$ 길이 구하기 가능

값 수정 \rightarrow 해당 인덱스 = 새로운 값 대입

del 이용 $\text{del } a[i] \rightarrow$ 인덱스 i 요소 삭제
 \uparrow 슬라이싱 이용 가능

a.append(숫자, 문자...) \rightarrow 리스트 맨 마지막에 () 추가

sort(): 숫자, 알파벳 정렬

reverse(): 리스트 고대로 뒤집기

a.index(리스트 요소): 리스트 요소의 인덱스 값 반환

\rightarrow 존재 X 이면 오류 발생

a.insert(인덱스, 리스트 요소)

\rightarrow 해당 인덱스에 리스트 요소 삽입 가능

a.remove(리스트 요소): 첫번째 나오는 리스트 요소 삭제

a.pop() \rightarrow 맨 마지막 요소 빼내기

(기존 리스트에서 해당 요소는 삭제)

\rightarrow 인덱스 불문 해당 요소 빼내고 삭제

a.count(리스트 요소) \rightarrow 개수 세기

a.extend(새 리스트) \rightarrow 원래의 리스트에 새 리스트 더함



* 튜플 ()로 정의

↳ 요소값 변경불가, 삭제 del로 불가능

튜플은 요소 개수 (1,) ^{← 컴파일러가 알아야 함}

tuple1 = 1, 2, 3 ← 튜플로 정의됨

* 인덱싱, 슬라이싱, 더하기, 곱하기 모두 리스트와 동일

* 딕셔너리

{key1: Value1, key2: Value2, key3: Value3}

↑ 리스트 넣기 가능

추가 d[2] = 'Value'

del [key] ⇒ key와 value 한 쌍이 모두 삭제

* key 사용으로 value 사용이 편리.

↳ 중복사용시 무시되므로 주의*

리스트는 key 값으로 사용불가 ∴ key는 변하면 X는 값!

리스트의 append, pop, insert, remove, sort 수행불가

keys → 각개체 리스트로 변환가능

Values → "

keys + values 두개 같이 쌍으로 얻으려면 items 사용

clear() → 딕셔너리 내부 key, value 모두 삭제

get(key) → value값 반환

↳ 없으면 None 반환