

Chapter1. 파이썬 용어들

Keyword : 파이썬이 만들어질때 이미 사용하겠다고 예약해 놓은 것, 이를 구분해야 하는 이유는 사용자가 이름을 정할때 키워드를 사용해서는 안되기 때문이다.

Identifier(식별자) : 변수 또는 함수의 이름을 붙일때 사용하는 단어, 몇가지 규칙이 존재한다.

1. 키워드를 사용해선 안된다.
2. 특수문자는 _만 허용된다.
3. 숫자로 시작하면 안된다.
4. 공백을 포함할 수 없다.

Snake_case : 단어사이에 _를 붙여 식별자를 만든다. ex) item_list

Camelcase : 단어들의 첫 글자를 대문자로 만들어 식별자를 만든다. ex) ItemList

식별자구분하기 : 첫번째 글자가 소문자라면 무조건 스네이크케이스, 대문자면 캐멀케이스이다.

캐멀케이스 = 클래스

스네이크케이스 = (뒤에괄호가없으면) 변수

스네이크케이스 = (뒤에괄호가있으면) 함수

Chapter2. 자료형

2-1 자료형과 문자열

자료형확인하기

- type() 을 사용하면 문자의 자료형을 확인할수있다.

SyntaxError: invalid syntax - 구문오류

escape character(이스케이프 문자)

- \와 함께 조합해서 사용하는 특수문자를 의미합니다.

\": 큰따옴표를 의미합니다.

\': 작은따옴표를 의미합니다.

\n : 줄바꿈을 의미합니다.

\t : 탭을 의미합니다.

\\ : 역슬래시\를 의미합니다.

문자열 연산자

+ : ex) print("안녕" + "하세요")

>>> 안녕하세요

이때 문자열은 문자열끼리 숫자는 숫자끼리 해야한다 그렇지 않으면 다음의 에러가 나타난다.

TypeError: can only concatenate str (not "int") to str

* : 문자열을 숫자와 * 연산자로 연결하면 문자열을 반복할 수 있습니다.

ex) print("안녕하세요" * 3)

>> 안녕하세요안녕하세요안녕하세요

[] : 문자열 내부의 문자하나를 선택하는 연산자입니다.

[:] : 문자열의 특정 범위를 선택할 때 사용하는 연산자도 있습니다.

```
ex)print("안녕하세요"[1:4])
```

```
>>녕하세
```

```
print("안녕하세요"[1:])
```

```
>>녕하세요
```

[]를 이용해 문자열의 특정 위치에 있는 문자를 참조하는것을 **인덱싱**이라고, [:]를 이용해 문자열의 일부를 추출하는것을 **슬라이싱**이라고 합니다.

IndexError(index out of range)

-인덱스에러는 리스트/문자열의 수를 넘는 요소/글자를 선택할 때 발생합니다. ex)print("안녕하세요"[10])

len() : 문자열의 길이를 구하는 함수

```
ex) print(len("안녕하세요"))
```

```
>>>5
```

2-2 숫자

숫자에는 두가지 종류가 존재합니다

정수 : integer 실수 : floating point

사칙 연산자 : + - * /

정수나누기 연산자(/) : 정수부분만

남김

```
ex)print(3/2)
```

```
>>>1.5
```

```
print(3//2)
```

```
>>>1
```

나머지 연산자(%) : 나머지를 구함

```
ex)print(5 % 2)
```

```
>>>1
```

제곱 연산자()** : 숫자를 제곱

```
ex)print(2 ** 3)
```

```
>>>8
```

2-3 변수와 입력

변수만들기 : 변수=값

변수 안에 있는 값을 사용하는것을 '변수 참조'라고 합니다

복합 대입 연산자

+= : 숫자 덧셈 후 대입

-= : 숫자 뺄셈 후 대입

***=** : 숫자 곱셈 후 대입

/= : 숫자 나눗셈 후 대입

%= : 숫자의 나머지를 구한 후 대입

****=** : 숫자 제곱 후 대입

```
ex) number = 100
```

```
number += 10
```

```
number += 20
```

```
print("number:", number)
```

```
>>>130
```

문자열 복합 대입 연산자

+= : 문자열 연결 후 대입

***=** : 문자열 반복 후 대입

```
ex)string = "안녕하세요"
```

```
string += "!"
```

```
print(string)
```

```
>>>안녕하세요!
```

사용자 입력 : input()

-사용자로부터 데이터를 입력받을때 사용 ex)

```
string = input("인사말을 입력하세요>")
```

```
인사말을 입력하세요> 안녕하세요
```

```
print(string)
```

```
>>>안녕하세요
```

input과 같이 함수의 결과로 나오는 값을 리턴값이라고합니다

!! input()함수는 사용자가 무엇을 입력해도 결과는 무조건 문자열 자료형입니다

ex)

```
number = input("숫자를 입력하세요>")
```

```
숫자를 입력하세요> 12345
```

```
print(type(number))
```

```
<class 'str'>
```

문자열을 숫자로 바꾸기

int() 함수 : 문자열을 int자료형으로 변환

float() 함수 : 문자열을 float자료형으로

ex)

```
string_a = input("입력A>")
```

```
int_a = int(string_a)
```

```
string_b = input("입력B>")
```

```
int_b = int(string_b)
```

```
print(string_a + string_b)
```

```
pring(int_a + int_b)
```

```
>>입력A> 273
```

```
>>입력B> 52
```

```
>>27352
```

```
>>325
```

ValueError 예외

:숫자가 아닌 것을 숫자로 변환하려고 할때

,실수를 정수로 변환하라고 할때

ex)int("안녕하세요")

숫자를 문자열로 바꾸기

```
str(숫자)
```

2-4 숫자와 문자열의 다양한 기능

문자열의 format() 함수

-숫자를 문자열로 변환합니다

-중괄호의 개수와 매개변수 개수는

일치시켜야합니다

ex)

```
string_a = "{}".format(10)
```

```
print(string_a)
```

```
print(type(string_a))
```

```
>>10
```

```
>><class 'str'>
```

{ }가 매개변수보다 많은 경우
indexError예외가 발생합니다

format() 함수의 다양한 기능

ex)

#정수로 출력

```
output_a = "{:d}".format(52)
```

```
>>52
```

#5칸띄우고 출력

```
output_a = "{:5d}".format(52)
```

```
>> 52
```

#빈칸을 0으로 채우기

```
output_c = "{:05d}".format(52)
```

```
>>0000052
```

#부호를 표시

```
output_d = "{:+d}".format(52)
```

```
>>+52
```

#양수는 공백,음수는-표시

```
output_e = "{: d}".format(52)
```

```
>>52
```

#부호를 앞으로 밀기

```
output_f = "{:=+5d}".format(52)
```

```
>>+ 52
```

```
input_a.stdout_a =
"{:f}".format(52.273)
>>52.273
-정수형출력때와 기능이 같음
#소수점 아래 자릿수 지정하기
output_a = "{:15.3f}".format(52.273)
>>          52.273
output_b = "{:15.2f}".format(52.273)
>>          52.27
#의미없는 0제거
output_a = 52.0
output_b = "{:g}".format(output_a)
>>52
```

대소문자 바꾸기: upper() 와 lower()

```
ex)
a = "Hello"
a.upper()
>>'HELLO'
a.lower()
>>'hello'
```

문자열 양옆의 공백 제거하기: strip()

```
ex)
input_a = "   "
    안녕하세요
문자열의 함수를 알아봅시다
"   "

print(input_a)
>>
    안녕하세요
문자열의 함수를 알아봅시다

print(input_a.strip())
>>안녕하세요
문자열의 함수를 알아봅시다
```

문자열 찾기: find() 와 rfind()

```
find() : 왼쪽부터 찾기
rfind() : 오른쪽 부터 찾기
ex)
output_a = "안녕안녕하세요".find("안녕")
print(output_a)
>>0
output_b = "안녕안녕하세요".rfind("안녕")
print(output_b)
>>2
```

문자열과 in 연산자

```
문자열 내부에 어떤 문자열이 있는지 확
인하려면 in 연산자를 사용합니다
ex)
print("안녕" in "안녕하세요")
>>True
print("잘자" in "안녕하세요")
>>False
```

문자열 자르기 : split()

```
문자열을 특정한 문자로 자를때 사용
ex)
a = "10 20 30 40 50".split(" ")
print(a)
>>['10', '20', '30', '40', '50']
실행결과로 리스트가 나옵니다
```