

## [Python] 파이썬 기초 1 - 변수(variable)에 대한 타입과 기본적인 함수 사용법

노트북: [TIL-MY]

만든 날짜: 2020-08-11 오후 9:48

URL: <https://continuous-development.tistory.com/61?category=736681>

# 나무늘보의 개발 블로그

홈

태그

Python

## [Python] 파이썬 기초 1 - 변수(variable)에 대한 타입과 기본적인 함수 사용법

· by 꾸까꾸 · 2020. 8. 10. · 수정 · 삭제

파이썬은 변수가 할당 될때 지정 된다.

Python Built-in Types

-Numeric

-Sequence

-Text Sequence

분류 전체보기 

Python 

Database

ASP.NET

Algorithm

Machine learning  
| Deep lear..

- Set
- Mapping(dict, tuple)
- Bool

## 변수 지정 방법

- Caeml Case : numberOfCollege -> method, function
- Pascal Case : NumberOfCollege -> class
- Snake Case : number\_of\_college -> method, function

# 파이썬이 가지고 있는 내장변수들이다. 이것들을 변수의 이름이나 클래스의 이름으로 지정해줄수 없다.

```
import keyword

# 파이썬이 가지고 있는 내장변수들이다. 이것들을 변수의 이름이나 클래스의 이름으로 지정해줄수 없다.
print(keyword.kwlist) # ['False', 'None', 'True', 'and'....]
print()
```

AWS

ETC..

R 

공지사항

글 보실 때 주의사항

: 최근글 : 인기글

[Python] 파이썬 기초 7 ....

2020.08.11

[Python] 파이썬 기초 6 ....

2020.08.11

[Python] 파이썬 기초 5 ....

2020.08.11

[Python] 파이썬 기초 4 ....

2020.08.11

[Python] 파이썬 기초 3 ....

2020.08.10

최근댓글

태그

python if문,  
파이썬 합집합,  
날짜함수,  
파이썬 차집합,  
DDL, 파이썬,

# 변수는 숫자로 시작할 수 없고 특수문자는 \_, \$만 허용한다.

```
# 변수는 숫자로 시작할 수 없고 특수문자는
_, $만 허용한다.

year = 2019
month = 8
day = 10
print('{}년{}월{}일'.format(year, month, day)) # 2019년8월10일
print(type(year), type(month), type(day))
# <class 'int'> <class 'int'> <class 'int'>

intValue = 123
floatValue = 3.14159
boolValue = True
strValue = 'jslim'

print(type(intValue), type(floatValue), type(boolValue), type(strValue)) # <class 'int'> <class 'float'> <class 'bool'> <class 'str'>
```

## # 형변환 type casting

```
# 형변환 type casting

num_str = "720"
num_num = 100

print(int(num_str) + num_num) # 820
print(num_str + str(num_num)) # 720100

year = "2020"

print(int(year)-1) # 2019
```

## # 데이터 타입

python if elif else,  
SQL,  
테이블 생성,  
설정, ggplot,  
python date type,  
AWS,  
Oracle SQL,  
파이썬 교집합,  
파이썬 dictionary,  
substr, 인스턴스,  
cbind, 사용법,  
rbind,  
python 데이트 타입,  
입,  
파이썬 if문,  
파이썬 함수,  
파이썬 date,  
Python, Oracle,  
파이썬 datatype,  
행렬,  
python if else

---

전체 방문자

# 167

Today : 5

Yesterday : 5

```
str01 = "python"
boolValue = True | False
str02 = 'Anaconda'
floatValue = 10.0
intValue = 20
```

# **list**는 배열

```
listValue = [str01, str02]
```

# **dict** 는 key와 value 형태로 데이터를 담는다.

```
dictValue = {"name" : "machine Learning",
"version" : 2.0
}
```

# 벡터 같이 데이터를 1차원 형태로 만들 수 있다. 객체의 집합으로서 한번 생성되면 변경 할 수 없다.

```
tupleValue = (3, 5, 7)
```

# 집합을 나타낸다.

```
setValue = {3, 5, 7}

print(type(list))
print(type(dict))
print(type(tuple))
print(type(set))
```

# 키보드 입력

# input()

```
inputNumber = int(input("숫자를 입력하세요: "))
sumValue = inputNumber + 100
print(sumValue)
```

# 파이썬 문자형 (중요)

```
str01 = "I am Python"
str02 = 'python'
str03 = """this is a
multiline
sample text"""

print(str01, str02, str03) # I am Python python this is a multiline sample text
query = "select * from emp" W
"where deptno = {no}"W
"order by eno desc"

print(query) # select * from emp where deptno = {no} order by eno desc

seqText = "Talk is cheap. Show me the code"
print(seqText) # Talk is cheap. Show me the code
```

# 객체지향이라는 것이 하나의 모듈에 필요한 함수나 변수를 가지고 다니는 것을 말한다. 이것을 클래스라고 한다.

# **dir()** - 내장함수를 나타낸다.

```
print(dir(seqText)) # ['__add__', '__class__  
__', '__contains__', '__delattr__',....]
```

# **slicing**

```
print(seqText[3]) # k  
print(seqText[-1]) # e  
print(seqText[0:3]) # Tal - 뒤에 인덱스 -1  
이다.
```

```
str_slicing = "Nice Python"  
print(str_slicing[0:4]) # Nice  
print(str_slicing[5:]) # Python  
print(str_slicing[5:11]) # Python  
print(str_slicing[:]) # Nice Python  
print(str_slicing[::2]) # Nc yhn / step을  
통해 원하는 순서의 값을 가져올수 있다.  
print(str_slicing[::3]) # Noyo  
print(str_slicing[0:len(str_slicing):2]) #  
Nc yhn  
print(str_slicing[-6:]) # Python  
print(str_slicing[::-1]) # nohtyP ecin / r  
everse 형식으로 된다.
```

# **인덱스 사용법**

# 아래의 문자열 에서 '홀' 만 출력하세요

```
string = "홀짝홀짝홀짝홀짝"  
print(string[::2]) # 홀홀홀홀
```

# 아래의 문자열을 거꾸로 뒤집어서 출력하세요

```
string = "PYTHON"  
print(string[::-1]) # NOHTYP  
strint = "python"
```

# **capitalize()** - 첫 문자를 대문자로 만든다.

```
string = "python"  
print("Capitalize : ", string.capitalize()  
()) # Capitalize : Python / 첫문자를 대문  
자로 만든다.
```

# **replace** - 해당 문자를 치환한다.

```
phone_number = "010-4603-2283"  
print(phone_number.replace("-", "")) # 010  
46032283
```

# 아래 문자열에서 소문자 a 를 대문자로 변경 한다  
면?

```
string = 'abcdefe2a346a345a'

print(string.replace("a", "A")) # Abcdefe2
A346A345A
```

# **split()** - 해당 문자를 기준으로 자른다.

# 아래 문자열에서 도메인만 출력한다면?

```
url = "http://naver.com"
url_split = url.split('.')
print(url_split[-1]) # com
```

# **strip()** , **rstrip()**, **lstrip()** - 공백제거

```
data = " 삼성전자 "
print(data.strip()) # 삼성전자 / 좌우 공백
제거
print(data.rstrip()) # 삼성전자 /오른쪽 공
백 제거
print(data.lstrip()) # 삼성전자 /왼쪽 공백
제거
```

# **upper()** - 대문자로 나타낸다.

# **lower()** - 소문자로 나타낸다.

```
ticker = "Btc_krw"

print(ticker.upper()) # BTC_KRW
print(ticker.lower()) # btc_krw
```



# **endswith()** - 값이 있는지를 true false로 나타낸다.

```
file_name = "report.xls"
isExists= file_name.endswith(("xls", "xlsx"))
print(isExists) # True
```

# **split()** - 지정한 문자를 경계로 나눈다.

```
string = "삼성전자 / LG전자 / Naver / Google / kakao"
interest = string.split("/")
print(interest) # ['삼성전자 ', ' LG전자 ', ' Naver ', ' Google ', ' kakao']
```

# **in, not in** -> True | False - 문자열을 가지고 값이 있는지 비교한다 - 시퀀스 개념의 text이기 때문이다.

```
myStr = "This is a sample Text"
print("sample" in myStr) # True
print("Text" not in myStr) # False
print("this" in myStr.lower()) # True
```

# **len()** - 해당 문자열의 길이 , **count()** - 해당 문자열

을 집계 ,

# **find()** 해당 문자열의 인덱스 값을 반환 (없을 경우  
-1 리턴)

# **index()** 처음 나오는 문자열을 return한다. (없을  
경우 에러는 반환한다.)

```
brand_name= "cocacola"  
result = len(brand_name) # 해당 문자열의  
    길이를 출력한다.  
print(result) # 8  
result = brand_name.count('c') # 해당 문자  
    열을 집계해서 나타낸다.  
print(result) # 3  
result = brand_name.find('o')  
print(result) # 1  
result = brand_name.find('z')  
print(result) # -1  
result = brand_name.index('a')  
print(result) # 3  
#result = brand_name.index('f')  
print(result) # 3
```

# **ord** - 해당 문자의 아스키 코드를 출력한다 / chr(n  
um) - 해당 숫자 값의 아스키 코드 값을 반환한다.

```
a = 'a'  
print(a) # a  
a = 'A'  
print(ord(a)) # 65  
print(chr(65)) # A
```

## 'Python' 카테고리의 다른 글

[Python] 파이썬 기초 3 - 튜플(tuple)에 대한 정의와 기본적인 함수 사용법 (0)	2020.08.10
[Python] 파이썬 기초 2 - 리스트(list)에 대한 정의와 기본적인 함수 사용법 (0)	2020.08.10
<u>[Python] 파이썬 기초 1 - 변수(variable)에 대한 타입과 기본적인 함수 사용법 (0)</u>	2020.08.10
[Python] python 기본 적인 print 문 (0)	2020.08.10
[Python] python 아나콘다 설치 및 가상환경 설정 하는 방법 (0)	2020.08.10
AWS내에서 Django 가상환경 구성하기 (0)	2019.11.19

## 태그

python variable

파이썬 기초

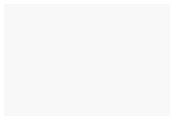
파이썬 변수

파이썬 타입

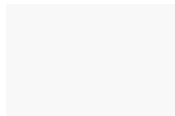
파이썬 함수

파이썬 형변환

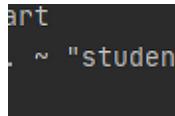
## 관련글



[Python] ...



[Python] ...



[Python] p...



[Python] p...

## 댓글 0

TEL. 02.1234.5678 / 경기 성남시 분당구 판교역로

© Kakao Corp.

