

파이썬의 식별자와 변수



1 파이썬의 식별자

① 파이썬의 식별자란?

식별자

변수, 함수, 모듈, 클래스 등 객체를 식별하는데 사용되는 이름

파이썬의 식별자와 변수

1 파이썬의 식별자

2 식별자의 조건

1 대·소문자를 구별함

```
A = 123
print(A)
```

123

```
a = 456
print(a)
```

456

2 식별자의 첫 글자는 문자 또는 언더스코어만 사용할 수 있음

- 식별자의 중간에는 숫자가 올 수 있음

```
a1 = 999
print(a1)
```

999

```
1a = 999
print(1a)
```

```
File "<ipython>" line 1, in <module>
    1a = 999
    ^
```

SyntaxError:

```
_a = 999
print(_a)
```

999

파이썬의 식별자와 변수

1 파이썬의 식별자

2 식별자의 조건

3

언더스코어를 제외한 특수문자 및 공백은 사용할 수 없음

```
a_ = 2019
print(a_)
```

2019

```
a& = 2019
print(a&)
```

```
File "<ipytho
a& = 2019
      ^
```

SyntaxError:

```
a* = 2019
print(a*)
```

```
File "<ipytho
a* = 2019
      ^
```

SyntaxError:

```
a@ = 2019
print(a@)
```

```
File "<ipytho
a@ = 2019
      ^
```

SyntaxError:



파이썬의 식별자와 변수

1 파이썬의 식별자

2 식별자의 조건

4 예약어는 사용할 수 없음

- 대·소문자를 구별하기 때문에 소문자 true는 변수로 사용 가능

```
True = 1  
print(True)
```

```
File "<ipython-input-46-1afe7eee66f0>", line 1
```

```
True = 1
```

```
^
```

```
SyntaxError: can't assign to keyword
```

파이썬의 식별자와 변수

2 파이썬의 변수

1 변수의 생성과 사용



파이썬은 변수를 생성하지 않고 바로 사용할 수 없음

```
a
```

```
-----  
NameError
```

```
<ipython-input-49-3f786850e387> in <
```

```
----> 1 a
```

```
NameError: name 'a' is not defined
```



파이썬은 C언어나 자바 등 다른 언어와는 다르게 변수 생성 시 변수의 타입을 적지 않음

```
a = 1  
b = '문자'  
c = 1.0  
d = 4+1j
```

파이썬의 식별자와 변수

2 파이썬의 변수

1 변수의 생성과 사용



변수에 값이 할당될 때 변수의 타입이 결정됨

```
a:    <class 'int'>
b:    <class 'str'>
c:    <class 'float'>
d:    <class 'complex'>
```

파이썬의 식별자와 변수

2 파이썬의 변수

2 변수의 삭제



del 함수를 사용해서 변수를 삭제할 수 있음

```
a = 1  
print(a)
```

1

```
del a  
print(a)
```

NameError

```
<ipython-input-59-c342333e421e> in <  
      1 del a  
----> 2 print(a)
```

NameError: name 'a' is not defined



파이썬의 입·출력 및 기초 문형

1 파이썬의 입력과 출력

1 파이썬에서 입력하기



파이썬에서 가장 기본적인 입력은 **input()** 함수를 사용해 받을 수 있음

```
input_num = input()
```

1

```
input("숫자를 입력해주세요 : ")
```

숫자를 입력해주세요 : 111

'111'



파이썬의 입·출력 및 기초 문형

1 파이썬의 입력과 출력

① 파이썬에서 입력하기



입력 값은 타입(Type)이 문자이기 때문에 숫자를 입력 받아 계산하기 위해서는 타입 변환이 필요

```
a = input("숫자를 입력해주세요 : ")
```

숫자를 입력해주세요 : 1

```
print(type(a))
```

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

#타입 변환

```
a = int(a)
```

```
print(type(a))
```

```
print(a)
```

```
<class 'int'>
```

```
1
```

파이썬의 입·출력 및 기초 문형

1 파이썬의 입력과 출력

2 파이썬에서 출력하기



파이썬에서 기본적인 출력은 **print()** 함수를 사용해 출력할 수 있음

- print(출력할 내용)

```
print(1234)
```

1234

```
a = 1  
print(a)
```

1

- 식별자가 없어도 단순 출력을 할 수 있음

```
print(a + 1)
```

2

```
print(4*6)
```

24

파이썬의 입·출력 및 기초 문형

1 파이썬의 입력과 출력

2 파이썬에서 출력하기



파이썬에서 기본적인 출력은 `print()` 함수를 사용해 출력할 수 있음

- 사칙연산 가능

```
print("1+1")  
print(1+1)
```

1+1

2

파이썬의 입·출력 및 기초 문형

1 파이썬의 입력과 출력

2 파이썬에서 출력하기



coma(,)를 활용해 여러 개의 값을 출력할 수 있음

- 구분자(sep)
: 기본 공백으로 되어 있고, 값의 변경 가능

```
print(11111, "안녕하세요", 345)
```

11111 안녕하세요 345

```
print(11111, "안녕하세요", 345, sep="!!")
```

11111!!안녕하세요!!345

```
print(11111, "안녕하세요", 345, sep="\n")
```

11111
안녕하세요
345

파이썬의 입·출력 및 기초 문형

1 파이썬의 입력과 출력

2 파이썬에서 출력하기



아무 것도 적지 않으면 한 줄이 추가됨

- end 옵션이 기본 개행으로 되어 있음

```
print(11)
print()
print("안녕하세요")
```

11

안녕하세요

```
print(11)
print("안녕하세요")
```

11
안녕하세요

```
print(11, end=",,,")
print("안녕하세요")
```

11,,,안녕하세요

파이썬의 입·출력 및 기초 문형

1 파이썬의 입력과 출력

2 파이썬에서 출력하기



구분자와 같이 파라미터 값을 변경해 출력문의 다양한 활용 가능

- Jupyter Notebook에서 **Shift + Tab**으로 해당 함수의 상세 설명 확인 가능

```
print()
```

Docstring:

```
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

Prints the values to a stream, or to sys.stdout by default.

Optional keyword arguments:

file: a file-like object (stream); defaults to the current sys.stdout.

sep: string inserted between values, default a space.

end: string appended after the last value, default a newline.

flush: whether to forcibly flush the stream.

Type: builtin_function_or_method

×

파이썬의 입·출력 및 기초 문형

2 파이썬의 기초 문형

1 파이썬의 주석



뒤에 오는 문장



첫 번째 줄, 일반적인 파이썬 문장 뒤에 올 수 있음

- Jupyter Notebook에서의 단축키는 **Ctrl + /**

```
# 주석입니다
print(1)
```

1

```
# 주석입니다
# 주석입니다
# 주석입니다
print(1)
```

1

```
#주석입니다.
print(1) #주석입니다.
#주석입니다.
#print(1)
```

1

파이썬의 입·출력 및 기초 문형



2 파이썬의 기초 문형

1 파이썬의 주석



여러 줄 주석은 홀따옴표, 쌍따옴표 세 개를 이어 사용함

```
'''
학습자님
이것은 여러줄
주석입니다.
'''
print(1)
```

1

```
"""
학습자님
이것은 여러줄
주석입니다.
"""
print(1)
```

1

파이썬의 입·출력 및 기초 문형

2 파이썬의 기초 문형

2 파이썬의 할당



할당문은 등호(=)를 사용함

```
a = 1+2
# a에 3을 할당합니다.
print(a)
```

3



표현식은 반드시 등호 오른쪽에 와야 함

```
1 + 2 = a
# 표현식이 왼쪽에 올 수 없습니다.
print(a)
```

File "<ipython-input-108-05664ac13a8e"

```
1 + 2 = a
```

^

SyntaxError: can't assign to operator

파이썬의 입·출력 및 기초 문형



2 파이썬의 기초 문형

2 파이썬의 할당



콤마나 등호를 활용해 여러 변수에 값을 할당할 수 있음

```
a , b = 1, 2  
print(a)  
print(b)
```

1
2

```
a = b = 1  
print(a)  
print(b)
```

1
1

파이썬의 입·출력 및 기초 문형



2 파이썬의 기초 문형

2 파이썬의 할당



세미콜론(;)을 활용해 한 줄에 여러 변수 값을 할당할 수 있음

- ‘;’은 문장이 끝났음을 의미

```
a = 1 ; b = 2 ; c = 3
print(a)
print(b)
print(c)
```

```
1
2
3
```

파이썬의 입·출력 및 기초 문형



2 파이썬의 기초 문형

2 파이썬의 할당



두 변수의 값을 콤마, 등호를 활용해서
서로 바꿀 수 있음

```
a = 1; b = 2
print(a, b)
```

1 2

```
a, b = b, a
print(a, b)
```

2 1

```
a = 1
print(a)
```

```
a += 10
print(a)
```

```
a -= 5
print(a)
```

1

11

6

파이썬의 입·출력 및 기초 문형



2 파이썬의 기초 문형

2 파이썬의 할당



사칙연산자를 함께 활용해 새로운 값을 할당할 수 있음

```
b = 2; c = 3
```

```
a = b+c  
print(a)
```

5

```
a = 1  
print(a)
```

```
a += 10  
print(a)
```

```
a -= 5  
print(a)
```

1
11
6

파이썬의 입·출력 및 기초 문형

2 파이썬의 기초 문형

2 파이썬의 할당



표현식이 아닌 할당문은 할당할 수 없음

```
b = 2; c = 3
```

```
a = (b=b+c)
```

```
print(a)
```

```
File "<ipython-input-121-4>:"
```

```
a = (b=b+c)
```

```
^
```

```
SyntaxError: invalid syntax
```

파이썬의 입·출력 및 기초 문형

2 파이썬의 기초 문형

3 파이썬의 연속 라인



역슬래시(\)를 사용해 한 줄을 여러 줄로 표현할 수 있음

- 코딩이 길어져 한 줄에 표현하기 어려운 경우 사용할 수 있음

```
a = 1
b = 2
```

```
a +
b
```

File "<ipython-input-103-8b9"

```
a +
  ^
```

SyntaxError: invalid syntax

```
a + \
b
```

```
3
```

파이썬의 예약어 및 내장 함수

1 파이썬의 예약어

① 예약어(Keyword)란?

파이썬에서 이미 문법적인 용도로 사용을 하고 있기 때문에 변수 등의 식별자로 사용할 수 없는 단어들



예약어는 변수로 사용할 수 없음

```
False = 1
```

```
File "<ipython-input-8-1950c547d36b>", line 1
```

```
False = 1
```

```
^
```

```
SyntaxError: can't assign to keyword
```


파이썬의 예약어 및 내장 함수

1 파이썬의 예약어

① 예약어(Keyword)란?



파이썬의 기본 모듈 중 하나인 Keyword 모듈을 import해서 확인 가능

```
import keyword
print(keyword.kwlist)
```

```
['False', 'None', 'True',
t', 'finally', 'for', 'fr
```



예약어의 개수는 총 33개

```
'False', 'None', 'True', 'and', 'as', 'assert', 'break',
'class', 'continue', 'def', 'del', 'elif', 'else', 'except',
'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is',
'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise',
'return', 'try', 'while', 'with', 'yield'
```



파이썬의 예약어 및 내장 함수

2 파이썬의 내장 함수

1 내장 함수란?

파이썬에서는 자주 사용되는 함수를
내장 함수라는 이름으로 기본적으로 제공

내장 함수 목록 URL

<https://docs.python.org/ko/3/library/functions.html>

파이썬의 예약어 및 내장 함수

2 파이썬의 내장 함수

2 대표적인 내장 함수



max, min, type, len, range, str, int ...

1

type() : 해당 식별자의 타입을 확인할 수 있음

```
a = 1
print(type(a))
```

<class 'int'>

```
a = "문자"
print(type(a))
```

<class 'str'>

```
a = 1+4j
print(type(a))
```

<class 'complex'>



파이썬의 예약어 및 내장 함수

2 파이썬의 내장 함수

2 대표적인 내장 함수

2

len() : 객체의 길이를 알 수 있음

```
a = '1234'
print(len(a))
```

4

```
a = [1,2,3,4,5]
print(len(a))
```

5

3

max(), min() : 최대값, 최소값을 출력함

```
print(max("abcdefg"))
```

g

```
print(min("12345"))
```

1

Run! 프로그래밍

Mission 1

한 줄에 여러 변수 값 할당

```
print("안녕하세요");print("학습자님");  
print("이번 회차도 수고하셨습니다.")
```

또는

```
print("안녕하세요","학습자님","이번 회차도  
수고하셨습니다.", sep='Wn')
```

Mission 2

입력하고 출력하기

```
a = input("첫 번째 숫자를 입력해주세요")  
b = input("두 번째 숫자를 입력해주세요")print(a+b)
```