

# 인공지능프로그래밍

게임콘텐츠학과 박경수

<https://github.com/ggorr/Machine-Learning/tree/master/Python>

# 교재


- ▶ A Byte Of Python
  - ▶ C. H. Swaroop

# 교재

▶ <https://github.com/swaroopch/byte-of-python/releases/>

## Release





### va62ee8caa3475e8733c40918bbe16719b4904693

 github-actions released this on 23 Jan

[update](#) [downloads](#) [info](#)

---

▼ Assets 4

 <a href="#">byte-of-python.epub</a>	1.12 MB
 <a href="#">byte-of-python.pdf</a>	4.01 MB
 <a href="#">Source code (zip)</a>	
 <a href="#">Source code (tar.gz)</a>	

# 교재

▶ [https://www.oss.kr/oss\\_guide](https://www.oss.kr/oss_guide)

공개SW 솔루션 프로파일  
**공개SW 가이드/보고서**  
공개SW 기술답  
공개SW 테스트

공개SW 보안취약점  
공개SW 활용 성공사례  
공개SW 기술지원

 과학기술정보통신부  
Ministry of Science and ICT

 정보통신산업진흥원  
National IT Industry Promotion Agency

 공개SW 개발자대회  
OSS GRAND DEVELOPERS CHALLENGE

 공개SW개발자 Lab

 한국공개SW  
활성화포럼

번호	제목	작성자	조회
353	[기고] 오픈소스 소프트웨어를 사용하기 전에 고려해...	OSS관리자1	85
352	[월간브리핑] 2019 공개SW 컨트리뷰션 페스티벌 성료	OSS관리자1	47
351	"AI 특허, 데이터 처리 컴퓨터상서 이뤄진다는 점 명...	OSS관리자1	31
350	[월간브리핑] 오픈소스도 전문교육이 필요하다	OSS관리자2	60
349	[기고]오픈소스 소프트웨어 교육에 관한 접근법	OSS관리자2	74
348	국내외 공개SW 교육 현황	OSS관리자2	72
347	[공개SW 월간브리핑] 한국의 개방형 OS, 리눅스 생태...	OSS관리자	99
346	보안성을 강화한 '구름OS'	OSS관리자	267
345	사용자 친화적인 '하모니카OS'	OSS관리자	126
344	[공개SW 월간브리핑]오픈소스 컴플라이언스 준수와...	OSS관리자	62

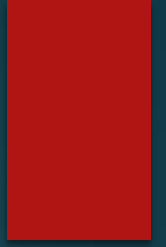
< 1 2 3 4 5 6 7 8 ... 35 36 >

제목

python

검색

# 제 5 장 기초



# 5.1 주석

- ▶ Python 2.x
  - ▶ `print '...'`
- ▶ Python 3.x
  - ▶ `print('...')`

예제: **Python 2.x**

```
print 'hello world' # Note that print is a statement
```

또다른 예제:

```
# Note that print is a statement  
print 'hello world'
```

```
print('hello world') # Note that print is a statement  
  
# Note that print is a statement  
print('hello world')
```

## 5.2 리터럴 상수

- ▶ 숫자
  - ▶ 5 , 1.23
- ▶ 문자열(string)
  - ▶ 'This is a string', "It's a string!"
  - ▶ 따옴표 ' 와 "는 같은 역할

## 5.3 숫자형

- ▶ Integer(정수형)
  - ▶ 2
- ▶ Float(부동 소수점 숫자형)
  - ▶ 3.23 , 52.3E-4
    - ▶  $52.3E-4 = 52.3 \times 10^{-4}$



### 숙련된 프로그래머들을 위한 주석

파이썬에서는 long 형이 따로 없습니다. 대신, int 형에 어떤 크기의 정수든지 담을 수 있습니다.



## 5.4 문자열



```
print('This is a string')
print("This is a string")
print("What's your name?")
print('''What's your
name?''')
```

```
"C:\Program Files\Python37\python.exe"
This is a string
This is a string
What's your name?
What's your
name?

Process finished with exit code 0
```



### C/C++ 프로그래머들을 위한 주석

파이썬에서는 `char` 형이 따로 구분되어 있지 않습니다. 파이썬에서는 이것이 딱히 필요가 없습니다. 곧 여러분도 `char` 형을 찾지 않게 될 것입니다.

## 5.4 문자열

### ▶ 문자열 포매팅

```
age = 20
name = 'Swaroop'
str1 = '{0} was {1} years old'.format(name, age)
str2 = 'Why is {0} playing with that python?'.format(name)
str3 = f'{name} was {age} years old when he wrote this book'

# old format
str4 = '%s was %d years old when he wrote this book' % (name, age)

str5 = name + ' was ' + str(age) + ' years old'
```

## 5.4 문자열

- ▶ 문자열 포매팅
  - ▶ 왼쪽 정렬 <
  - ▶ 가운데 정렬 ^
  - ▶ 오른쪽 정렬 >

```
x = 1.2
print(f'x prints {x}')
print(f'x:7.2 prints {x:7.2}')
# 공백을 0으로 채운다
print(f'x:07.2 prints {x:07.2}')
# 가운데 정렬(^)하고 공백을 _로 채운다
print(f'x:_^7.2 prints {x:_^7.2}')
# 왼쪽 정렬(<)하고 공백을 @으로 채운다
print(f'x:@<7.2 prints {x:@<7.2}')
# 오른쪽 정렬(>)하고 공백을 $로 채운다
print(f'x:$>7.2 prints {x:$>7.2}')
```

```
x prints 1.2
x:7.2 prints      1.2
x:07.2 prints 00001.2
x:_^7.2 prints __1.2__
x:@<7.2 prints 1.2@@@
x:$>7.2 prints $$$$1.2
```

## 5.4 문자열

- ▶ 이스케이프 문자(escape sequence)

- ▶ \n – 줄 바꿈

- ▶ \t – tab

- ▶ \\ – \

```
print('abc\nde\\f')  
print(r'abc\nde\\f')  
print(r'C:\Users\worker\python')
```

```
"C:\Program Files\Python37\python.exe  
abc  
de\f  
abc\nde\\f  
C:\Users\worker\python  
  
Process finished with exit code 0
```

## 5.14 들여쓰기

- ▶ 행 앞의 공백이 일정해야 한다

```
print('abc\nde\\f')
print(r'abc\nde\\f')
print(r'C:\Users\worker\python')
```

```
for i in range(10):
    j = i + 1
    print(j)
```