



# 지능로봇공학과

## INTELLIGENT ROBOT

담당 교수	임종관 교수님
학과	지능로봇학과
학번	1558021
이름	이영섭

# 목 차

code03-01 . . . . . 3P

code03-02 . . . . . 4P

code03-03 . . . . . 5P

code03-04 . . . . . 6P

code03-05 . . . . . 7P

응용예제1 . . . . . 8P~9P

응용예제2 . . . . . 10P

참고문헌 . . . . . 11P

※code03-01

```
print("%d"%123) #정수의 자릿수만큼 출력
print("%5d"%123) #정수를 출력(5자리 확보, 오른쪽으로 정렬)
print("%05d"%123) #정수를 출력(5자리 확보, 오른쪽으로 정렬, 빈칸을 0으로 채움)
print("%f"%123.45) #실수 소수점아래 6자리까지 무조건 출력
print("%7.1f"%123.45) #실수를 출력(7자리 확보, 소수점아래 둘째자리에서 반올림)
print("%7.3f"%123.45) #실수를 출력(7자리 확보, 소수점아래 넷째자리에서 반올림)
print("%s"%python) #문자열을 자릿수만큼 출력
print("%10s"%python) #문자열을 출력(10자리 확보)
```

```
print("%d"%123)
print("%5d"%123)
print("%05d"%123)
print("%f"%123.45)
print("%7.1f"%123.45)
print("%7.3f"%123.45)
print("%s"%python)
print("%10s"%python)

===== RESTART: C:\Users#\
=====
123
   123
  00123
123.450000
   123.5
  123.450
python
   python
>>>
```

※code03-02

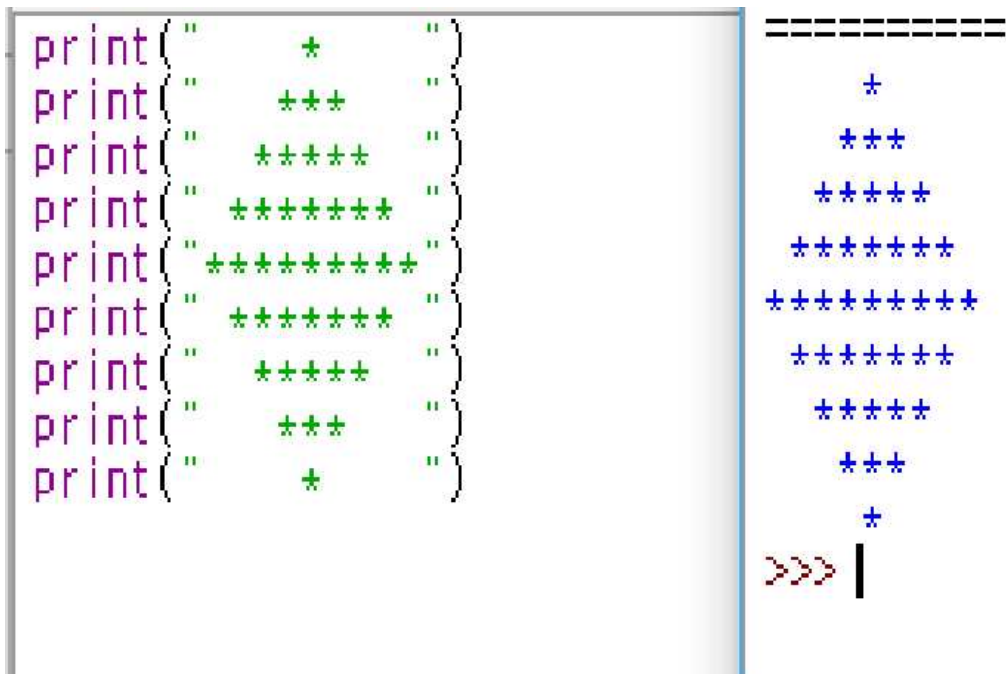
```
print("\n줄바꿈\n연습") #\n은 줄바꿈으로 인식하여 출력된다.
print("\t탭키\t연습") #\t은 다음탭으로 이동하여 출력된다.
print("글자가\"강조\"되는 효과1") #\"는 \"를 그대로 출력하여 글자가 강조되어 출력
된다.
print("글자가\'강조\'되는 효과2") #\'는 \'를 그대로 출력하여 글자가 강조되어 출력
된다.
print("\\\\\\\\\\\\역슬래시 세개 출력") #\\는 \를 출력한다.
print(r"\n\t\"\\를 그대로출력") #r\"는 이스케이프문자까지도 그대로 출력한다.
```

```
print("\n줄바꿈\n연습")
print("\t탭키\t연습")
print("글자가\"강조\"되는 효과1")
print("글자가\'강조\'되는 효과2")
print("\\\\\\\\\\\\역슬래시 세개 출력")
print(r"\n\t\"\\를 그대로출력")
```

줄바꿈  
연습  
          탭키      연습  
글자가"강조"되는 효과1  
글자가'강조'되는 효과2  
\\\\\\\\\\역슬래시 세개 출력  
\\n\t\\"\\를 그대로출력  
>>> |

※code03-03

```
print("    *    ") #그대로 출력
print("   ***   ") #그대로 출력
print("  ***** ") #그대로 출력
print(" ***** ") #그대로 출력
print("*****") #그대로 출력
print(" ***** ") #그대로 출력
print("  ***** ") #그대로 출력
print("   ***   ") #그대로 출력
print("    *    ") #그대로 출력
```



※code03-04

```
sel=int(input("입력 진수 결정(16/10/8/2):")) #입력될 진수를 결정한다
num=input("값 입력:") #문자열을 입력받는다(16진수일 경우에 대비하여 문자
열로 입력받음)
if sel==16: #16진수일경우
    num10=int(num,16) #16진수를 10진수로 바꾼다(문자열=>정수)
if sel==10: #10진수일경우
    num10=int(num,10) #(문자열=>정수)
if sel==8: #16진수일경우
    num10=int(num,8) #16진수를 10진수로 바꾼다(문자열=>정수)
if sel==2: #16진수일경우
    num10=int(num,2) #16진수를 10진수로 바꾼다(문자열=>정수)
print("16진수==>",hex(num10)) #10진수를 16진수로 출력한다
print("10진수==>",num10) #10진수로 출력한다
print("8진수==>",oct(num10)) #10진수를 8진수로 출력한다
print("2진수==>",bin(num10)) #10진수를 2진수로 출력한다
```

```
sel=int(input("입력 진수 결정(16/10/8/2):"))
num=input("값 입력:")
if sel==16:
    num10=int(num,16)
if sel==10:
    num10=int(num,10)
if sel==8:
    num10=int(num,8)
if sel==2:
    num10=int(num,2)
print("16진수==>",hex(num10))
print("10진수==>",num10)
print("8진수==>",oct(num10))
print("2진수==>",bin(num10))
```

```
Python 3.5.3 (v3.5.3:1880cb95a742,
D64)] on win32
Type "copyright", "credits" or "lic
>>>
===== RESTART: C:/Users/명섭/[
=====
입력 진수 결정(16/10/8/2):16
값 입력:ae
16진수==> 0xae
10진수==> 174
8진수==> 0b256
2진수==> 0b10101110
>>>
```

※code03-05

```
def myFunc(): #함수를 선언한다
    print('함수를 호출함') #출력한다
gVar=100 #전역변수를 선언한다
if __name__=='__main__': #파이썬 내부의 특수한 변수인 __name__변수
    에 __main__값을 자동으로 설정
    print('메인 함수 부분이 실행됩니다.') #출력
    myFunc() #함수 실행
    print('전역 변수 값:',gVar) #출력
```

```
def myFunc():
    print('함수를 호출함')
gVar=100
if __name__=='__main__':
    print('메인 함수 부분이 실행됩니다.')
    myFunc()
    print('전역 변수 값:',gVar)
```

```
Python 3.7.1 Shell
Type "copyright", "credits" or "
>>>
===== RESTART: C:/Users/영민/
=====
메인 함수 부분이 실행됩니다.
함수를 호출함
전역 변수 값: 100
>>> |
```

## ※응용예제1

```
import sys #sys를 임포트한다
intVar,floatVar,boolVar,strVar,listVar,tupleVar,dictVar,setVar=[None]*8
#변수를 선언한다
if __name__=="__main__": #파이썬 내부의 특수한 변수인 __name__변수에
__main__값을 자동으로 설정
    intVar=0 #변수를 대입한다(정수)
    floatVar=0.0 #변수를 대입한다(실수)
    boolVar=True #변수를 대입한다(불형)
    strVar='' #변수를 대입한다(문자열)
    listVar=[] #변수를 대입한다(list형)
    tupleVar=() #변수를 대입한다(tuple형)
    dictVar={} #변수를 대입한다(dictionary형)
    setVar=set() #변수를 대입한다(set형)
    print('int형 기본 크기-->',sys.getsizeof(intVar)) #데이터형의 기본크기
    를 바이트단위로 출력
    print('float형 기본 크기-->',sys.getsizeof(floatVar)) #데이터형의 기본
    크기를 바이트단위로 출력
    print('bool형 기본 크기-->',sys.getsizeof(boolVar)) #데이터형의 기본크
    기를 바이트단위로 출력
    print('str형 기본 크기-->',sys.getsizeof(strVar)) #데이터형의 기본크기
    를 바이트단위로 출력
    print('list형 기본 크기-->',sys.getsizeof(listVar)) #데이터형의 기본크기
    를 바이트단위로 출력
    print('tuple형 기본 크기-->',sys.getsizeof(tupleVar)) #데이터형의 기본
    크기를 바이트단위로 출력
    print('dictionary형 기본 크기-->',sys.getsizeof(dictVar)) #데이터형의
    기본크기를 바이트단위로 출력
    print('set형 기본 크기-->',sys.getsizeof(setVar)) #데이터형의 기본크기
    를 바이트단위로 출력
```



```

import sys
intVar, floatVar, boolVar, strVar, listVar, tupleVar, dictVar, setVar=[None]*8
if __name__=="__main__":
    intVar=0
    floatVar=0.0
    boolVar=True
    strVar=''
    listVar=[]
    tupleVar=()
    dictVar={}
    setVar=set()
    print('int형 기본 크기-->',sys.getsizeof(intVar))
    print('float형 기본 크기-->',sys.getsizeof(floatVar))
    print('bool형 기본 크기-->',sys.getsizeof(boolVar))
    print('str형 기본 크기-->',sys.getsizeof(strVar))
    print('list형 기본 크기-->',sys.getsizeof(listVar))
    print('tuple형 기본 크기-->',sys.getsizeof(tupleVar))
    print('dictionary형 기본 크기-->',sys.getsizeof(dictVar))
    print('set형 기본 크기-->',sys.getsizeof(setVar))

```

```

Python 3.5.3 (v3.5.3:1880cb95a742,
D64)) on win32
Type "copyright", "credits" or "li
>>>
===== RESTART: C:/Users/영{
y =====
int형 기본 크기--> 24
float형 기본 크기--> 24
bool형 기본 크기--> 28
str형 기본 크기--> 49
list형 기본 크기--> 64
tuple형 기본 크기--> 48
dictionary형 기본 크기--> 288
set형 기본 크기--> 16
>>>

```

## ※응용예제2

```
inStr='' #inStr변수를 준비한다
if __name__=="__main__": #파이썬 내부의 특수한 변수인 __name__변수에
__main__값을 자동으로 설정
    inStr=input('문자열을 입력-->') #문자열을 입력받는다
    for i in range(len(inStr)-1,-1,-1): #for문을 이용하여 뒤에서부터 한글
자씩 불러온다
        print('%c'%inStr[i],end='') #위에서 불러온 글자를 출력한다
```

```
inStr=''
if __name__=="__main__":
    inStr=input('문자열을 입력-->')
    for i in range(len(inStr)-1,-1,-1):
        print('%c'%inStr[i],end='')
```

```
===== RESTART: C:/Users/김
> =====
문자열을 입력-->고급프로그래밍
미래그로프급고
>>>
```

우재남, 파이썬 of beginner, 한빛아카데미,  
2017  
60P~90P