

# SW개발기초 중간고사

- 01. 파이썬 소개
- 02. 파이썬 사용해보기
- 03. 연산자와 변수 활용
- 04. 기본 출력 함수: print
- 05. 자료형
- 06. input 함수
- 07. 터틀그래픽
- 08. 양식문자
- 09. 조건문

## 01. 파이썬 소개

파이썬은 귀도 반 로섬이 1991년 만듬

오픈소스 라이브러리

인터프리터

## 02. 파이썬 사용해보기

셸 모드: >>>이 들어가는 모드, print()를 생략 가능. 바로바로 결과 나옴

코드 편집기: 저장 후 실행해야 결과 나옴.

## 03. 연산자와 변수 활용

+ - \* / // % \*\*

## 04. 기본 출력 함수: print

```
import time
time.sleep(1)
#몇 초 뒤 다음 줄 실행하는 코드
```

```

변수="A"
print(변수, "를 출력해봅시다.")
#A 를 출력해봅시다.

#이렇게 #을 붙이면 주석(코드 실행X)
"""
이것도 주석입니다. (print()함수 밖에 있을 때)
"""
'''
이것도 주석입니다. (print()함수 밖에 있을 때)
'''

print("""
\n 새로운 줄
\t 탭(스페이스4칸)
\' 작은 따옴표'
\" 큰 따옴표"
\\ 문자\
\b 뒤에 한 칸 지우기

이스케이프 문자는 모두 문자열(초록글씨) 안에서 써야함
""")

print("Hello", end=" ") #Hello <<<공백 1칸. 끝부분에 무엇을 넣을지(기본: \n)
print("Easy", "Python", sep="") #쉼표 사이에 무엇이 들어갈지. (기본: " ")

```

## 05. 자료형

```

'문자열입니다'

"문자열입니다"

"""이건 여러줄 문자열
입니다"""

'''이것도 여러줄 문자열
입니다'''

'A' <-- 문자

1 <-- 정수

1.0 <-- 실수

int()
float()

```

```

str()

bin(4) '0b100' #2진수
ord('A') 65 #문자의 유니코드
chr(65) 'A' #유니코드를 문자로
type() #자료형 알려줌
len("Python") #Python의 길이(몇 자)를 int형으로 알려줌

msg="파이썬의 모든 연산은"+str(10)+"같이 str()을 사용하여 자료형을 바꾼뒤"+"더하기 기호로 합친다"

```

## 06. input 함수

변수 = input() #값 입력시 '변수'에 문자열로 저장한다.

## 07. 터틀그래픽

```

import turtle
t1=turtle.Turtle() #동일: t1=turtle.Turtle(); t1.shape("turtle")
t1.shape("turtle") #종류: arrow, triangle, classic, turtle, circle

t.color("red") #펜색
t.fillcolor("orange") #채우기 색

t.begin_fill()
t.end_fill()

```



```

import turtle as t
t.forward(100) #t.fd(100)

```

```

t.backward(100) #t.bk(100)
t.right(90) #t.rt(90)
t.left(90) #t.lt(90)
t.home() #원점, 0,0으로 이동

t.circle(100, 90) #반지름, 몇 도까지 그릴지 반지름100, 90도만

받을값 = t.textinput("제목", "내용") #input()함수와 비슷함
t.write("쓸 텍스트 내용")

t.penup()
t.pendown()

```

## 08. 양식문자

```

"%%" #%
"{" "{
"}" #}

"""
%d 정수
%f 실수
%g 소수점에 따라 정수, 실수 자동표시 (1.0 입력시: 1, 1.5 입력시: 1.5)
%s 문자열
%c 문자 ('A')
%o 8진수
%x 16진수

```

위 형식문자는 모두 문자열(초록글씨) 안에서 써야함

```

"""
>>> "이렇게 쓰며, %d 순서대로 %s." % (0, "출력합니다.")
#이렇게 쓰며 0 순서대로 출력합니다.
>>> "문자열 뒤에 %기호를 %s 가장 중요해요." % "붙이는 것이"
#문자열 뒤에 %기호를 붙이는 것이 가장 중요해요.
>>> "%5.2f를 붙이면 5칸 자릿수, 소수점은 2자리까지" % 1.23456
# 1.23를 붙이면 5칸 자릿수, 소수점은 2자리까지
>>> "%05d는 앞에 5칸 자릿수 빈칸 0으로 채운다." % (1)
#00001는 앞에 5칸 자릿수 빈칸 0으로 채운다.

```

```

"""
d 10진수
f 실수
e 지수
b 2진수
o 8진수
x 16진수
s 문자열

```

문자 c는 없어요

위 포맷문자는 모두 문자열(초록글씨) 안에서 써야함

```

"""

```

```
>>> "이런 식으로 {}".format ("쓸 수 있다.")
#이런 식으로 쓸 수 있다.
>>> "이런 식으로도 {0} {1}".format ("쓸 수", "있다.")
#이런 식으로도 쓸 수 있다.
>>> "이런 식을 {0:s} 많이 {1:s}".format ("제일", "쓴다.")
#이런 식을 제일 많이 쓴다.
>>> "포맷문자는 {1:s} 중요하므로 {0:s}!".format("주의", "위치가")
#포맷문자는 위치가 중요하므로 주의!
>>> "%문자는 못하는데 포맷문자는 {0:,d} 심표 쓸 수 있어요".format(1000)
#%문자는 못하는데 포맷문자는 1,000 심표 쓸 수 있어요
```

## 09. 조건문

if

if ... else ...

if ... elif ... elif ... else

```
if 조건문a : #콜론 사용하기
    명령문a #들여쓰기는 Tab 한번만!
elif 조건문b : #elif문은 if 다음에(또는 elif다음에) 들어가야합니다.
    명령문b
else : #else문은 조건문X
    명령문c
```

- 조건1 and 조건2    둘다 참이면 실행  $p \wedge q$
- 조건1 or 조건2    하나라도 참이면 실행  $p \vee q$
- not (조건)    조건이 참이면 False, 거짓이면 True  $\sim p$
- $x == y$
- $x != y$
- $x > y$
- $x \geq y$
- $x < y$
- $x \leq y$