

# | 파이썬 스터디 |

1주차

# | 스터디 개요

- 장소: 학술정보원 2층 교육실
- 시간: 월(13시 ~ 14시), 목(12시 ~ 13시), 금(13시 ~ 14시)
- 진도: 수업계획서에서 중간고사 전 범위까지
- 질문: 언제나 환영
- 연락: 010-2536-8746, 카카오톡

# index

## 1. 프로젝트 준비

- 파이썬 다운로드
- 소스 파일 생성
- 기본 코드 형식

## 2. 변수와 연산자

- 변수
- 연산자
- 형변환

## 3. 입출력

- print
- Input
- 입출력 고급

## 4. 기타

- 교환 알고리즘
- O시시스템 주의사항

# 1. 프로젝트 준비



## | Python IDLE

- <https://www.python.org/downloads/release/python-360/>

### 프로젝트 준비

파이썬 다운로드

소스 파일 생성

기본 코드 형식

|   |         |                        |                                  |          |                     |
|---|---------|------------------------|----------------------------------|----------|---------------------|
| <a href="#">Windows x86-64<br/>embeddable zip<br/>file</a>  | Windows | for<br>AMD64/EM64T/x64 | 0ec0caeea75bae5d2771cf619917c71f | 6925798  | <a href="#">SIG</a> |
| <a href="#">Windows x86-64<br/>executable<br/>installer</a> | Windows | for<br>AMD64/EM64T/x64 | 71c9d30c1110abf7f80a428970ab8ec2 | 31505640 | <a href="#">SIG</a> |
| <a href="#">Windows x86-64<br/>web-based<br/>installer</a>  | Windows | for<br>AMD64/EM64T/x64 | 25b8b6c93a098dfade3b014630f9508e | 1312376  | <a href="#">SIG</a> |
| <a href="#">Windows x86<br/>embeddable zip<br/>file</a>     | Windows |                        | 1adf2fb735c5000af32d42c39136727c | 6315855  | <a href="#">SIG</a> |
| <a href="#">Windows x86<br/>executable<br/>installer</a>    | Windows |                        | 38d9b036b25725f6acb553d4aece4db4 | 30566536 | <a href="#">SIG</a> |
| <a href="#">Windows x86<br/>web-based<br/>installer</a>     | Windows |                        | f71f4590be2cc5cdc43069594d4ea98d | 1286984  | <a href="#">SIG</a> |



## 프로젝트 준비

파이썬 다운로드

소스 파일 생성

기본 코드 형식

## | Python IDLE

- <https://www.python.org/downloads/release/python-360/>
- 자신의 PC가 32비트면 x86, 64비트면 x86-64를 설치  
\*비트 확인법: 내컴퓨터(우클릭) – 속성 – 시스템 종류
- Executable installer: exe파일의 용량이 큰 대신  
exe파일만 있으면 네트워크가 없어도 다운 가능
- Web-based installer: exe파일의 용량이 작은 대신  
네트워크에 연결되어 있어야 다운 가능



## 프로젝트 준비

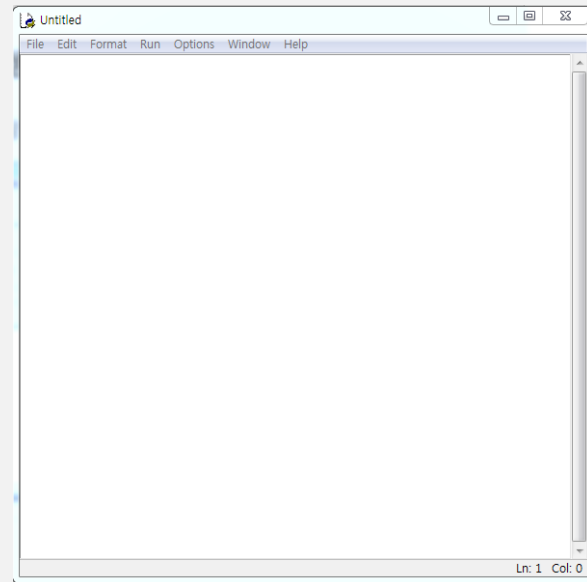
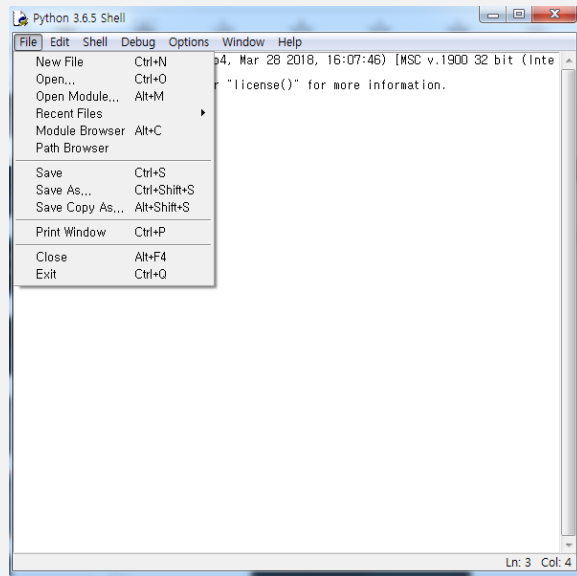
파이썬 다운로드

소스 파일 생성

기본 코드 형식

## | 소스파일 생성

- File -> New File





## 프로젝트 준비

파이썬 다운로드

소스 파일 생성

기본 코드 형식

## | 기본 코드 형식

```
string = input()
print(string)
```

```
#include <stdio.h>

int main()
{
    char str[100];

    scanf("%s", str);
    printf("%s", str);

    return 0;
}
```





## 프로젝트 준비

파이썬 다운로드

소스 파일 생성

기본 코드 형식

## | 기본 코드 형식

```
for i in range(0,5):  
    string = input()  
    print(string)
```

```
for i in range(0,5):  
    string = input()  
    print(string)
```

- 변수 선언 X
- 코드 끝에 세미콜론 X
- 중괄호 필요 X  
-> 대신 콜론 필요

**들여쓰기 중요**

- 간편, 편리

## 2. 변수와 연산자



## 변수와 연산자

변수  
연산자  
형변환

## | 변수와 자료형

- 그릇과 같은 역할, 변할 수 있는 수  
안에는 값이 저장됨
- int (정수) : 1, -5, 123
- float (실수) : 0.5, -2.6, 7.0
- char (문자) : 'a', 'B', 'A'
- string (문자열) : "ABC", "Hello World"
- bool (논리) : True (1), False (0)



## 변수와 연산자

변수

연산자

형변환

## | 연산자

- `+` , `-` , `*` , `/`
- `%` (나머지), `//`(몫)
- `**` (제곱), `**0.5` (제곱근)
- `=` (대입 연산자)
- `a = a + b`  
→ `a+b`의 값을 `a`에 저장한다

`a = 9, b = 5, c = 2, d = 3`

```
print(a+b)      → 14
print(a-b)      → 4
print(a*b)      → 45
print(a/b)      → 1.8
print(a%b)      → 4
print(c**d)     → 8
print(a**0.5)   → 3.0
print(a//b)     → 1
a = b
print(a)        → 5
```



## 변수와 연산자

변수  
연산자  
형변환

### | 연산자

- $+=$ ,  $-=$ ,  $/=$ ,  $*=$ ,  $\%=$ ,  $**=$ ..
- $a += b$   
→  $a = a + b$
- $a *= b$   
→  $a = a * b$

$a = 10, b = 5$

```
a+=b  
print(a)    → 15  
a-=b  
print(a)    → 10  
a*=b  
print(a)    → 50  
a/=b  
print(a)    → 10.0
```



## 변수와 연산자

변수

연산자

형변환

### | 연산자

- `<<, >>`  
-> (`<<: *(2^n)`, `>>: /(2^n)`)
- `a = 2 -> 0010`  
`a << 2 = 8 -> 1000`
- `round` (반올림)

`a = 4, b = 3.3, c = 3.5`

```
print(a>>2)      → 1
print(a<<2)      → 16

print(round(b))   → 3
print(round(c))   → 4
```



## 변수와 연산자

변수

연산자

형변환

### | 연산자

- == , !=

a==b: a랑 b가 같은가?

a!=b: a랑 b가 다른가?

- <, >, <=, >=

- and, or, not

and: 둘다 참이면 True

or: 하나라도 참이면 True

not: 부정(True->False), (False->True)

a = 1, b = 2, c = 2,

boolA= True, boolB = False

|                        |         |
|------------------------|---------|
| print(a==b)            | → False |
| print(a!=b)            | → True  |
| print(a>b)             | → False |
| print(a<b)             | → True  |
| print(a>=c)            | → False |
| print(a<=c)            | → True  |
|                        |         |
| print(boolA and boolB) | → False |
| print(boolA or boolB)  | → True  |
| print(not boolA)       | → False |



## 변수와 연산자

변수  
연산자  
형변환

### | 형변환

•  $a = 3, b = 1.7, c = 5, d = 6$

- `int(value), chr(value)`
- `float(value), str(value)`
- `int(a+0.5)`

-> ex)  $a = 3.3$

->  $a+0.5 = 3.8, \text{int}(a) = 3$

ex)  $a = 3.5$

->  $a+0.5 = 4.0, \text{int}(a) = 4$

```
print(float(a))    → 3.0  
print(int(b))      → 1
```

```
print(c+d)         → 11  
c = str(c)  
d = str(d)  
print(c+d)         → 56
```





## 변수와 연산자

변수

연산자

형변환

## | 형변환

- C

$a+50 = 182, \quad 182/100 = 1.82$

$\text{int}(1.82) = 1, \quad 1*100 = 100$

- d

$b+50 = 208, \quad 208/100 = 2.08$

$\text{int}(2.08) = 2, \quad 2*100 = 200$

$a = 132, b = 158$

```
a = 132
```

```
b = 158
```

```
c = int((a+50)/100)*100
```

```
d = int((b+50)/100)*100
```

```
print(c, d)
```

```
= RESTART: C:/
```

```
100 200
```

```
>>>
```

### 3. 입출력



## 입출력

print

input

입출력 고급

## | 출력

```
print("Hello")  
print("Hello", "World")  
print("Hello" + "World")
```

```
== RESTART: C:/Users/1  
Hello  
Hello World  
HelloWorld  
>>>
```

- 따옴표 안의 문자열 출력.
- 콤마로 문자열 이을 시 띄어쓰기 됨
- +기호로 문자열 합칠 시 띄어쓰기 안됨



## 입출력

print

input

입출력 고급

## | 출력

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

```
string1 = "abc"  
string2 = "def"  
print(string1 + string2)
```

```
...  
== RESTART: C:/Users/ons  
3  
abcdef  
>>>
```

- 변수, 연산의 결과 출력 가능

- +기호

→ 변수가 정수, 실수일 경우

결과값으로 + 연산의 결과가 나옴  
(다른 연산도 가능)

→ 변수가 문자, 문자열일 경우

결과값으로 두 문자열을 합친  
결과가 나옴



## 입출력

print

input

입출력 고급

## | 출력

```
print("ABCDEFGHII₩  
KLMNOPQR")  
print("₩"₩'₩₩")
```

```
== RESTART: C:/User  
ABCDEFGH I JKLMNOPQR  
" '₩  
>>> |
```

- 출력할 문자열이 긴 경우 ₩기호를 사용하여 여러줄에 작성 가능
- ", ', ₩ 문자등을 출력하기 위해서는 ₩뒤에 각 문자를 붙여야 출력됨
- +) 탭키: ₩t, 줄바꿈: ₩n



## 입출력

print

input

입출력 고급

## | 출력

```
i = 123  
f = 3.14  
s = "Hello"
```

```
print("%d %f %s" % (i, f, s))  
print("%5d %5.1f %7s" %  
      (i,f,s))  
print("%05d %05.1f %7s"  
      % (i,f,s))
```

```
== PSIAHI : C:/Users/o  
123 3.140000 Hello  
 123   3.1   Hello  
00123 003.1   Hello  
>>>
```

- 서식문자

정수: %d, 실수: %f, 문자: %c,

문자열: %s

- 자리맞춤

%5d

|  |  |   |   |   |
|--|--|---|---|---|
|  |  | 1 | 2 | 3 |
|--|--|---|---|---|

%05d

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 |
|---|---|---|---|---|



## 입출력

print

input

입출력 고급

## | 입력

```
s = input()
a = int(input())

print(s, a)
```

```
== RESTART: C:/Users,
Hello
5
Hello 5
>>> |
```

- input()

input 함수는 입력된 값을 문자열 형태로 받는다.

- int(input())

정수로 입력받고 싶다면 형변환 시킨다.



## 입출력

print

input

입출력 고급

## | 고급

```
print("Hello", end = "")  
print("World")  
print("My Name is",  
end = "#")  
print("홍길동")
```

```
== RESTART: C:/User  
HelloWorld  
My Name is#홍길동  
>>>
```

- end = char
- print 함수 종료 시 종료문자를 무엇으로 할지 결정

기본값: \n (개행)





## 입출력

print

input

입출력 고급

## | 고급

```
a = 1
b = 2
print(a, b, sep = ' + ',
      end = ' ')
print('=', a+b)
```

```
///
```

```
== RESTART: C:/Ue
```

```
1 + 2 = 3
```

```
>>>
```

- sep = char
- print 함수는 ,로 구분시  
자동으로 공백이 생김
- sep을 쓰면 공백대신 무엇으로  
구분할 지 결정가능



## 입출력

print

input

입출력 고급

## | 고급

```
print("Hello".center(10))  
print("Hello".rjust(10))  
print("Hello".ljust(10))  
print("Hello".zfill(10))
```

```
...  
== RESTART: C:/U  
    Hello  
        Hello  
Hello  
00000Hello  
>>> |
```

- center: 가운데정렬
- rjust: 우측정렬
- ljust: 좌측정렬
- zfill: 빈공간 0으로 채우면서  
우측정렬



## 입출력

print

input

입출력 고급

## | 고급

```
a, b = input().split()
a = int(a)
b = int(b)
print(a+b)
```

```
a,b = input().split(',')
a = int(a)
b = int(b)
print(a*b)
```

```
== RESTART: (
```

```
3 4
```

```
7
```

```
3,4
```

```
12
```

```
>>>
```

- 가로 입력
- `x, y = input().split()`

- `split(char Separator)`

구분자를 기준으로 값을 구분해낸다.

(기본값: 공백)

ex) Separator = ','

세미콜론을 기준으로 구분

## 4. 기타



기타

교환 알고리즘

0시스템 주의사항

## | 교환 알고리즘

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

```
a = b  
b = a
```

```
print(a)  
print(b)
```

a [1] b [2]

a [2] b [2]

a [2] b [2]

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

```
c = a  
a = b  
b = c
```

```
print(a)  
print(b)
```

a [1] b [2]

a [1] b [2] c [1]

a [2] b [2] c [1]

a [2] b [1] c [1]



## 기타

교환 알고리즘

이시스템 주의사항

## I 이 시스템 주의사항

- 문제의 출력 예시와 자신의 코드 출력 비교  
공백(' ') 혹은 개행('\n')문자 등으로 출력이 다르면 X  
과재의 출력과 완전히 일치해야함
- 런타임 에러 && 컴파일 에러  
대부분 문법 오류(문장 끝에 ';' 부재, 스펠링 틀림)  
결과값이 달라도 런타임에러 발생할 수 있음
- 타임리미트 에러  
반복문 탈출 조건이 없는 등의 이유로 실행시간 길어짐



| 스터디 1차 |

감사합니다~