

# 1주차: 변수, 입출력

```
33 self.fingerprints = {}
34 self.logdups = True
35 self.debug = debug
36 self.logger = logging.getLogger(__name__)
37 if path:
38     self.file = open(os.path.join(path, 'requests.log'), 'a')
39     self.file.seek(0)
40     self.fingerprints.update({x: 0 for x in self.fingerprints.keys()})
41
42 @classmethod
43 def from_settings(cls, settings):
44     debug = settings.get('DEBUG', False)
45     return cls(job_dir=settings.get('JOB_DIR', None))
46
47 def request_seen(self, request):
48     fp = self.request_fingerprint(request)
49     if fp in self.fingerprints:
50         return True
51     self.fingerprints.add(fp)
52     if self.file:
53         self.file.write(fp + os.linesep)
54
55 def request_fingerprint(self, request):
56     fp = fingerprint(request)
```

# 커리큘럼

1. 변수, 입출력
2. 조건문, 반복문
3. 배열, 튜플, 세트, 딕셔너리
4. 함수
5. class
6. 알고리즘 입문, 그리디 알고리즘
7. 재귀함수
8. 탐색
9. DP(Dynamic Programming)

# 진행 방향

1. 커리큘럼 순으로 진행
2. 과제는 6문제 정도 최소 1번, 2번 문제는 풀어볼 것,  
뒷번호로 가면 갈수록 어려움
3. 건의 사항, 질문 사항은 언제든지 알려주시면  
감사하겠습니다.

# Python 언어의 특징

문법이 쉬운 편이다.

✓ Hello World! 출력

```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
    return 0;
}
```

C 언어

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Java

```
print("Hello World!")
```

Python

# Python 언어의 특징

✓ 인공지능 라이브러리가 많다.

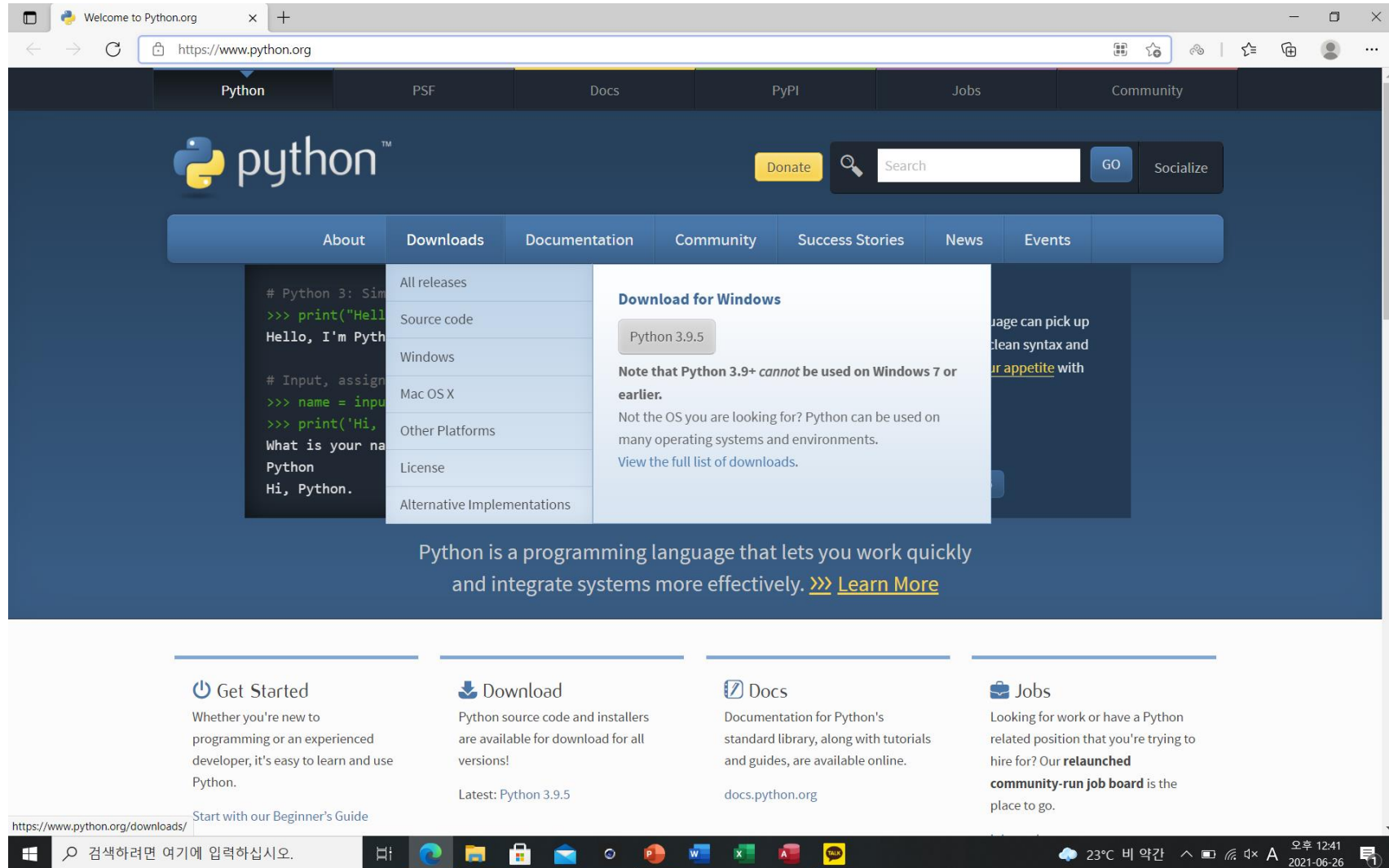
- ML(Machine Learning) Scikit-learn, Tensorflow 등
- DL(Deep Learning) Tensorflow, PyTorch 등

✓ 인터프리터 언어이다.

- Compiler (속도가 빠름) - C
- Interpreter (속도가 느림) - Python, MATLAB

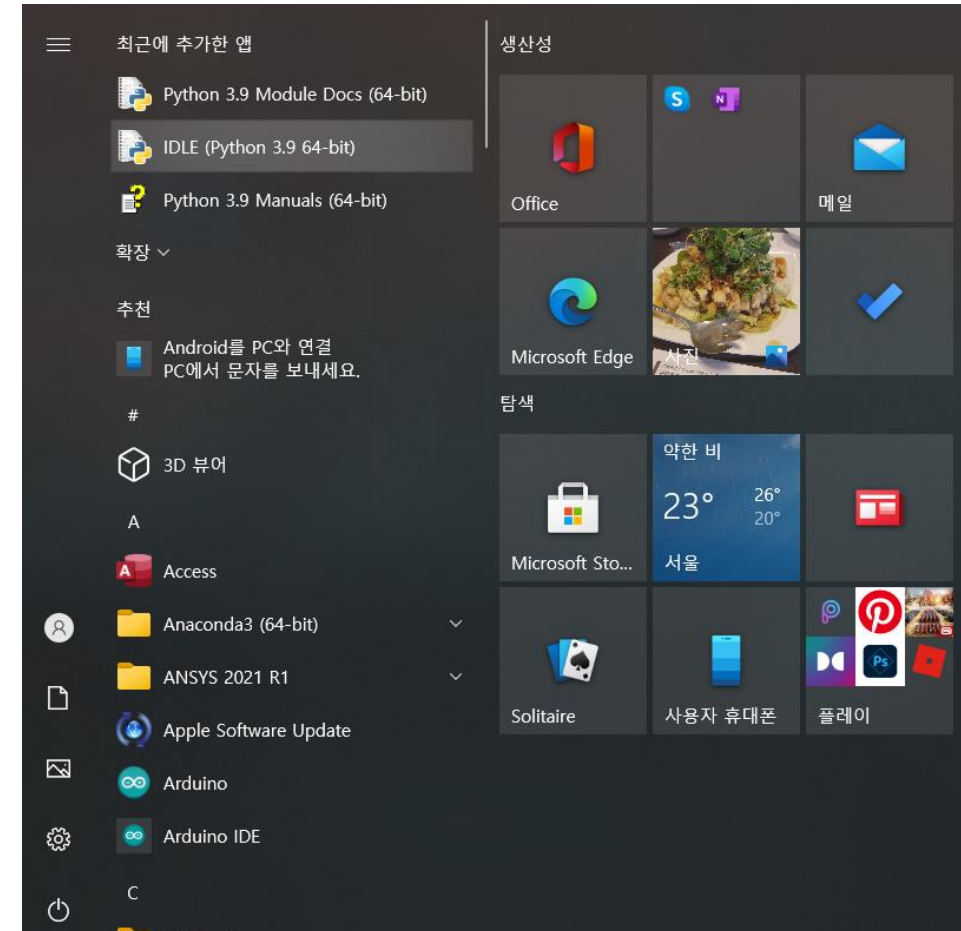
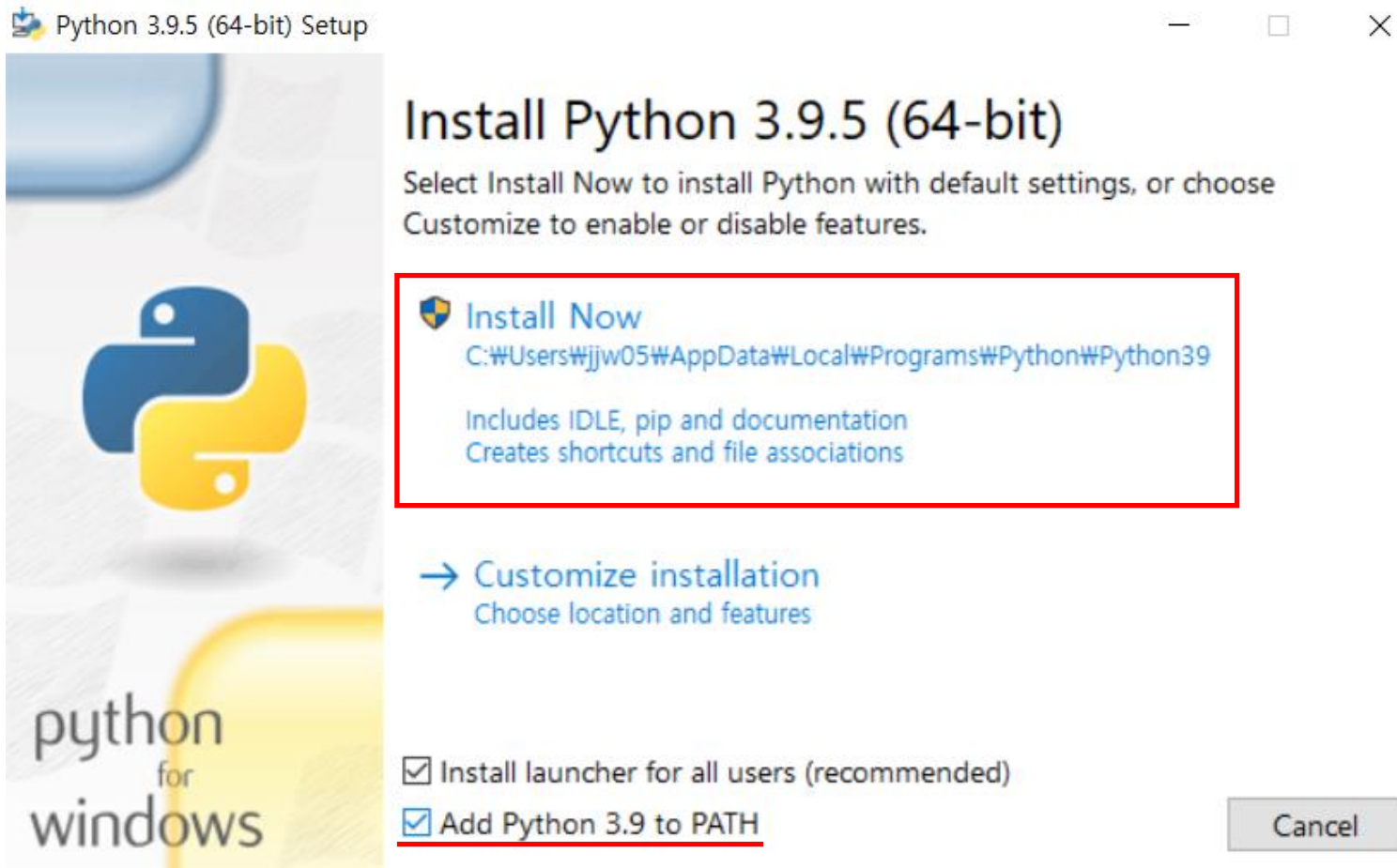
# Python 다운로드

<https://www.python.org/>

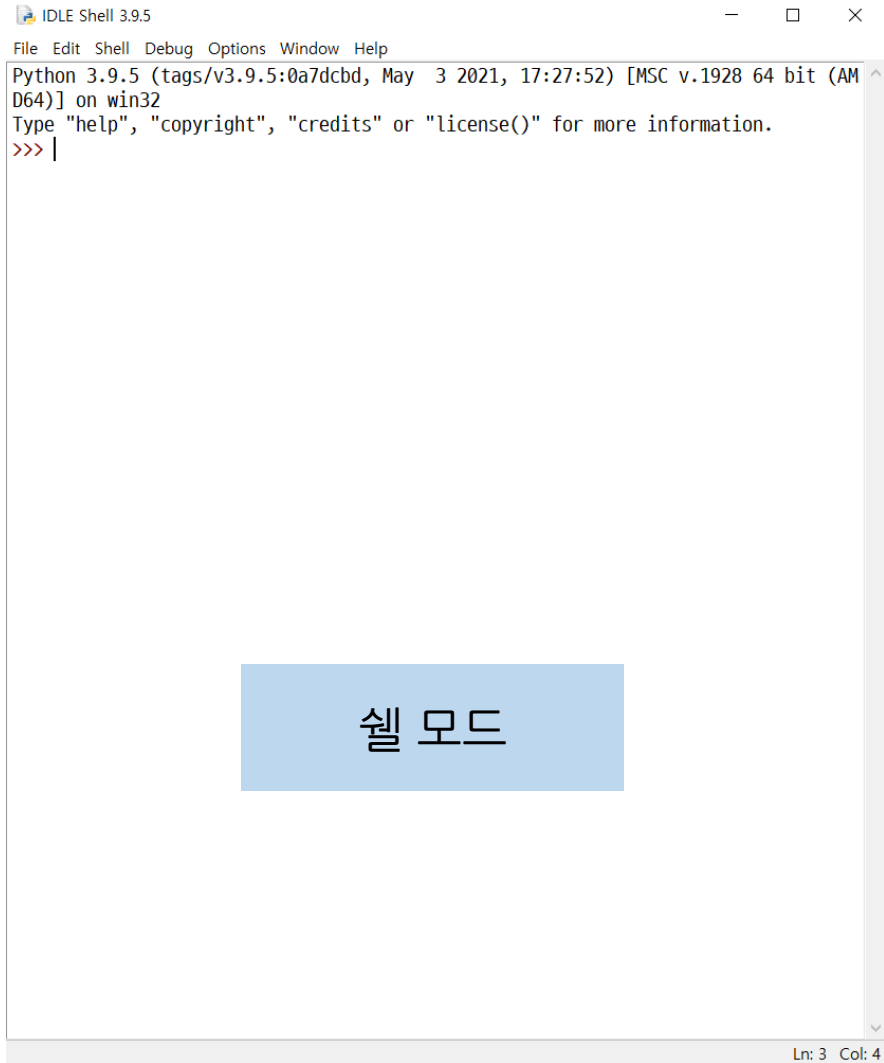




# Python 다운로드



# Python IDLE



The image shows the Python IDLE Shell 3.9.5 window. The title bar reads "IDLE Shell 3.9.5". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the Python 3.9.5 startup message: "Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32". Below this, it says "Type 'help', 'copyright', 'credits' or 'license()' for more information." and shows the prompt ">>> |". The status bar at the bottom indicates "Ln: 3 Col: 4".

셸 모드

File + New File



The image shows the Python IDLE Editor window. The title bar reads "untitled". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The main text area is empty. The status bar at the bottom indicates "Ln: 1 Col: 0".

에디터 모드



# 기본 연산

연산자	의미	사용 예시	결과
+	덧셈	$3+2$	5
-	뺄셈	$3-2$	1
*	곱셈	$3*2$	6
/	나눗셈	$3/2$	1.5
//	정수형 나눗셈	$3//2$	1
%	나머지	$3\%2$	1
**	지수	$3**2$	9

# 변수

```
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 3+2
5
>>> 3-2
1
>>> 3*2
6
>>> 3/2
1.5
>>> 3//2
1
>>> 3%2
1
>>> 3**2
9
>>> |
```

```
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', '__peg_parser__', 'and', 'as', 'assert', 'async', 'await', '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']
```

## 필요성:

- 어떤 의미가 있는 값인지 이름 기억
- 반복해서 쓸 값 - 이름 지정

## 형태:

- $x=100$  (변수=대입 값)
- 등호 : 같다(X), 대입하다(O)

## 이름 규칙:

- 영문, 숫자, \_ 만 사용 가능
- 숫자로 시작할 수 없음
- 공백 불가능
- 특정 이름은 불가능

# 변수

```
>>> a=3
>>> a
```

```
>>> a=3
>>> a=5
>>> a
```

```
>>> a=3
>>> a=a+2
>>> a
```

```
>>> a=3
>>> a=5
>>> a=a+a
>>> a
```

```
>>> a=3
>>> b=2
>>> a=a+b
>>> a
```

```
>>> width=4
>>> height=5
>>> area=0.5*width*height
>>> area
```

# 복합 대입 연산자

연산자	의미	사용 예시	결과
<code>+=</code>	덧셈 후 대입	<code>a+=b</code>	<code>a=a+b</code>
<code>-=</code>	뺄셈 후 대입	<code>a-=b</code>	<code>a=a-b</code>
<code>*=</code>	곱셈 후 대입	<code>a*=b</code>	<code>a=a*b</code>
<code>/=</code>	나눗셈 후 대입	<code>a/=b</code>	<code>a=a/b</code>
<code>%=</code>	나머지 구한 후 대입	<code>a%=b</code>	<code>a=a%b</code>
<code>**=</code>	제곱 후 대입	<code>a**=b</code>	<code>a=a**b</code>

# 비교 연산자

연산자	의미	사용 예시	결과
==	같다	3==2	False
!=	다르다	3!=2	True
>	크다	3>2	True
<	작다	3<2	False
>=	크거나 같다	3>=2	True
<=	작거나 같다	3<=2	False

Boolean

# 논리 연산자

연산자	의미	사용 예시	결과
and	양쪽이 True인 경우에만 True, 나머지는 False	3>2 and 3>=2 3>2 and 3==2	True False
or	양쪽이 False인 경우에만 False, 나머지는 True	3<2 and 3==2 3>2 and 3==2	False True
not	결과 반전	not False	True



# 자료형

- int(정수), float(실수), str(문자열)
- Python은 자료형을 자동으로 파악.
- 현재의 자료형이 궁금하면 type() 함수를 사용.

```
>>> a=0
>>> type(a)
<class 'int'>
>>> a=10
>>> type(a)
<class 'int'>
>>> a=3.14
>>> type(a)
<class 'float'>
>>> a='Hello World!'
>>> type(a)
<class 'str'>
```

```
>>> width=4
>>> height=5
>>> area=0.5*width*height
>>> area
10.0
>>> type(area)
<class 'float'>
>>> int(area)
10
>>> str(area)
'10.0'
```

# 입출력

## ✓출력

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

value : 자료형을 가리지 않고 출력 가능

sep : value1, value2 사이 넣을 문자열, 기본은 공백

end : 모두 출력하고 마지막에 넣을 문자열, 기본은 줄 바꾸기(\n : 줄 바꾸기, \t : tab, \w : 'w' 출력)

## ✓입력

```
input(prompt=None, /)
```

prompt : 쉘 모드에서 기본적으로 있는 '>>>', 기본은 지정하지 않음

기본 입력 자료형 : **문자열**

# 문자열

- ✓ 문자열 합치기 : +
- ✓ 문자열 반복 : '문자열' \* 숫자
- ✓ 문자열 길이 : len(문자열), 공백을 포함
- ✓ 형식 지정자:
- ✓ format() :  
    '문자열 {0} 문자열 {1} ...'.format(변수0, 변수1, ...)

형식 지정자	의미
%d	정수형
%f	실수형
%s	문자열
%%	%

# 예제(과제)

밑변과 높이 값을 받고 삼각형과 사각형의 넓이를 출력하시오.

단, 모든 숫자는 정수형으로 가정한다.

밑변 길이: 4  
높이: 7

입력 예시

삼각형 넓이: 14, 사각형 넓이: 28

출력 예시