

# 창직 IoT 종합설계입문

파이썬 (1)



## 파이썬 소개 (1)



Guido van Rossum



## 파이썬 소개 (2)

파이썬은 1991년 Guido van Rossum이 발표한 고급 프로그래밍 언어입니다.

“PEP 20 – The Zen of Python” 에서 파이썬의 철학을 확인할 수 있습니다.

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- ...

Google에서는 Google Python Style Guide를 통해 파이썬 코드 작성에 대한 가이드라인을 제시하고 있습니다.



## 파이썬 소개 (3)

---

### 특징

- 인터프리터 언어.
- 고수준의 자료형 때문에 복잡한 연산을 한 문장으로 표현 가능.
- 문장의 묶음은 괄호 대신 들여쓰기를 통해 이루어짐.
- 변수나 인자의 선언이 필요 없음.
- 확장 가능.
- 다양한 플랫폼에서 활용 가능하며 라이브러리(모듈)이 풍부.



## 파이썬 소개 (4)

---

### 참고 자료

#### 파이썬 자습서

- <https://docs.python.org/ko/3/tutorial/index.html>

#### 점프 투 파이썬

- <https://wikidocs.net/book/1>

#### 왕초보를 위한 Python

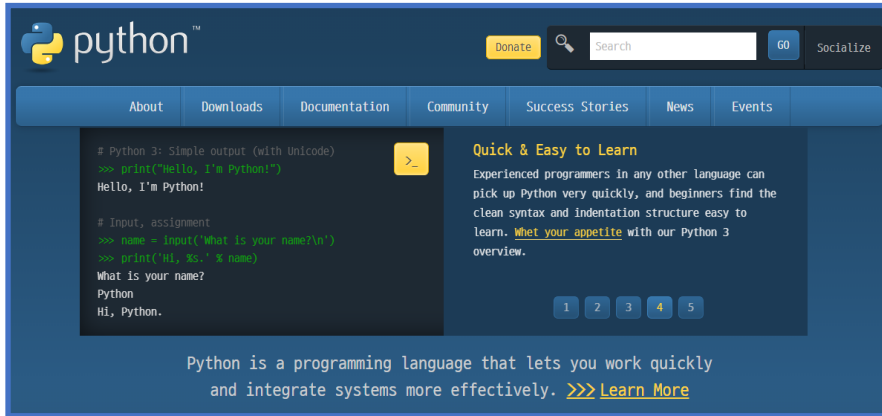
- <https://wikidocs.net/book/2>

이외에도 유튜브 등 동영상 강의 참고 가능



## 파이썬 설치 (1)

1. <https://www.python.org> 접속합니다.

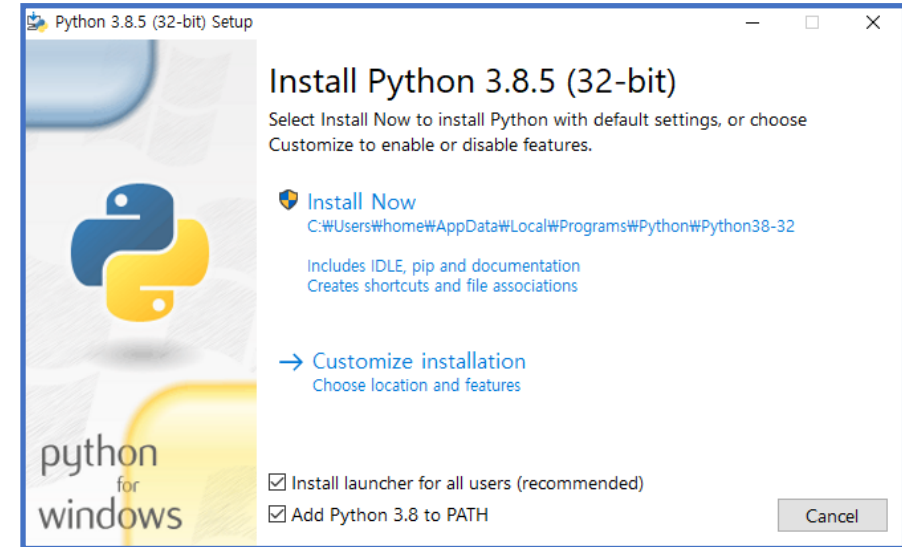


2. Downloads 탭을 클릭한 다음, Download Python 3.X.X 눌러 파이썬을 다운로드 합니다. 이 때, 자신이 사용하고 있는 OS에 맞는 버전을 다운로드 해야 합니다.



## 파이썬 설치 (2)

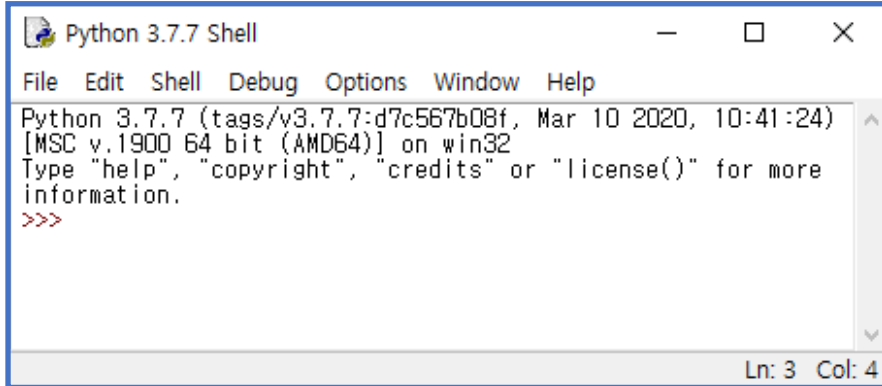
3. Add Python 3.X to PATH 항목 체크 후 Install Now 항목을 클릭하여 다운로드를 진행합니다.





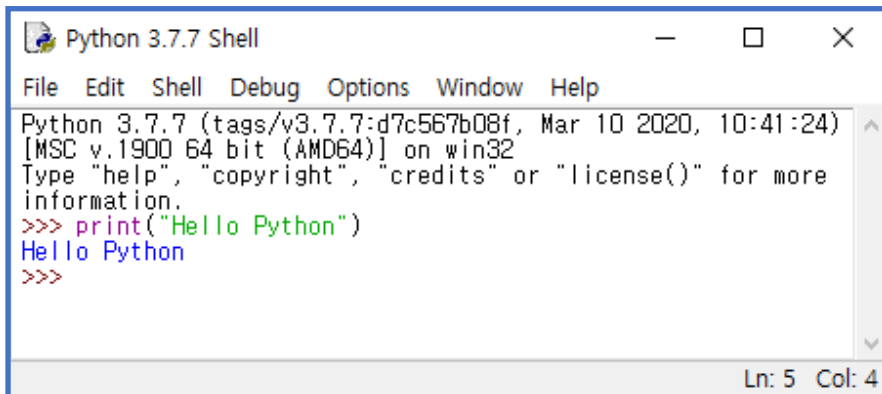
## 파이썬 설치 (3)

4. 설치가 완료되었다면 다음과 같이 IDLE 에디터가 설치되어 있는 것을 확인할 수 있습니다.



```
Python 3.7.7 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24)
[MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>>
```

5. `print( "Hello Python" )`을 입력한 뒤 결과를 확인해봅니다.



```
Python 3.7.7 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24)
[MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>> print("Hello Python")
Hello Python
>>>
```



## 파이썬 설치 (4)

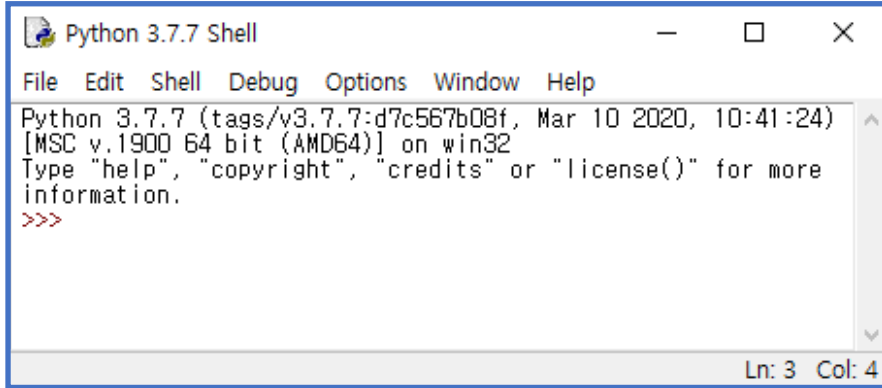
6. IDLE 에디터 이외에도 파이썬을 위한 다양한 통합개발환경(IDE, Integrated Development Environment)이 존재합니다.

- Jupyter Notebook
- PyCharm
- Spyder
- ...



## 셸

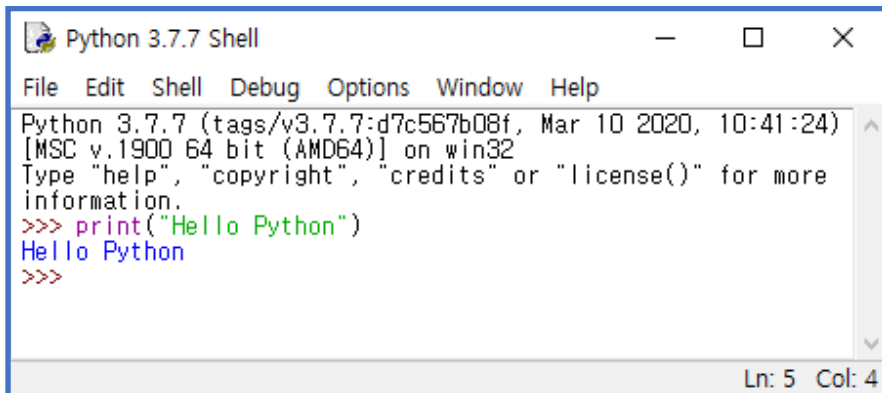
- 비교적 짧은 코드이거나 계산기 등으로 활용할 때, 셸을 이용하면 결과를 바로 확인할 수 있습니다.



```
Python 3.7.7 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24)
[MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>>
```

Ln: 3 Col: 4

- `print( "Hello Python" )`을 입력하면 셸에 Hello Python이 출력되는 것을 확인할 수 있습니다.



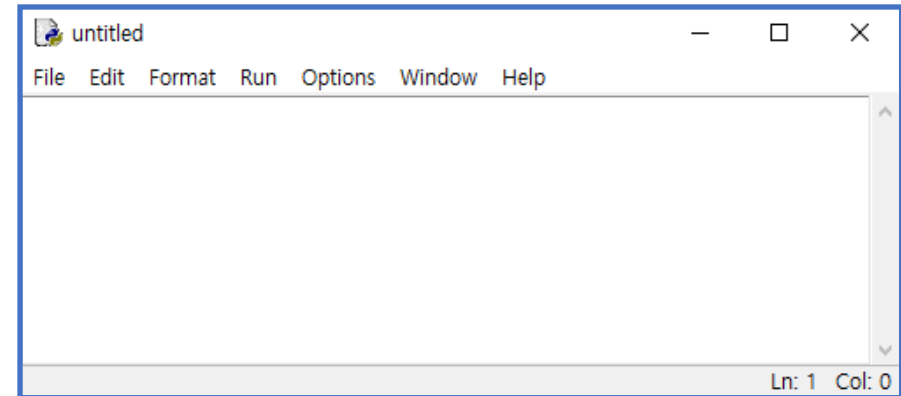
```
Python 3.7.7 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24)
[MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>> print("Hello Python")
Hello Python
>>>
```

Ln: 5 Col: 4



## 코드 편집기

- 셸에서 File → New File을 클릭하면 코드 편집기를 열 수 있습니다.
- 코드 편집기는 코드가 길어질 때, 사용합니다.

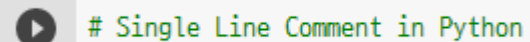


```
untitled
File Edit Format Run Options Window Help
```

Ln: 1 Col: 0

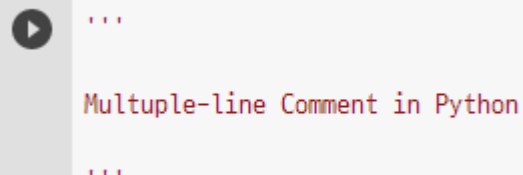
## 주석

- 한 줄 (행) 단위 주석은 #을 사용합니다.



```
# Single Line Comment in Python
```

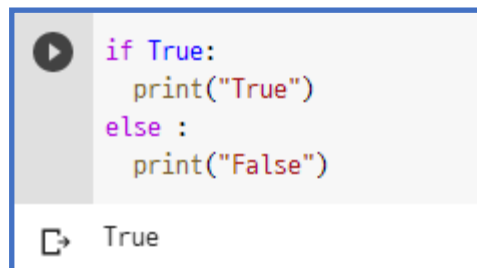
- 블록 단위 주석은 작은 따옴표를 3개 사용합니다.



```
'''  
Multiple-line Comment in Python  
'''
```

## 들여쓰기

- 파이썬은 코드 블록을 나누기 위한 { } 가 없습니다.
- 코드 블록은 들여쓰기를 통해 구분됩니다.



```
if True:  
    print("True")  
else :  
    print("False")
```

True

- 같은 코드 블록 내에서는 들여쓰기 칸 수가 같아야 합니다.



```
if True:  
    print("True")  
else :  
    print("False")
```

File "<ipython-input-9-bbd839fa6043>", line 2  
 print("True")  
 ^  
IndentationError: expected an indented block



## 행 결합 / 행 분리

---

### 행 결합 / 행 분리

- 파이썬은 C 언어와 다르게 구문의 종료 시 세미콜론이 필요하지 않습니다.
- 한 줄 (행)이 길어 여러 줄로 작성할 때는 역슬래시를 사용합니다.



```
print("Hello, Python! \nPython is a very simple language.")
```



```
Hello, Python! Python is a very simple language.
```



## 식별자

---

### 식별자

- 변수, 함수 또는 클래스 등을 구별하기 위한 이름입니다.

### 식별자 명명 규칙

- 문자, 숫자, 밑줄 기호를 사용할 수 있습니다.
- 대소문자를 구분합니다.
- 숫자로 시작하거나 공백을 포함할 수 없습니다.
- 밑줄 이외의 기호는 사용할 수 없습니다.
- 예약어는 사용할 수 없습니다.





# 예약어

## 예약어

- 예약어는 파이썬 프로그램 내에 이미 기능이 존재하는 문자열입니다.

and	def	exec
assert	del	finally
break	elif	for
class	else	from
continue	except	global
if	not	return
import	or	try
in	pass	while
is	print	with
lambda	raise	yield



# 파이썬 입출력 함수

## 출력 함수 print

- print 함수를 통해 출력할 수 있습니다.

```
print("Hello Python")
```

```
Hello Python
```

## 입력 함수 input

- input 함수를 통해 입력할 수 있습니다.

```
input("Enter anything : ")
```

```
Enter anything : Hello Python
```

```
'Hello Python'
```



# 연산자

## 연산자의 종류

파이썬 연산자는 크게 다음과 같이 분류할 수 있습니다.

- 산술 연산자 (Arithmetic Operator)
- 할당 연산자 (Assignment Operator)
- 비교 연산자 (Comparison Operator)
- 논리 연산자 (Logical Operator)
- 비트 연산자 (Bitwise Operator)
- 멤버 연산자 (Membership Operator)
- 식별 연산자 (Identity Operator)



# 산술 연산자

## 산술 연산자 (Arithmetic Operator)

- 연산자의 우선 순위는 일반적인 사칙 연산과 같습니다.

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

- 계산 결과는 아래와 같습니다.

```
print( "2 + 3 = ", 2 + 3) # 덧셈
print( "2 - 3 = ", 2 - 3) # 뺄셈
print( "2 * 3 = ", 2 * 3) # 곱셈
print("2 ** 3 = ", 2 ** 3) # 거듭제곱
print( "2 / 3 = ", 2 / 3) # 나눗셈
print("2 // 3 = ", 1 // 3) # 몫
print( "2 % 3 = ", 1 % 3) # 나머지
```

```
2 + 3 = 5
2 - 3 = -1
2 * 3 = 6
2 ** 3 = 8
2 / 3 = 0.6666666666666666
2 // 3 = 0
2 % 3 = 1
```