

# Python 기본 강의

멋쟁이사자처럼 11기 운영진 강경아

# Python 다운로드

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

# Python 다운로드

The screenshot shows the Python.org website with the 'Downloads' menu highlighted. The browser's address bar shows 'python.org'. The website's navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this, a secondary navigation bar contains links for About, Downloads (highlighted), Documentation, Community, Success Stories, News, and Events. The 'Downloads' dropdown menu is open, showing options: All releases, Source code, Windows, macOS, Other Platforms, License, and Alternative Implementations. The 'Windows' option is selected, leading to the 'Download for Windows' section. This section features a button for 'Python 3.11.2' and a note stating: 'Note that Python 3.9+ cannot be used on Windows 7 or earlier. Not the OS you are looking for? Python can be used on many operating systems and environments. View the full list of downloads.' The footer of the website includes links for 'Get Started', 'Download', 'Docs', and 'Jobs'.

Welcome to Python.org x 캡처 단축키 - Google Search x +

python.org

번역 출석 개발 사회 라틴어 독일어 기타 북마크

Python PSF Docs PyPI Jobs Community

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

All releases  
Source code  
Windows  
macOS  
Other Platforms  
License  
Alternative Implementations

**Download for Windows**

Python 3.11.2

**Note that Python 3.9+ cannot be used on Windows 7 or earlier.**

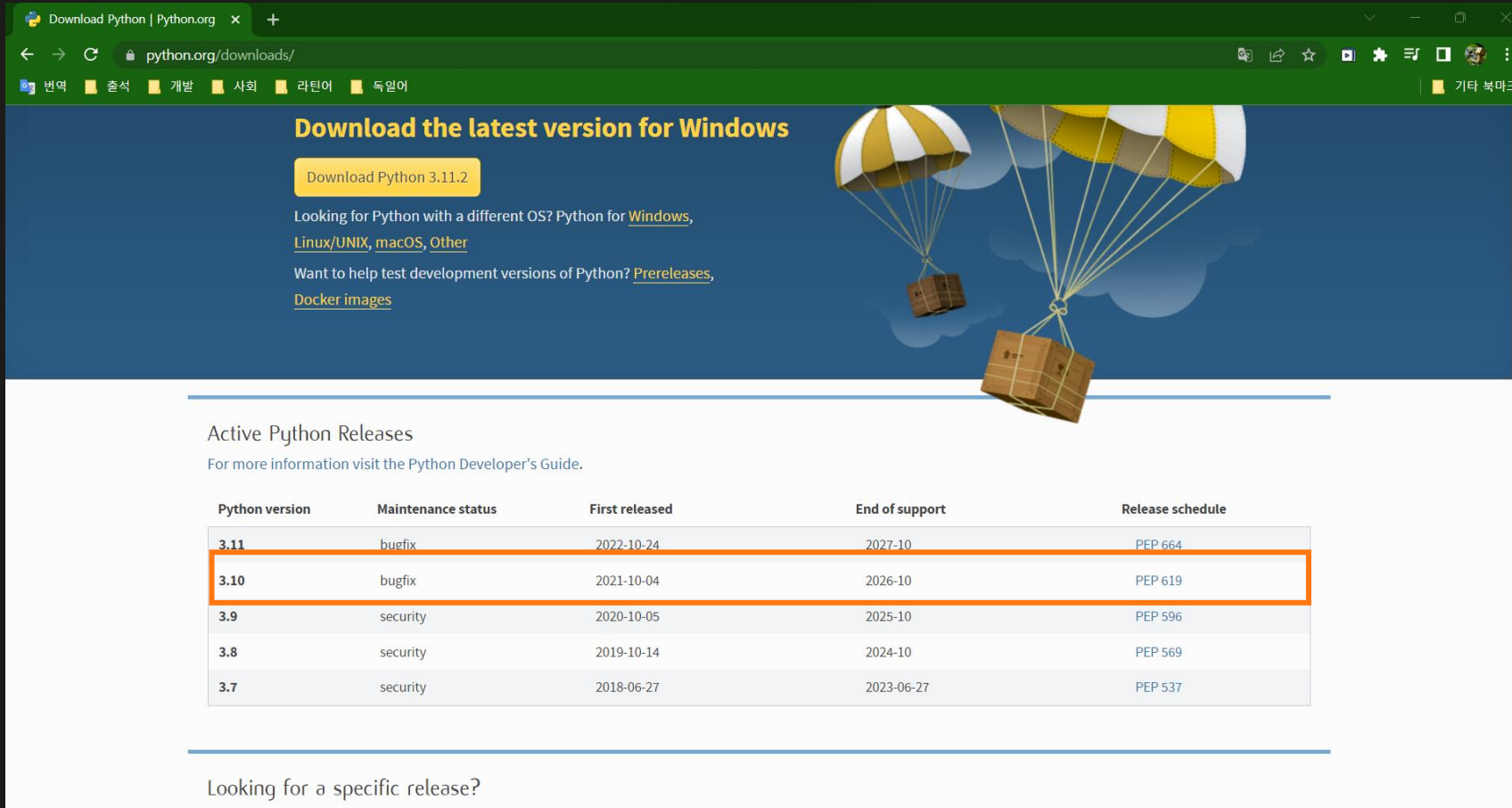
Not the OS you are looking for? Python can be used on many operating systems and environments. View the full list of downloads.

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

Get Started Download Docs Jobs

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

# Python 다운로드

A screenshot of the Python.org download page. The browser window shows the URL 'python.org/downloads/'. The page has a dark blue header with the text 'Download the latest version for Windows' and a yellow button 'Download Python 3.11.2'. Below this, there are links for 'Looking for Python with a different OS?' and 'Want to help test development versions of Python?'. The main content area is white and features a section titled 'Active Python Releases' with a table of release information. The table has five columns: 'Python version', 'Maintenance status', 'First released', 'End of support', and 'Release schedule'. The rows show versions 3.11, 3.10, 3.9, 3.8, and 3.7. The row for version 3.10 is highlighted with an orange border. At the bottom, there is a link 'Looking for a specific release?'.

Download Python | Python.org

python.org/downloads/

Download the latest version for Windows

Download Python 3.11.2

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)

Want to help test development versions of Python? [Prereleases](#), [Docker images](#)

### Active Python Releases

For more information visit the [Python Developer's Guide](#).

Python version	Maintenance status	First released	End of support	Release schedule
3.11	bugfix	2022-10-24	2027-10	PEP 664
3.10	bugfix	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569
3.7	security	2018-06-27	2023-06-27	PEP 537

[Looking for a specific release?](#)

# Python 확장 프로그램 설치

The screenshot shows the Visual Studio Code interface with the Python extension being installed. The left sidebar shows the 'Extensions' view with 'python' selected. The main area displays the 'Python' extension by Microsoft, version v2023.2.0, with 77,234,063 downloads and 527 ratings. The extension description mentions IntelliSense (Pylance), Linting, Debugging, Jupyter Notebooks, code formatting, refactoring, unit tests, and more. A large orange text overlay reads '디버깅, 자동완성 등을 지원.' (Supports debugging, auto-completion, etc.). The 'Python extension for Visual Studio Code' section describes the extension's features and support for `vscode.dev`. The 'Installed extensions' section states that the Python extension will automatically install the Pylance, Jupyter, and isort extensions. The 'Quick start' section includes a step to install a supported version of Python on the system.

python

Python v2023.2.0  
Microsoft 77,234,063 ★★★★★ (527)  
IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Notebooks, code formatting, refactoring, unit tests, and more.

사용 안 함 | 제거 | 시험판 버전으로 전환

이 확장은 전역적으로 사용하도록 설정되었습니다.

디버깅, 자동완성 등을 지원.

세부 정보 기능 기여도 변경 로그 확장 팩 런타임 상태

### Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python language (for all actively supported versions of the language: >=3.7), including features such as IntelliSense (Pylance), linting, debugging, code navigation, code formatting, refactoring, variable explorer, test explorer, and more!

### Support for `vscode.dev`

The Python extension does offer some support when running on `vscode.dev` (which includes `github.dev`). This includes partial IntelliSense for open files in the editor.

### Installed extensions

The Python extension will automatically install the Pylance, Jupyter and isort extensions to give you the best experience when working with Python files and Jupyter notebooks. However, Pylance is an optional dependency, meaning the Python extension will remain fully functional if it fails to be installed. You can also uninstall it at the expense of some features if you're using a different language server.

### Quick start

- Step 1. Install a supported version of Python on your system (note: that the system install of Python on macOS is not

범주  
Programming Languages Debuggers  
Linters Formatters Other  
Data Science Machine Learning  
Notebooks

확장 리소스  
마켓플레이스  
저장소  
라이선스  
Microsoft

더 많은 정보  
게시됨 2016. 1. 20. 0시 3분 11초  
마지막으로 릴리 2023. 2. 15. 19시 39분 14초  
스팀  
마지막으로 업데이트 2023. 2. 3. 13시 19분 34초  
이트함 날짜  
식별자 ms-python.python

# print문

- 입력
  - `print( "hello world " )`
- 출력
  - `hello world`

# 변수

- 변수는 데이터를 할당해서 사용할 수 있는 공간
- 파이썬의 경우 \_ 대신 대문자를 활용하자(**Camel Case**)
  - 예) **likeLion**

# 변수

입력

**num=42**

**char="hello world "**

**tf=True**

**print(num)**

**print(char)**

**print(tf)**

출력

**42**

**hello world**

**True** # 앞글자가 대문자!



# 리스트

- 변수에 여러 개의 값을 지정하고 싶을 때 사용.
- 다른 언어에서는 배열이라고 부르기도 합니다.
- 이외에도 튜플, 딕셔너리 등의 자료형이 있습니다.

예시)

입력

```
>list = [ 42, "hello world ", True]
```

# list

입력

```
list = [ 42, "hello world ",  
True]  
print(list)
```

출력

```
[ 42, "hello world ", True]
```

# input

- 값을 받는 함수
- **input( “할 말” )**
- 입력하라는 표시의 커서가 나옵니다.

# input

입력

```
input( "값을 입력해주세요")  
value = input("값을 입력해주세요")  
print(value)
```

터미널(42를 입력한 경우)

```
>값을 입력해주세요 42  
>42
```

# 조건문

**if (조건1):**

    (조건1이 참이라면 할 일)

**elif (조건2):**

    (조건1은 참이 아니고 조건2가 참이라면 할 일)

**else:**

    (조건1, 조건2가 참이 아니면 할 일)

# 조건문

## 입력

```
dinner = input( “식사했나요?(y/n)” )  
if dinner == y:  
    print("맛있게 드셨군요!")  
else:  
    print( “식사하세요.” )
```

## 출력

**y**라고 답할 경우  
➤맛있게 드셨군요! 출력

**n**라고 답할 경우  
➤식사하세요. 출력

# 모듈

- 파일 상단에 **import** (불러올 것) **from** (파일 이름)
- 파이썬 내장 함수인 **random**의 경우 **import random**만 적으면 끝!

# 반복문(for)

**for i in range(시작, 끝):**  
(시작~끝-1만큼 반복할 것)

**for i in range(n):**  
(n만큼 반복할 것)

```
3 i = 0
4 sum = 0
5
6 # for i in range(1, 101):
7 #     print(i)
8
9 for i in range(1, 101):
10     sum = sum + i
11
12 print(sum)
```



1부터 100까지 합을 구하는 것



# 반복문(while)

- **while** 조건문:  
(조건이 참이라면 실행할 코드)
- 예시  
    **while True:**  
        **print( “멋쟁이사자처럼” )**  
        **break**  
=> 멋쟁이사자처럼 한 번 출력!

# 뭐 먹을까?

간단하게 만들어보아요!

# 음식을 리스트에 입력

```
1  import random 파이썬 내장 함수인 random 함수 호출!
2
3  ## 음식을 리스트에 입력하는 부분
4  food = []
5  while True:
6      ans = input("추가할 음식을 입력해주세요. 종료할 경우 stop을 입력해주세요.")
7      if ans=="stop": 답이 stop이면 while문을 빠져나옵니다.
8          break
9      else:
10         food.append(ans) food라는 리스트에 답을 추가합니다.
11         print(food, "가 입력되었습니다.")
```

# 뭘 먹을지 뽑아보자

```
##음식 중에서 하나 고르는 부분
num1 = len(food)-1  리스트는 0부터 시작하기 때문에 -1을 해줍니다.
num2 = input("몇 번 돌릴까요?")
for i in range(int(num2)): num2에 담긴 정수만큼 반복해줍니다.
    roll = random.randrange(0,num1) 0부터 num1사이의 숫자 중 하나를 임의로 뽑습니다.
    print("랜덤으로 먹을 것 추천!", food[roll])
```

- **len** 함수: 리스트의 개수를 구하는 데 사용됩니다.

# 실행시켜봅시다!

- 터미널 창에 **python (파일명)** 입력
- 실행모습

```
문제  출력 디버그 콘솔 터미널

● PS C:\경상대\멋쟁이사자처럼\교육자료\파이썬 기초실습> python test.py
추가할 음식을 입력해주세요. 종료할 경우 stop을 입력해주세요.치킨
['치킨'] 가 입력되었습니다.
추가할 음식을 입력해주세요. 종료할 경우 stop을 입력해주세요.피자
['치킨', '피자'] 가 입력되었습니다.
추가할 음식을 입력해주세요. 종료할 경우 stop을 입력해주세요.족발
['치킨', '피자', '족발'] 가 입력되었습니다.
추가할 음식을 입력해주세요. 종료할 경우 stop을 입력해주세요.삼겹살
['치킨', '피자', '족발', '삼겹살'] 가 입력되었습니다.
추가할 음식을 입력해주세요. 종료할 경우 stop을 입력해주세요.stop
○ 몇 번 돌릴까요?2
랜덤으로 먹을 것 추천! 족발
랜덤으로 먹을 것 추천! 피자
```

# Crawling

크롤링의 기본

# 준비운동

1. 터미널 창에 **pip install requests** 작성

# 200, 서버와의 연결이 잘 됐어요 :)

## • 입력

```
import requests

url = "http://naver.com"
response = requests.get(url)
print(response.status_code)
```

## • 터미널

```
>200
```



# 기본적인 크롤링 해보기

- 입력

```
import requests

url = "http://naver.com"
response = requests.get(url)
print(response.status_code)
print(response.text)
```

- 터미널

**Just do it!**

# 준비운동 2

- 터미널 창에 **pip install bs4** 입력

# 특정 태그를 가져와보자!

## • 입력

```
import requests
from bs4 import BeautifulSoup
url = "http://naver.com"
response = requests.get(url)
soup = BeautifulSoup(response.content,
"html.parser")
print(soup.h1)
```

## • 터미널

```
><h1
class="logo_default">
<a class="logo_naver"
data-cik="top.logo"
href="/"><span
class="blind">네이버
</span></a>
</h1>
```

**Q.** 태그를 빼고 가져오려면 어떻게 해야 할까요?

# 해당하는 모든 태그를 가져와보자!

## • 입력

```
import requests
from bs4 import BeautifulSoup

url = "http://naver.com"
response = requests.get(url)

soup = BeautifulSoup(response.content,
"html.parser")
print(soup.find_all("h2"))
```

## • 터미널

```
>[<h2
class="blind">뉴스
스탠드</h2>, <h2
class="blind">주제
별 캐스트</h2>, <h2
class="blind">Sig
n in</h2>, <h2
class="blind">타임
스퀘어</h2>]
```

# 잠깐! 오류가 나요

- 404가 뜨는 경우
  - 사이트에서 사용자를 부적절한 봇이라고 인식해서 차단을 막는 경우가 있어요.
- 해결방안
  1. <https://www.useragentstring.com/>에 들어간다.



# 잠깐! 오류가 나요

## 2. 코드를 다음과 같이 수정해줍니다.

### • 입력

```
import requests
from bs4 import BeautifulSoup

url = "http://naver.com"
headers = {'User-Agent': '(복사한 user agent를 이곳에 붙여넣기)'}

response = requests.get(url=url, headers=headers)

soup = BeautifulSoup(response.content, "html.parser")
print(soup.find_all("h2"))
```

### • 터미널

>