

Lecture 04

Understanding the Library & Directory Management

- Library
- Python Structure
- Directory Management

Python » Korean » 3.8.11 » Documentation »

내려받기

이 문서 내려받기

버전별 설명서

Python 3.11 (in development)
Python 3.10 (pre-release)
Python 3.9 (stable)
Python 3.8 (security-fixes)
Python 3.7 (security-fixes)
Python 3.6 (security-fixes)
Python 3.5 (EOL)
Python 2.7 (EOL)
[모든 버전](#)

기타 자원

PEP 색인
초보자 가이드
도서 목록
오디오/비디오 토크
파이썬 개발자 지침서

Python 3.8.11 문서

환영합니다! 파이썬 3.8.11 설명서의 한국어 번역입니다.

설명서의 파트들:

파이썬 3.8의 새로운 기능은?

2.0 이후의 모든 "새로운 기능" 문서

자습서

여기에서 시작하세요

라이브러리 레퍼런스

베개 밑에 넣어 두세요

언어 레퍼런스

문법과 언어 요소들을 설명합니다

파이썬 설정 및 사용법

여러 플랫폼에서 파이썬을 사용하는 법

파이썬 HOWTO

특정 주제에 대한 심층적인 문서

파이썬 모듈 설치하기

파이썬 패키지 색인 및 기타 소스에서 설치하기

파이썬 모듈 배포하기

다른 사람들이 설치할 수 있도록 모듈을 게시하기

확장 및 내장

C/C++ 프로그래머를 위한 자습서

파이썬/C API

C/C++ 프로그래머를 위한 레퍼런스

FAQs

자주 나오는 질문들 (답도 있습니다!)

Python Library

1. Built-in function
 - 1.1 abs() / sum()
2. External Functions and List
 - 2.1 os.getcwd()
 - 2.2 sys.path
3. Python Structure
 - 3.1 Package
 - 3.2 Function
 - 3.3 Module
4. Directory Management
 - 4.1 getcwd() / chdir()
 - 4.2 listdir()
 - 4.3 os.path.exists()
 - 4.4 mkdir()
 - 4.5 makedirs()
 - 4.6 rmdir()
 - 4.7 removedirs()

Python » Korean » 3.8.6rc1 » Documentation »

<https://docs.python.org/ko/3.8/tutorial/index.html>

이전 항목

Changelog

다음 항목

1. 입맛 돋우기

현재 문서

버그 보고하기
소스 보기

파이썬 자습서

파이썬은 배우기 쉽고, 강력한 프로그래밍 언어입니다. 효율적인 자료 구조들과 객체 지향 프로그래밍에 대해 간단하고도 효과적인 접근법을 제공합니다. 우아한 문법과 동적 타이핑(dynamic typing)은, 인터프리터 적인 특징들과 더불어, 대부분 플랫폼과 다양한 문제 영역에서 스크립트 작성과 빠른 응용 프로그램 개발에 이상적인 환경을 제공합니다.

파이썬 인터프리터와 풍부한 표준 라이브러리는 소스나 바이너리 형태로 파이썬 웹 사이트, <https://www.python.org/>, 에서 무료로 제공되고, 자유롭게 배포할 수 있습니다. 같은 사이트는 제삼자들이 무료로 제공하는 확장 모듈, 프로그램, 도구, 문서들의 배포판이나 링크를 포함합니다.

파이썬 인터프리터는 C 나 C++ (또는 C에서 호출 가능한 다른 언어들)로 구현된 새 함수나 자료 구조를 쉽게 추가할 수 있습니다. 파이썬은 고객화 가능한 응용 프로그램을 위한 확장 언어로도 적합합니다.

이 학습서는 파이썬 언어와 **1** 기본 개념과 기능들을 격식 없이 소개합니다. 파이썬 인터프리터를 직접 만져볼 수 있도록 돕지만, 모든 독립적이기 때문에 오프라인에서 읽기에도 적합합니다.

표준 객체들과 모듈들에 대한 설명은 **파이썬 표준 라이브러리** 를 보세요. 파이썬 언어 레퍼런스는 언어에 대한 좀더 형식적인 정의를 제공합니다. C 나 C++ 로 확장하려면 **파이썬 인터프리터 확장 및 내장** 와 **파이썬/C API 레퍼런스 설명서** 를 읽으세요. 파이썬을 깊이 있게 다룬 책들도 많습니다.

이 자습서는 포괄적이려고 시도하지 않습니다. 모든 기능을 다루지는 않는데, 심지어 자주 사용되는 기능조차도 그렇습니다. 대신에, 파이썬의 가장 주목할만한 기능들을 소개하고, 언어의 맛과 스타일에 대한 전체적인 인상을 제공합니다. 이 학습서를 읽은 후에는 파이썬 모듈과 프로그램을 작성할 수 있고 **파이썬 표준 라이브러리** 에 기술된 다양한 파이썬 라이브러리 모듈들에 대해 학습할 수 있는 준비가 될 것입니다.

용어집 또한 훑어볼 만한 가치가 있습니다.

- 1. 입맛 돋우기
- 2. 파이썬 인터프리터 사용하기
 - 2.1. 인터프리터 실행하기
 - 2.1.1. 인자 전달
 - 2.1.2. 대화형 모드
 - 2.2. 인터프리터와 환경

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

Python » Korean 3.8.6rc1 Documentation »

<https://docs.python.org/ko/3.8/library/index.html#library-index>

이전 항목

10. 전체 문법 규칙

다음 항목

소개

현재 문서

버그 보고하기

소스 보기

파이썬 표준 라이브러리

파이썬 언어 레퍼런스는 파이썬 언어의 정확한 문법과 의미를 설명하고 있지만, 이 라이브러리 레퍼런스 설명서는 파이썬과 함께 배포되는 표준 라이브러리를 설명합니다. 또한, 파이썬 배포판에 일반적으로 포함되어있는 선택적 구성 요소 중 일부를 설명합니다.

파이썬의 표준 라이브러리는 매우 광범위하며, 아래 나열된 긴 목록에 표시된 대로 다양한 기능을 제공합니다. 라이브러리에는 일상적인 프로그래밍에서 발생하는 많은 문제에 대한 표준적인 해결책을 제공하는 파이썬으로 작성된 모듈뿐만 아니라, 파일 I/O와 같은 시스템 기능에 액세스하는 (C로 작성된) 내장 모듈들이 포함됩니다 (이 모듈들이 없다면 파이썬 프로그래머가 액세스할 방법은 없습니다). 이 모듈 중 일부는 플랫폼 관련 사항을 플랫폼 중립적인 API들로 추상화시킴으로써, 파이썬 프로그램의 이식성을 권장하고 개선하도록 명시적으로 설계되었습니다.

윈도우 플랫폼용 파이썬 설치 프로그램은 일반적으로 전체 표준 라이브러리를 포함하며 종종 많은 추가 구성 요소도 포함합니다. 유닉스와 같은 운영체제의 경우, 파이썬은 일반적으로 패키지 모음으로 제공되기 때문에, 운영체제와 함께 제공되는 패키지 도구를 사용하여 선택적 구성 요소의 일부 또는 전부를 구해야 할 수 있습니다.

표준 라이브러리 외에도, 수천 가지 컴포넌트(개별 프로그램과 모듈부터 패키지 및 전체 응용 프로그램 개발 프레임워크까지)가 늘어나고 있는데, [파이썬 패키지 색인](#) 에서 얻을 수 있습니다.

2

- 소개
 - 가용성에 대한 참고 사항
- **내장 함수**
- 내장 상수
 - site 모듈에 의해 추가된 상수들
- 내장형
 - 논리값 검사
 - 논리 연산 — and, or, not
 - 비교
 - 숫자 형 — int, float, complex
 - 이터레이터 형
 - 시퀀스 형 — list, tuple, range

Python Library

1. Built-in function

1.1 `abs()` / `sum()`

2. External Functions and List

2.1 `os.getcwd()`

2.2 `sys.path`

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 `getcwd()` / `chdir()`

4.2 `listdir()`

4.3 `os.path.exists()`

4.4 `mkdir()`

4.5 `makedirs()`

4.6 `rmdir()`

4.7 `removedirs()`

Python » Korean » 3.8.6rc1 » Documentation » 파이썬 표준 라이브러리 »

<https://docs.python.org/ko/3.8/library/functions.html>

이전 항목

소개

다음 항목

내장 상수

현재 문서

버그 보고하기
소스 보기

내장 함수

파이썬 인터프리터에는 항상 사용할 수 있는 많은 함수와 형이 내장되어 있습니다. 여기에서 알파벳 순으로 나열합니다.

내장 함수				
3 <code>abs()</code>	<code>delattr()</code>	<code>hash()</code>	<code>memoryview()</code>	<code>set()</code>
<code>all()</code>	<code>dict()</code>	<code>help()</code>	<code>min()</code>	<code>setattr()</code>
<code>any()</code>	<code>dir()</code>	<code>hex()</code>	<code>next()</code>	<code>slice()</code>
<code>ascii()</code>	<code>divmod()</code>	<code>id()</code>	<code>object()</code>	<code>sorted()</code>
<code>bin()</code>	<code>enumerate()</code>	<code>input()</code>	<code>oct()</code>	<code>staticmethod()</code>
<code>bool()</code>	<code>eval()</code>	<code>int()</code>	<code>open()</code>	<code>str()</code>
<code>breakpoint()</code>	<code>exec()</code>	<code>isinstance()</code>	<code>ord()</code>	4 <code>sum()</code>
<code>bytearray()</code>	<code>filter()</code>	<code>issubclass()</code>	<code>pow()</code>	<code>super()</code>
<code>bytes()</code>	<code>float()</code>	<code>iter()</code>	<code>print()</code>	<code>tuple()</code>
<code>callable()</code>	<code>format()</code>	<code>len()</code>	<code>property()</code>	<code>type()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>list()</code>	<code>range()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>locals()</code>	<code>repr()</code>	<code>zip()</code>
<code>compile()</code>	<code>globals()</code>	<code>map()</code>	<code>reversed()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hasattr()</code>	<code>max()</code>	<code>round()</code>	

`abs(x)`

숫자의 절댓값을 돌려줍니다. 인자는 정수 또는 실수입니다. 인자가 복소수면 그 크기가 반환됩니다. `x`가 `__abs__()`를 정의하면, `abs(x)`는 `x.__abs__()`를 반환합니다.

Python Library

1. Built-in function

1.1 `abs()` / `sum()`

2. External Functions and List

2.1 `os.getcwd()`

2.2 `sys.path`

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 `getcwd()` / `chdir()`

4.2 `listdir()`

4.3 `os.path.exists()`

4.4 `mkdir()`

4.5 `makedirs()`

4.6 `rmdir()`

4.7 `removedirs()`

<https://docs.python.org/ko/3.8/library/functions.html#abs>

`abs(x)`

숫자의 절댓값을 돌려줍니다. 인자는 정수 또는 실수입니다. 인자가 복소수면 그 크기가 반환됩니다. `x`가 `__abs__()`를 정의하면, `abs(x)`는 `x.__abs__()`를 반환합니다.

<https://docs.python.org/ko/3.8/library/functions.html#sum>

`sum(iterable, /, start=0)`

`start` 및 `iterable`의 항목들을 왼쪽에서 오른쪽으로 합하고 합계를 돌려줍니다. `iterable`의 항목은 일반적으로 숫자며 시작 값은 문자열이 될 수 없습니다.

어떤 경우에는 `sum()`에 대한 좋은 대안이 있습니다. 문자열의 시퀀스를 연결하는 가장 선호되고 빠른 방법은 `''.join(sequence)`를 호출하는 것입니다. 확장된 정밀도로 부동 소수점 값을 더하려면 `math.fsum()`를 보세요. 일련의 이터러블들을 연결하려면 `itertools.chain()`를 고려해보세요.

버전 3.8에서 변경: `start` 매개 변수는 키워드 인자로만 지정될 수 있습니다.

Python Library

1. Built-in function

1.1 `abs()` / `sum()`

2. External Functions and List

2.1 `os.getcwd()`

2.2 `sys.path`

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 `getcwd()` / `chdir()`

4.2 `listdir()`

4.3 `os.path.exists()`

4.4 `mkdir()`

4.5 `makedirs()`

4.6 `rmdir()`

4.7 `removedirs()`

표준 라이브러리 - 내장함수

■ `abs()`, `sum()`

```
1 a=-23.5
```

```
1 a
```

-23.5

```
1 abs(a)
```

23.5

```
1 sum([1,2,3,4,5,6,7,8,9,10])
```

55

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

■ getcwd()

표준 라이브러리 – 내장함수에 해당되지 않는 함수

```
1 getcwd()
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-8-80c72379bb65> in <module>  
----> 1 getcwd()
```

```
NameError: name 'getcwd' is not defined
```

<https://docs.python.org/ko/3.8/genindex.html>

Python » Korean » 3.8.6rc1 » Documentation » 파이썬 표준 라이브러리 »

빠른 검색

이동

이전 | 다음 | 모듈 색인

색인

알파벳별 색인:

기호 | _ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | [| \ |] | ^ | _ | ` | { | | } | ~ |

<https://docs.python.org/ko/3.8/genindex-G.html>

Python Library

1. Built-in function
1.1 abs() / sum()

2. External Functions and List
2.1 os.getcwd()
2.2 sys.path

3. Python Structure
3.1 Package
3.2 Function
3.3 Module

4. Directory Management
4.1 getcwd() / chdir()
4.2 listdir()
4.3 os.path.exists()
4.4 mkdir()
4.5 makedirs()
4.6 rmdir()
4.7 removedirs()

■ getcwd()

Python » Korean » 3.8.6rc1 » Documentation »

getcwd

1/3

^ v x

이동 | 모듈 | 색인

Index – G

G.722
gaierror
gamma() (math 모듈)
gammavariate() (random 모듈)
garbage (gc 모듈)
garbage collection
(가비지 수거)
gather() (curses.textpad.Textbox 메서드)
gauss() (random 모듈)
gc (모듈)
gc_collect() (test.support 모듈)
gcd() (fractions 모듈)
(math 모듈)
ge() (operator 모듈)
gen_lib_options() (distutils.compiler 모듈)
gen_preprocess_options() (distutils.compiler 모듈)
gen_uuid() (msilib 모듈)
generate_help() (distutils.fancy_getopt.FancyGetopt 메서드)
generate_tokens() (tokenize 모듈)
generator

getclasstree() (inspect 모듈)
getclosurevars() (inspect 모듈)
GetColumnInfo() (msilib.View 메서드)
getColumnNumber() (xml.sax.xmlreader.Locator 메서드)
getcomments() (inspect 모듈)
getcompname() (aifc.aifc 메서드)
(sunau.AU_read 메서드)
(wave.Wave_read 메서드)
getcomptype() (aifc.aifc 메서드)
(sunau.AU_read 메서드)
(wave.Wave_read 메서드)
getContentHandler()
(xml.sax.xmlreader.XMLReader 메서드)
getcontext() (decimal 모듈)
getcoroutinelocals() (inspect 모듈)
getcoroutinestate() (inspect 모듈)
getctime() (os.path 모듈)
getcwd() (os 모듈)
getcwdb() (os 모듈)
getcwdu (2to3 fixer)

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

■ getcwd()

current working directory

<https://docs.python.org/ko/3.8/library/os.html#os.getcwd>

표준 라이브러리 – 내장함수에 해당되지 않는 함수

getcwd() – os 모듈을 임포트한 후 사용가능한 함수

```
os.getcwd()
```

현재 작업 디렉터리를 나타내는 문자열을 반환합니다.

Python Library

1. Built-in function

1.1 `abs()` / `sum()`

2. External Functions and List

2.1 `os.getcwd()`

2.2 `sys.path`

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 `getcwd()` / `chdir()`

4.2 `listdir()`

4.3 `os.path.exists()`

4.4 `mkdir()`

4.5 `makedirs()`

4.6 `rmdir()`

4.7 `removedirs()`

■ OS

색인

알파벳별 색인:

기호 | _ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | **O** | P | Q | R | S | T | U | V | W | X | Y | Z | ^ | _ | ` | ~ |

<https://docs.python.org/ko/3.8/genindex.html>

<https://docs.python.org/ko/3.8/genindex-O.html>

`open()` (aifc 모듈)
(bz2 모듈)
(codecs 모듈)
(dbm 모듈)
(dbm.dumb 모듈)
(dbm.gnu 모듈)
(dbm.ndbm 모듈)
(distutils.text_file.TextFile 메서드)
(gzip 모듈)
(imaplib.IMAP4 메서드)
(io 모듈)
(lzma 모듈)
(os 모듈)
(ossaudiodev 모듈)
(pathlib.Path 메서드)

`OrderedDict` (collections 클래스)
(typing 클래스)
`origin` (importlib.machinery.ModuleSpec의 속성)
`origin_req_host` (urllib.request.Request의 속성)
`origin_server` (wsgiref.handlers.BaseHandler의 속성)
os
os 모듈
os (모듈)
`os.path` (모듈)
`os.environ` (wsgiref.handlers.BaseHandler의 속성)
`OSError`
`ossaudiodev` (모듈)
`OSSAudioError`
`outfile`
json.tool command line option

Python Library

1. Built-in function

1.1 `abs()` / `sum()`

2. External Functions and List

2.1 `os.getcwd()`

2.2 `sys.path`

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 `getcwd()` / `chdir()`

4.2 `listdir()`

4.3 `os.path.exists()`

4.4 `mkdir()`

4.5 `makedirs()`

4.6 `rmdir()`

4.7 `removedirs()`

■ OS

<https://docs.python.org/ko/3.8/library/os.html#module-os>

표준 라이브러리 – 모듈

os — 기타 운영 체제 인터페이스

소스 코드: [Lib/os.py](#)

이 모듈은 운영 체제 종속 기능을 사용하는 이식성 있는 방법을 제공합니다. 파일을 읽거나 쓰고 싶으면 `open()` 을 보세요, 경로를 조작하려면 `os.path` 모듈을 보시고, 명령 줄에서 주어진 모든 파일의 모든 줄을 읽으려면 `fileinput` 모듈을 보십시오. 임시 파일과 디렉터리를 만들려면 `tempfile` 모듈을 보시고, 고수준의 파일과 디렉터리 처리는 `shutil` 모듈을 보십시오.

Python Library

1. Built-in function
1.1 abs() / sum()

2. External Functions and List
2.1 os.getcwd()
2.2 sys.path

3. Python Structure
3.1 Package
3.2 Function
3.3 Module

4. Directory Management
4.1 getcwd() / chdir()
4.2 listdir()
4.3 os.path.exists()
4.4 mkdir()
4.5 makedirs()
4.6 rmdir()
4.7 removedirs()

■ getcwd()

실습

```
1 import os
2 os.getcwd()
```

'D:\\\\ai'

```
1 os.system('mkdir today')
2 os.getcwd()
```

'D:\\\\ai'

```
1 os.chdir('today')
2 os.getcwd()
```

'D:\\\\ai\\\\today'

```
1 os.chdir('c:/temp')
2 os.getcwd()
```

'c:\\\\temp'

경로 구분
/ 또는 \\ (\\)

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

■ getcwd()

실습

```
1 type(os)
```

module

```
1 type(os.getcwd)
```

builtin_function_or_method

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

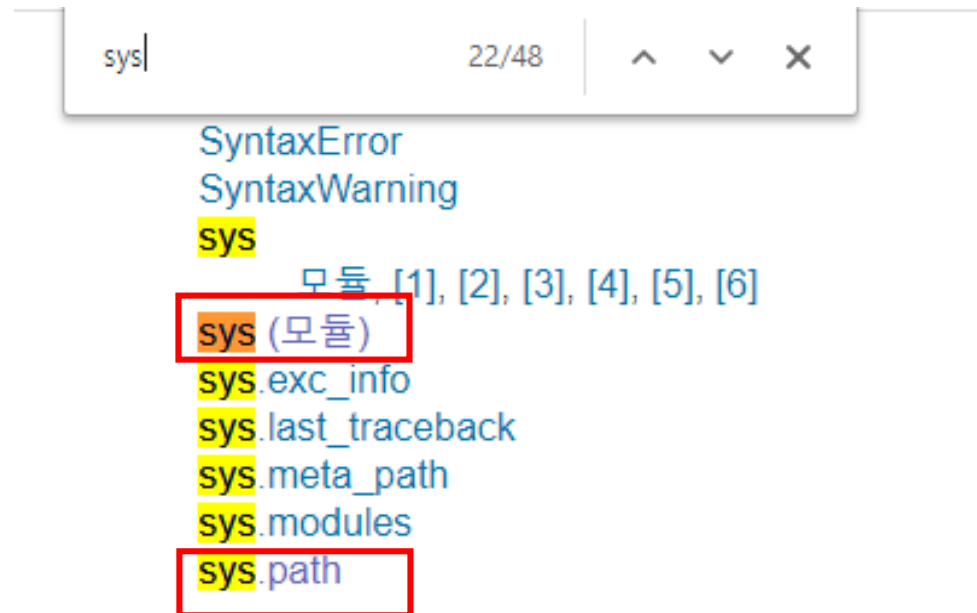
4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

■ sys

<https://docs.python.org/ko/3.8/genindex-S.html>



Python Library

1. Built-in function

1.1 `abs()` / `sum()`

2. External Functions and List

2.1 `os.getcwd()`

2.2 `sys.path`

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 `getcwd()` / `chdir()`

4.2 `listdir()`

4.3 `os.path.exists()`

4.4 `mkdir()`

4.5 `makedirs()`

4.6 `rmdir()`

4.7 `removedirs()`

■ `sys`

<https://docs.python.org/ko/3.8/library/sys.html#module-sys>

표준 라이브러리 - 모듈

`sys` — 시스템 특정 파라미터와 함수

이 모듈은 인터프리터에 의해 사용되거나 유지되는 일부 변수와 인터프리터와 강하게 상호 작용하는 함수에 대한 액세스를 제공합니다. 항상 사용 가능합니다.

`sys.abiflags`

표준 `configure` 스크립트를 사용하여 파이썬을 빌드한 POSIX 시스템에서, 이것은 [PEP 3149](#)에 지정된 ABI 플래그를 포함합니다.

버전 3.8에서 변경: 기본 플래그는 빈 문자열이 되었습니다 (pymalloc을 위한 `m` 플래그가 제거되었습니다).

버전 3.2에 추가.

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

■ path

<https://docs.python.org/ko/3.8/library/sys.html#module-sys>

표준 라이브러리 - 리스트

path - sys 모듈을 임포트한 후 사용가능한 리스트

path	7/18	^	v	x
------	------	---	---	---

sys.path

모듈의 검색 경로를 지정하는 문자열 리스트. 환경 변수 PYTHONPATH와 설치 종속 기본값으로 초기화되었습니다.

프로그램 시작 시 초기화된 대로, 이 리스트의 첫 번째 항목인 path[0]은 파이썬 인터프리터를 호출하는 데 사용된 스크립트가 포함된 디렉터리입니다. 스크립트 디렉터리를 사용할 수 없으면 (예를 들어, 인터프리터가 대화형으로 호출되거나 표준 입력에서 스크립트를 읽을 때) path[0]은 빈 문자열이 되는데, 파이썬이 현재 디렉터리에서 모듈을 먼저 검색하도록 합니다. 스크립트 디렉터리가 PYTHONPATH의 결과로 삽입된 항목 앞에 삽입됨에 유의하십시오.

프로그램은 자체 목적으로 이 리스트를 자유롭게 수정할 수 있습니다. 문자열과 바이트열만 sys.path에 추가해야 합니다; 임포트 하는 동안 다른 모든 데이터형은 무시됩니다.

더 보기: 모듈 site. 이것은 .pth 파일을 사용하여 sys.path를 확장하는 방법에 대해 설명합니다.

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

path 실습

```
1 import sys
2 sys.path
```

```
['D:\\ai',
 'C:\\Users\\tina\\anaconda3\\python38.zip',
 'C:\\Users\\tina\\anaconda3\\DLLs',
 'C:\\Users\\tina\\anaconda3\\lib',
 'C:\\Users\\tina\\anaconda3',
 '',
 'C:\\Users\\tina\\anaconda3\\lib\\site-packages',
 'C:\\Users\\tina\\anaconda3\\lib\\site-packages\\win32',
 'C:\\Users\\tina\\anaconda3\\lib\\site-packages\\win32\\lib',
 'C:\\Users\\tina\\anaconda3\\lib\\site-packages\\Pythonwin',
 'C:\\Users\\tina\\anaconda3\\lib\\site-packages\\IPython\\extensions',
 'C:\\Users\\tina\\.ipython']
```

```
1 print(type(sys))
2 print(type(sys.path))
```

```
<class 'module'>
<class 'list'>
```

```
1 sys.path[0]
```

```
'D:\\ai'
```

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

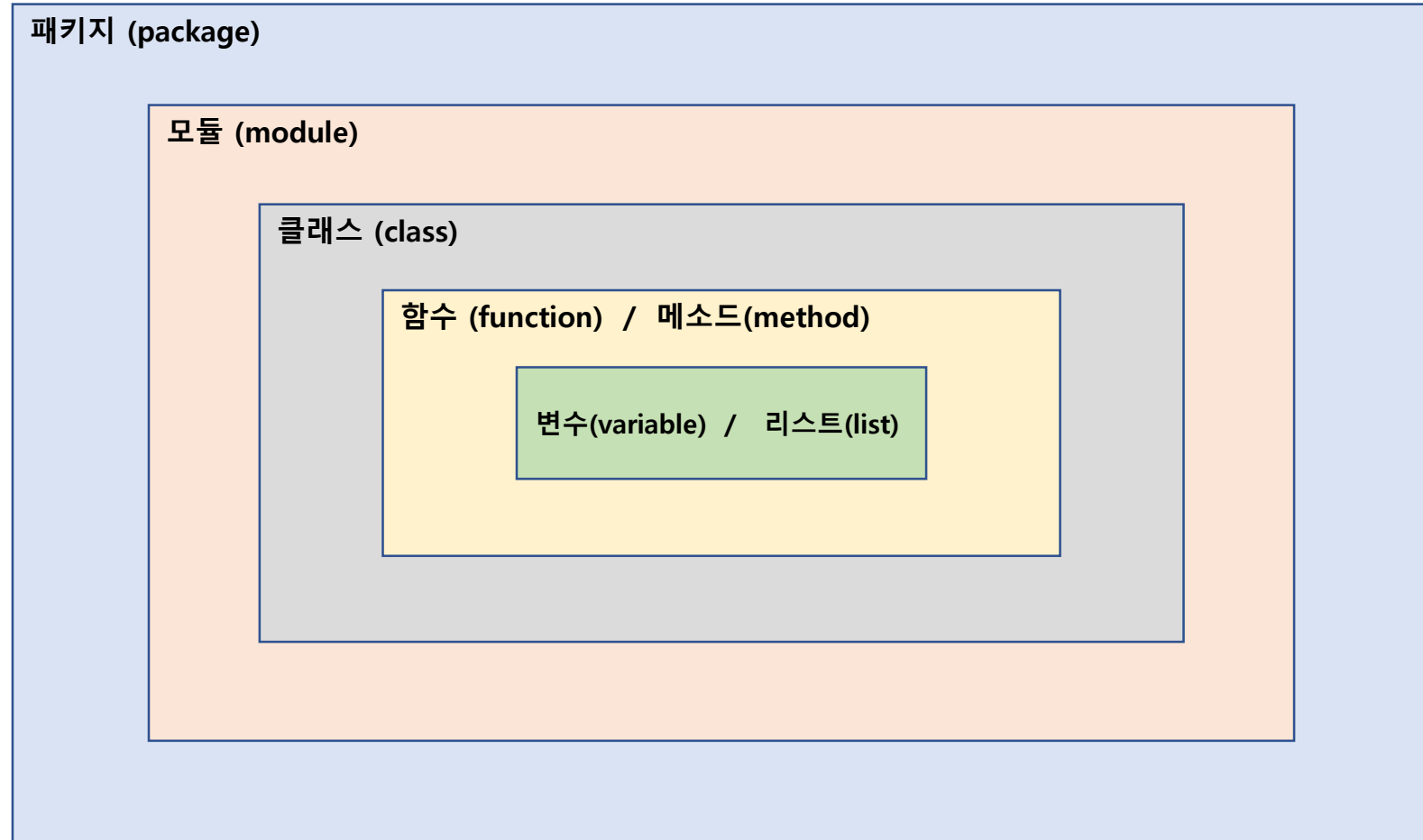
4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()



Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

■ 패키지(Package)

- Anaconda의 경우 conda를 이용하여 패키지 관리 가능

- 패키지 설치

(base) C:\Users\tina>conda install *packagename*

- 패키지 삭제

(base) C:\Users\tina>conda uninstall *packagename*

- 설치된 패키지 목록 검색

(base) C:\Users\tina>conda list

- 설치된 특정 패키지 검색

(base) C:\Users\tina>conda search *packagename*

예시 : (base) C:\Users\tina>conda search beautifulsoup

Anaconda
Navigator
Environment
활용 가능

Python Library

1. Built-in function

1.1 `abs()` / `sum()`

2. External Functions and List

2.1 `os.getcwd()`

2.2 `sys.path`

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 `getcwd()` / `chdir()`

4.2 `listdir()`

4.3 `os.path.exists()`

4.4 `mkdir()`

4.5 `makedirs()`

4.6 `rmdir()`

4.7 `removedirs()`

■ 패키지(Package)

- 파이썬 공식 홈페이지를 이용한 경우 pip를 이용하여 패키지 관리 가능
- Python 2.7.9 과 python 3.4 이상 버전에 기본으로 포함되어 있음

- 패키지 설치

(base) C:\Users\tina>pip install *packagename*

- 패키지 삭제

(base) C:\Users\tina>pip uninstall *packagename*

- 설치된 패키지 목록 검색

(base) C:\Users\tina>pip list

- 설치된 특정 패키지 검색

(base) C:\Users\tina>pip search *packagename*

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

■ 함수(Function)

def 함수이름 (parameter) :

 함수가 실행되면 실행할 문장 1

 함수가 실행되면 실행할 문장 2

return 출력할 결과값

예제 1

```
1 def p(x,y) :  
2     print('%s 과 %s 의 더한 값은 %s 입니다' %(x,y,x+y))  
3     result = '%s 과 %s 의 더한 값은 %s 입니다' %(x,y,x+y)  
4     return (result)  
5  
6 p(3,4)
```

3 과 4 의 더한 값은 7 입니다

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

[parameter로 받을 값의 개수가 유동적일 경우]

예제 2

```
1 def p(*x) :  
2     hap = 0  
3     for i in x :  
4         hap += i  
5     return('입력하신 숫자의 합은 %s 입니다' %hap)
```

```
1 p(1,3,5)
```

'입력하신 숫자의 합은 9 입니다'

```
1 p(2,4,6,8)
```

'입력하신 숫자의 합은 20 입니다'

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

[parameter로 받을 값의 개수가 유동적일 경우]

예제 3

```
1 def p(*x) :  
2     hap = 0  
3     for i in x :  
4         hap += i  
5     return(hap)  
6  
7 h1 = p(1,3,5)  
8 print("p함수 return 값 : %d" %h1)  
9  
10 h2 = p(2,4,6,8)  
11 print("p함수 return 값 : %d" %h2)  
12
```

p함수 return 값 : 9

p함수 return 값 : 20

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

■ 모듈(Module)

- 변수, 함수, 클래스를 재사용하기 위해서 파일로 저장해 둔 코드

- 모듈 활용 실습

(1) 실습 : 모듈을 직접 만들어서 사용하기

(2) 실습 : 모듈의 저장 경로 변경하여 사용하기

(3) 실습 : 모듈 안의 특정 함수만 불러와서 사용하기

Python Library

1. Built-in function

1.1 `abs()` / `sum()`

2. External Functions and List

2.1 `os.getcwd()`

2.2 `sys.path`

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 `getcwd()` / `chdir()`

4.2 `listdir()`

4.3 `os.path.exists()`

4.4 `mkdir()`

4.5 `makedirs()`

4.6 `rmdir()`

4.7 `removedirs()`

(1) 실습

모듈을 직접 만들어서 사용하기

작업 디렉토리를 `D:/ai` 라고 가정하고 실습

(1) 실습

모듈 만들기

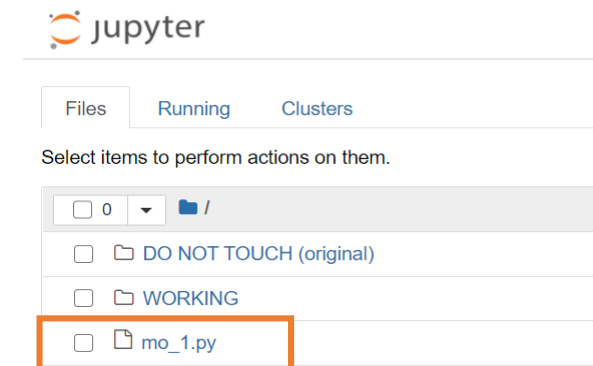
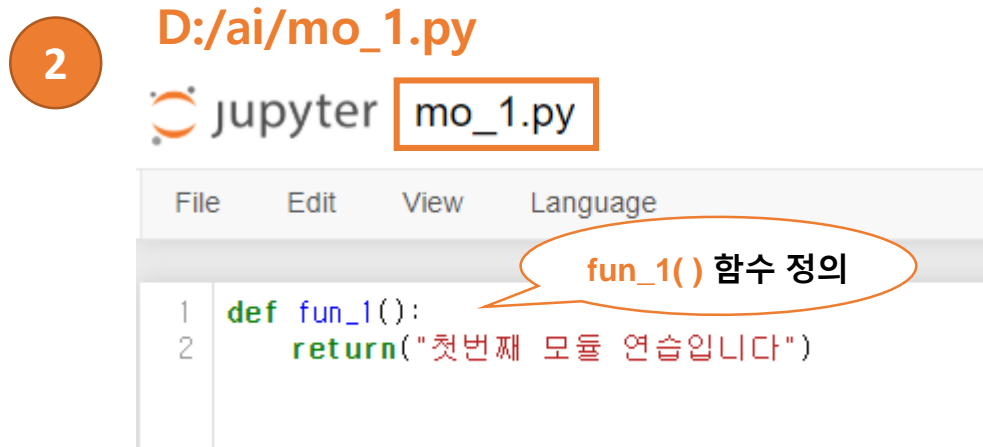
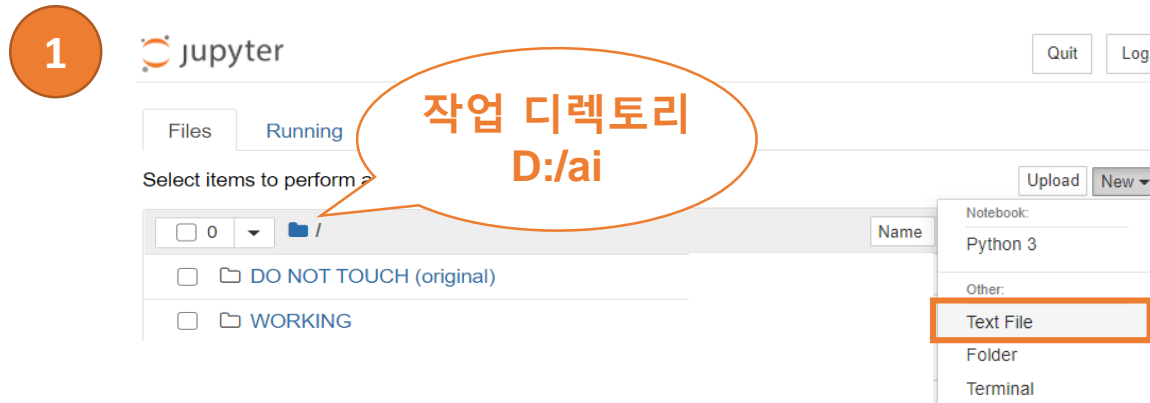
Python Library

1. Built-in function
1.1 abs() / sum()

2. External Functions and List
2.1 os.getcwd()
2.2 sys.path

3. Python Structure
3.1 Package
3.2 Function
3.3 Module

4. Directory Management
4.1 getcwd() / chdir()
4.2 listdir()
4.3 os.path.exists()
4.4 mkdir()
4.5 makedirs()
4.6 rmdir()
4.7 removedirs()

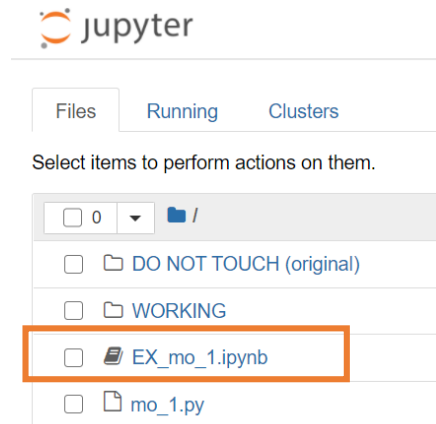


(1) 실습

모듈 사용하기



'첫번째 모듈 연습입니다'



Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

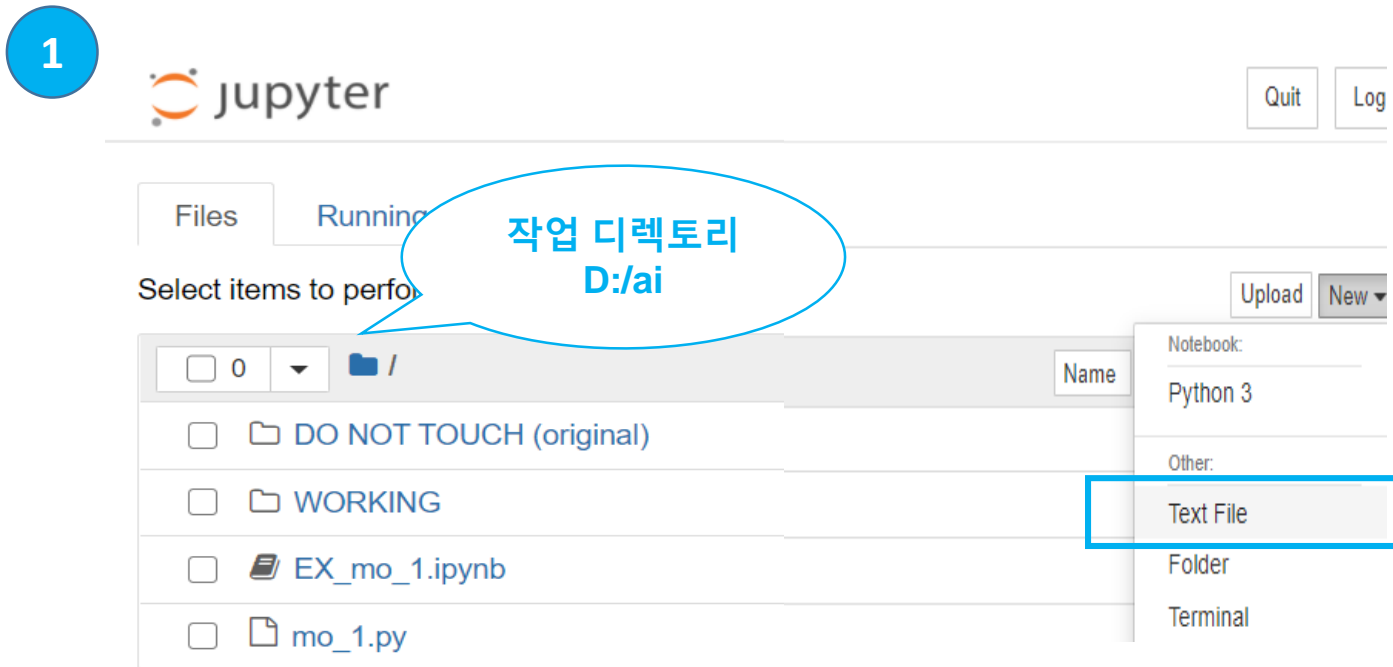
4.7 removedirs()

(2) 실습

모듈의 저장 경로 변경하여 사용하기

작업 디렉토리를 D:/ai 라고 가정하고 실습

(2) 실습 모듈 만들기



(2) 실습 모듈 만들기

Python Library

1. Built-in function
1.1 `abs()` / `sum()`

2. External Functions
and List
2.1 `os.getcwd()`
2.2 `sys.path`

3. Python Structure
3.1 Package
3.2 Function
3.3 Module

4. Directory Management
4.1 `getcwd()` / `chdir()`
4.2 `listdir()`
4.3 `os.path.exists()`
4.4 `mkdir()`
4.5 `makedirs()`
4.6 `rmdir()`
4.7 `removedirs()`

2 D:/ai/mo_2.py

jupyter **mo_2.py**

```
File Edit View Language
1 def fun_1():
2     return("mo_2 모듈 안에 있는 첫번째 함수")
3
4 def fun_2():
5     return("mo_2 모듈 안에 있는 두번째 함수")
6
7 def fun_3():
8     return("mo_2 모듈 안에 있는 세번째 함수")
9
10 def fun_4():
11     return("mo_2 모듈 안에 있는 네번째 함수")
```



jupyter

Files Running Clusters

Select items to perform actions on them.

☐ 0 ▾ /

☐ DO NOT TOUCH (original)

☐ WORKING

☐ EX_mo_1.ipynb

☐ mo_1.py

☐ **mo_2.py**

(2) 실습 모듈 사용하기

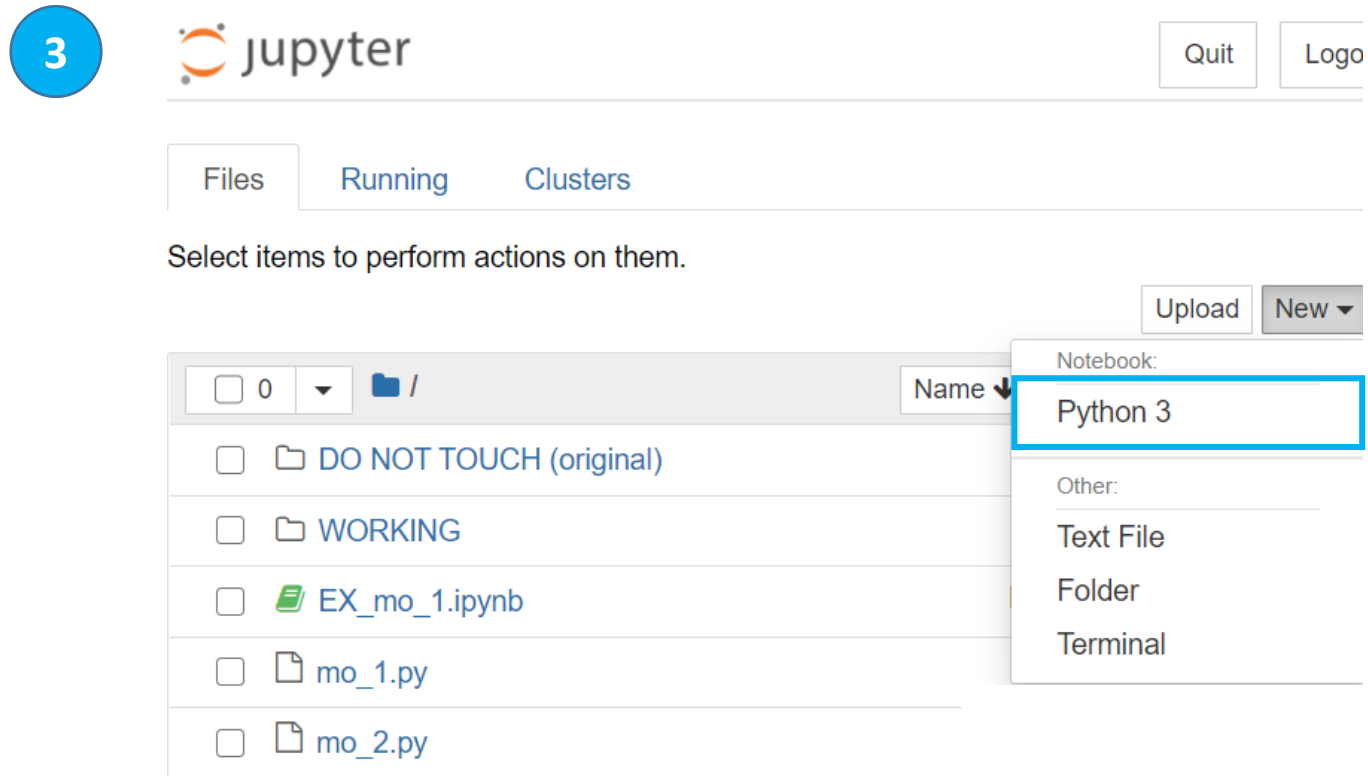
Python Library

1. Built-in function
1.1 `abs()` / `sum()`

2. External Functions
and List
2.1 `os.getcwd()`
2.2 `sys.path`

3. Python Structure
3.1 Package
3.2 Function
3.3 Module

4. Directory Management
4.1 `getcwd()` / `chdir()`
4.2 `listdir()`
4.3 `os.path.exists()`
4.4 `mkdir()`
4.5 `makedirs()`
4.6 `rmdir()`
4.7 `removedirs()`



(2) 실습

모듈 사용하기

4

D:/ai/EX_mo_2.ipynb

```
1 import os
2 os.mkdir('D:/temp00')
3 os.chdir('D:/temp00')
4 os.getcwd()
```

'D:\\\\temp00'

```
1 os.chdir('D:/ai')
2 os.getcwd()
```

'D:\\\\ai'

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

(2) 실습 모듈 사용하기

5

D:/ai/EX_mo_2.ipynb

```
1 os.listdir('D:/ai')
```

```
['.ipynb_checkpoints',  
'DO NOT TOUCH (original)',  
'EX_mo_1.ipynb',  
'EX_mo_2.ipynb',  
'mo_1.py',  
'mo_2.py',  
'WORKING',  
'__pycache__']
```

```
1 # mo_2.py 모듈이 있는지 찾는다  
2  
3 result = [i for i in os.listdir('D:/ai') if 'mo_2.py' in i]  
4 print(result)
```

```
['mo_2.py']
```

Python Library

1. Built-in function
1.1 abs() / sum()

2. External Functions
and List
2.1 os.getcwd()
2.2 sys.path

3. Python Structure
3.1 Package
3.2 Function
3.3 Module

4. Directory Management
4.1 getcwd() / chdir()
4.2 listdir()
4.3 os.path.exists()
4.4 mkdir()
4.5 makedirs()
4.6 rmdir()
4.7 removedirs()

6

D:/ai/EX_mo_2.ipynb

(2) 실습

모듈 사용하기

```
1 import shutil
2 shutil.move('mo_2.py', 'D:/temp00')
```

'D:/temp00###mo_2.py'

D:/ai/mo_2.py 모듈의 저장 경로를
D:/temp00으로 이동하기

```
1 os.listdir('D:/ai')
```

```
['.ipynb_checkpoints',
'DO NOT TOUCH (original)',
'EX_mo_1.ipynb',
'EX_mo_2.ipynb',
'mo_1.py',
'WORKING',
'__pycache__']
```

mo_2.py는 작업 디렉토리 D:/ai 에 없고,
D:/temp00 에 있음

```
1 os.listdir('D:/temp00')
```

```
['mo_2.py']
```

Python Library

1. Built-in function
1.1 abs() / sum()

2. External Functions
and List
2.1 os.getcwd()
2.2 sys.path

3. Python Structure
3.1 Package
3.2 Function
3.3 Module

4. Directory Management
4.1 getcwd() / chdir()
4.2 listdir()
4.3 os.path.exists()
4.4 mkdir()
4.5 makedirs()
4.6 rmdir()
4.7 removedirs()

7

D:/ai/EX_mo_2.ipynb

(2) 실습

모듈 사용하기

Python Library

1. Built-in function
1.1 abs() / sum()

2. External Functions
and List
2.1 os.getcwd()
2.2 sys.path

3. Python Structure
3.1 Package
3.2 Function
3.3 Module

4. Directory Management
4.1 getcwd() / chdir()
4.2 listdir()
4.3 os.path.exists()
4.4 mkdir()
4.5 makedirs()
4.6 rmdir()
4.7 removedirs()

```
1 # temp00 디렉토리에 mo_2.py 모듈이 있는지 찾는다
2
3 result = [i for i in os.listdir('D:/temp00') if 'mo_2.py' in i]
4 print(result)
```

```
['mo_2.py']
```

```
1 os.listdir('D:/temp00')
```

```
['mo_2.py']
```

```
1 os.getcwd()
```

```
'D:\\\\ai'
```

mo_2.py는 작업 디렉토리 D:/ai 에 없으므로 에러 발생

```
1 import mo_2
```

ModuleNotFoundError

Traceback (most recent call last)

<ipython-input-1-4127ab16c4cd> in <module>

----> 1 import mo_2

ModuleNotFoundError: No module named 'mo_2'

8

D:/ai/EX_mo_2.ipynb

(2) 실습

모듈 사용하기

Python Library

1. Built-in function
1.1 abs() / sum()

2. External Functions
and List
2.1 os.getcwd()
2.2 sys.path

3. Python Structure
3.1 Package
3.2 Function
3.3 Module

4. Directory Management
4.1 getcwd() / chdir()
4.2 listdir()
4.3 os.path.exists()
4.4 mkdir()
4.5 makedirs()
4.6 rmdir()
4.7 removedirs()

```
1 import sys
2 print(sys.path)
```

```
['D:\\ai', 'C:\\Users\\tina\\anaconda3\\python38.zip', 'C:\\Users\\tina\\anaconda3\\DLLs', 'C:\\Users\\tina\\anaconda3\\lib', 'C:\\Users\\tina\\anaconda3', '', 'C:\\Users\\tina\\anaconda3\\lib\\site-packages', 'C:\\Users\\tina\\anaconda3\\lib\\site-packages\\win32', 'C:\\Users\\tina\\anaconda3\\lib\\site-packages\\win32\\lib', 'C:\\Users\\tina\\anaconda3\\lib\\site-packages\\Pythonwin', 'C:\\Users\\tina\\anaconda3\\lib\\site-packages\\IPython\\extensions', 'C:\\Users\\tina\\.ipython']
```

D:/temp00 을 path에 추가

```
1 sys.path.append('D:/temp00')
2 print(sys.path)
```

append된 path는 Kernel을 restart하면 reset됨

```
['D:\\ai', 'C:\\Users\\tina\\anaconda3\\python38.zip', 'C:\\Users\\tina\\anaconda3\\DLLs', 'C:\\Users\\tina\\anaconda3\\lib', 'C:\\Users\\tina\\anaconda3', '', 'C:\\Users\\tina\\anaconda3\\lib\\site-packages', 'C:\\Users\\tina\\anaconda3\\lib\\site-packages\\win32', 'C:\\Users\\tina\\anaconda3\\lib\\site-packages\\win32\\lib', 'C:\\Users\\tina\\anaconda3\\lib\\site-packages\\Pythonwin', 'C:\\Users\\tina\\anaconda3\\lib\\site-packages\\IPython\\extensions', 'C:\\Users\\tina\\.ipython', 'D:/temp00']
```

9

D:/ai/EX_mo_2.ipynb

(2) 실습
모듈 사용하기

```
1 os.getcwd()
```

```
'D:\\\\ai '
```

D:/temp00은 path설정이 되어있고,
mo_2.py 는 D:/temp00 에 있으므로 import 가능

```
1 import mo_2
```

module name 다음에 . (dot)을 사용하여 경로를 구분한 다음
function name 사용

```
1 mo_2.fun_1()
```

'mo_2 모듈 안에 있는 첫번째 함수'

```
1 mo_2.fun_2()
```

'mo_2 모듈 안에 있는 두번째 함수'

```
1 mo_2.fun_3()
```

'mo_2 모듈 안에 있는 세번째 함수'

```
1 mo_2.fun_4()
```

'mo_2 모듈 안에 있는 네번째 함수'

Python Library

1. Built-in function
1.1 abs() / sum()

2. External Functions
and List
2.1 os.getcwd()
2.2 sys.path

3. Python Structure
3.1 Package
3.2 Function
3.3 Module

4. Directory Management
4.1 getcwd() / chdir()
4.2 listdir()
4.3 os.path.exists()
4.4 mkdir()
4.5 makedirs()
4.6 rmdir()
4.7 removedirs()

(2) 실습

모듈 사용하기

10

D:/ai/EX_mo_2.ipynb

function name만
사용하여 호출하면 에러

```
1 print(fun_2())
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-10-f04f69a8e712> in <module>  
----> 1 print(fun_2())
```

```
NameError: name 'fun_2' is not defined
```

Python Library

1. Built-in function
1.1 abs() / sum()

2. External Functions
and List
2.1 os.getcwd()
2.2 sys.path

3. Python Structure
3.1 Package
3.2 Function
3.3 Module

4. Directory Management
4.1 getcwd() / chdir()
4.2 listdir()
4.3 os.path.exists()
4.4 mkdir()
4.5 makedirs()
4.6 rmdir()
4.7 removedirs()

Python Library

1. Built-in function

1.1 `abs()` / `sum()`

2. External Functions and List

2.1 `os.getcwd()`

2.2 `sys.path`

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 `getcwd()` / `chdir()`

4.2 `listdir()`

4.3 `os.path.exists()`

4.4 `mkdir()`

4.5 `makedirs()`

4.6 `rmdir()`

4.7 `removedirs()`

(3) 실습

모듈 안의 특정 함수만 불러와서 사용하기

작업 디렉토리를 `D:/ai` 라고 가정하고 실습

(3) 실습

모듈 만들기

모듈 사용하기

1

D:/ai/food.py

작업 디렉토리
D:/ai

jupyter food.py

File Edit View Language

```
1 name = "홍길동"
2
3 def cook():
4     print("요리하고.")
5     return(1)
6
7 def eat():
8     print("먹고.")
9     return(2)
10
11
12 def sleep():
13     print("잠들다.")
14     return(2)
```



D:/ai/EX_mo_3.ipynb

작업 디렉토리
D:/ai

```
1 import food
```

```
1 food.cook()
2 food.eat()
3 food.sleep()
4 print("그가 바로", food.name, "입니다")
```

요리하고.
먹고.
잠들다.
그가 바로 홍길동 입니다

(3) 실습 모듈 사용하기

2

D:/ai/EX_mo_3.ipynb

```
1 import food
2 print(name)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-9-44f8436829e6> in <module>
      1 import food
----> 2 print(name)
```

NameError: name 'name' is not defined

```
1 cook()
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-10-b935b1fef2dc> in <module>
----> 1 cook()
```

NameError: name 'cook' is not defined

D:/ai/food.py

 jupyter **food.py**

File	Edit	View	Language
------	------	------	----------

```
1 name = "홍길동"
2
3 def cook():
4     print("요리하고.")
5     return(1)
6
7 def eat():
8     print("먹고.")
9     return(2)
10
11
12 def sleep():
13     print("잠들다.")
14     return(2)
```

Python Library

1. Built-in function

- 1.1 abs() / sum()

2. External Functions and List

- 2.1 os.getcwd()
- 2.2 sys.path

3. Python Structure

- 3.1 Package
- 3.2 Function
- 3.3 Module

4. Directory Management

- 4.1 getcwd() / chdir()
- 4.2 listdir()
- 4.3 os.path.exists()
- 4.4 mkdir()
- 4.5 makedirs()
- 4.6 rmdir()
- 4.7 removedirs()

(3) 실습 모듈 사용하기

3

D:/ai/EX_mo_3.ipynb

from **module name** import **function name**

```
1 from food import cook, name
2 print(name)
3 a = cook()
4 print("리턴 값 %d " %a)
```

홍길동
요리하고.
리턴 값 1

function name만
사용하여 호출

D:/ai/food.py

jupyter food.py

File Edit View Language

```
1 name = "홍길동"
2
3 def cook():
4     print("요리하고.")
5     return(1)
6
7 def eat():
8     print("먹고.")
9     return(2)
10
11
12 def sleep():
13     print("잠들다.")
14     return(2)
```

(3) 실습 모듈 사용하기

4 D:/ai/EX_mo_3.ipynb

```
1 from food import cook, name
2 print(name)
3 b = eat()
4 print("리턴 값 %d" %b)
```

홍길동

NameError

Traceback (most

recent call last)

<ipython-input-3-87783910d520> in <module>

1 from food import cook, name

2 print(name)

----> 3 b = eat()

4 print("리턴 값 %d" %b)

NameError: name 'eat' is not defined

D:/ai/food.py

jupyter food.py

File Edit View Language

```
1 name = "홍길동"
2
3 def cook():
4     print("요리하고.")
5     return(1)
6
7 def eat():
8     print("먹고.")
9     return(2)
10
11
12 def sleep():
13     print("잠들다.")
14     return(2)
```

(3) 실습 모듈 사용하기

5

D:/ai/EX_mo_3.ipynb

```
1 from food import *
2 print(name)
3 print(cook(), eat(), sleep())
```

홍길동
요리하고.
먹고.
잠들다.
1 2 2

D:/ai/food.py

 jupyter **food.py**

File Edit View Language

```
1 name = "홍길동"
2
3 def cook():
4     print("요리하고.")
5     return(1)
6
7 def eat():
8     print("먹고.")
9     return(2)
10
11
12 def sleep():
13     print("잠들다.")
14     return(2)
```

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

1. 관련 함수

1) getcwd() 함수 / chdir() 함수

```
1 import os
2 print('현재 디렉토리:', os.getcwd() )
3
4 os.chdir("D:/temp00")
5 print('이동 후 디렉토리:', os.getcwd() )
```

현재 디렉토리 : D:\wai

이동 후 디렉토리 : D:\temp00

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

2) listdir() - 지정된 디렉토리 내의 내용(파일 및 하위 디렉토리)을 출력하는 함수

```
1 os.listdir("D:/ai")
```

```
[ '.ipynb_checkpoints',  
  '04_Data File Management (1~24 slide).ipynb',  
  '04_Data File Management (43~ slide).ipynb',  
  'ai 폴더 관리시 참고 사항',  
  'DO NOT TOUCH (original)',  
  'EX_mo_1.ipynb',  
  'EX_mo_2.ipynb',  
  'EX_mo_3.ipynb',  
  'food.py',  
  'mo_1.py',  
  'WORKING',  
  '__pycache__']
```

```
1 for i in os.listdir("D:/ai") :  
2     print(i)
```

```
.ipynb_checkpoints  
04_Data File Management (1~24 slide).ipynb  
04_Data File Management (43~ slide).ipynb  
ai 폴더 관리시 참고 사항  
DO NOT TOUCH (original)  
EX_mo_1.ipynb  
EX_mo_2.ipynb  
EX_mo_3.ipynb  
food.py  
mo_1.py  
WORKING  
__pycache__
```

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

3) os.path.exists() - 특정 디렉토리의 존재 여부를 확인하는 함수

```
1 print( os.path.exists("D:/ai") )  
2 print( os.path.exists("D:/temp00") )  
3 print( os.path.exists("D:/temp33") )
```

True

True

False

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

4) mkdir() - 디렉토리를 생성하는 함수

```
1 os.mkdir("D:/temp11")
2 os.mkdir("D:/temp11/test")
3 print( os.path.exists("D:/temp11") )
4 print( os.path.exists("D:/temp11/test") )
```

True

True

temp22 디렉토리가 없는
상태에서 test 디렉토리
생성을 시도 → Error

```
1 os.mkdir("D:/temp22/test")
```

FileNotFoundError

Traceback (most recent call last)

<ipython-input-11-75b9b379e2b3> in <module>

----> 1 os.mkdir("D:/temp22/test")

FileNotFoundError: [WinError 3] 지정된 경로를 찾을 수 없습니다: 'D:/temp22/test'

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

5) makedirs() - 디렉토리를 생성하는 함수

```
1 os.makedirs("D:/temp22/test")
2 print( os.path.exists("D:/temp22") )
3 print( os.path.exists("D:/temp22/test") )
```

True

True

temp22 디렉토리가 없는
상태에서 test 디렉토리 생성을 시도
→ temp22는 물론 하위 디렉토리에
test 디렉토리가 생성됨

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

6) rmdir() - 디렉토리를 삭제하는 함수, 디렉토리 삭제는 해당 디렉토리가 반드시 비어 있어야함

```
1 os.rmdir("D:/temp22")
2 print( os.path.exists("D:/temp22") )
```

OSError Traceback (most recent call last)

<ipython-input-24-73f44d4293e3> in <module>

```
----> 1 os.rmdir("D:/temp22")
      2 print( os.path.exists("D:/temp22") )
```

OSError: [WinError 145] 디렉터리가 비어 있지 않습니다: 'D:/temp22'

```
1 os.rmdir("D:/temp22/test")
2 print( os.path.exists("D:/temp22/test") )
```

False

Python Library

1. Built-in function

1.1 abs() / sum()

2. External Functions and List

2.1 os.getcwd()

2.2 sys.path

3. Python Structure

3.1 Package

3.2 Function

3.3 Module

4. Directory Management

4.1 getcwd() / chdir()

4.2 listdir()

4.3 os.path.exists()

4.4 mkdir()

4.5 makedirs()

4.6 rmdir()

4.7 removedirs()

7) removedirs() - 해당하는 모든 디렉토리를 삭제하는 함수,

디렉토리 삭제는 해당 디렉토리가 반드시 비어 있어야함

```
1 print( os.path.exists("D:/temp11") )
2 print( os.path.exists("D:/temp11/test") )
```

True

True

```
1 os.rmdir("D:/temp11")
2 print( os.path.exists("D:/temp11") )
```

OSError Traceback (most recent call last)

<ipython-input-33-026c8eef8b01> in <module>

```
----> 1 os.rmdir("D:/temp11")
      2 print( os.path.exists("D:/temp11") )
```

OSError: [WinError 145] 디렉터리가 비어 있지 않습니다: 'D:/temp11'

```
1 os.removedirs("D:/temp11/test")
2 print( os.path.exists("D:/temp11") )
3 print( os.path.exists("D:/temp11/test") )
```

False

False

temp11, test 디렉토리
모두 삭제됨