

학습 내용

2부. 프로그래밍 언어 활용



6장. 모듈과 패키지

- 1. 파이썬 표준 모듈
- 2. 사용자 정의 모듈

7장. 객체지향 프로그래밍

8장. 예외 처리

9장. 파일 입/출력 프로그래밍

10장. 데이터베이스 연동

1.1 파이썬 모듈

1절. 파이썬 모듈 사용하기

- 모듈은 파이썬 정의와 문장을 담고 있는 파일(함수 또는 변수 정의)
- 파일 이름은 접미어 .py가 추가 된 모듈 이름
- 함수 또는 변수의 정의를 파일에 넣고 스크립트 또는 인터프리터의 대화형 인스턴스에서 사용하는 방법을 가지고 있음

1.2 파이썬 표준 모듈

1절. 파이썬 모듈 사용하기

- 파이썬 라이브러리 레퍼런스(Python Library Reference)에서 설명하는 **표준 모듈 라이브러리**를 제공
- 일부 모듈은 인터프리터에 내장되어 효율성이나 시스템 호출과 같은 운영 체제 기본 요소 또는 내장되어 있는 작업에 대한 접근을 제공
- 일부 모듈 세트는 기본 플랫폼에 종속
- 문자열(string), 날짜(date), 시간(time), 수학(math), 분수(fractions), 랜덤(random), 파일(file), sqlite3, os, sys, threading, unittest, xml, email, http 등 200여개의 다양한 모듈이 존재
- 일부 모듈 세트는 기본 플랫폼에 종속될 수 있음
- 파이썬 라이브러리 레퍼런스(Python Library Reference)
 - <https://docs.python.org/3/library/index.html>

1.3 import 하는 방법 : import ~

1절. 파이썬 모듈 사용하기

- import *모듈명*
- 모듈 안의 함수들은 모듈 이름을 붙여 사용

```
import time
```

```
time.ctime()
```

```
'Mon Mar 25 15:32:45 2019'
```

1.3 import 하는 방법 : from A import B

1절. 파이썬 모듈 사용하기

- from *패키지명* import *모듈명* # 패키지는 *directory*, 모듈은 *.py파일*
- from *모듈명* import *함수명*

```
from time import ctime  
ctime()
```

```
'Mon Mar 25 15:33:22 2019'
```

```
from time import *  
ctime()
```

```
'Mon Mar 25 15:35:57 2019'
```

1.3 import 하는 방법 : import A as B

1절. 파이썬 모듈 사용하기

- import *패키지명* as *패키지별칭*
- import *모듈명* as *모듈별칭*
- A 모듈 또는 패키지의 이름이 길 경우 별칭을 주어 짧게 쓰기 위한 용도로 사용

```
import time as t  
t.ctime()
```

```
'Mon Mar 25 15:39:56 2019'
```

1.4 dir()

1절. 파이썬 모듈 사용하기

- 모듈이 정의한 이름을 정렬된 문자열 목록으로 반환

```
import math
print(dir(math))
```

```
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']
```

2.1 모듈 만들기

2절. 사용자 정의 모듈

- C:/pylib/a.py 파일로 작성

```
def test():  
    '''  
    모듈의 test함수  
    '''  
    print('모듈 테스트입니다')  
    print('더운 날씨 건강조심하세요')  
if __name__ == '__main__':  
    test()  
    print(__name__)
```


2.2 모듈 가져오기

2절. 사용자 정의 모듈

- 모듈을 import 하면...
 - 해당 이름을 가진 내장 모듈 검색, sys.path 변수에 지정된 디렉토리들 검색
- sys.path 초기화
 - 입력 스크립트가 들어있는 디렉토리
 - PYTHONPATH 환경 변수에 지정한 디렉토리
 - 표준 라이브러리 디렉토리. ex) C:\big\IDE\Python\Python39\Lib
- sys.path
 - sys.path.insert(index, path) 또는 sys.path.append(path)로 추가
 - sys.path.remove(path)로 제거

```
import sys
sys.path.append('C:/pylib')
```

3.5 패키지 설치 및 삭제

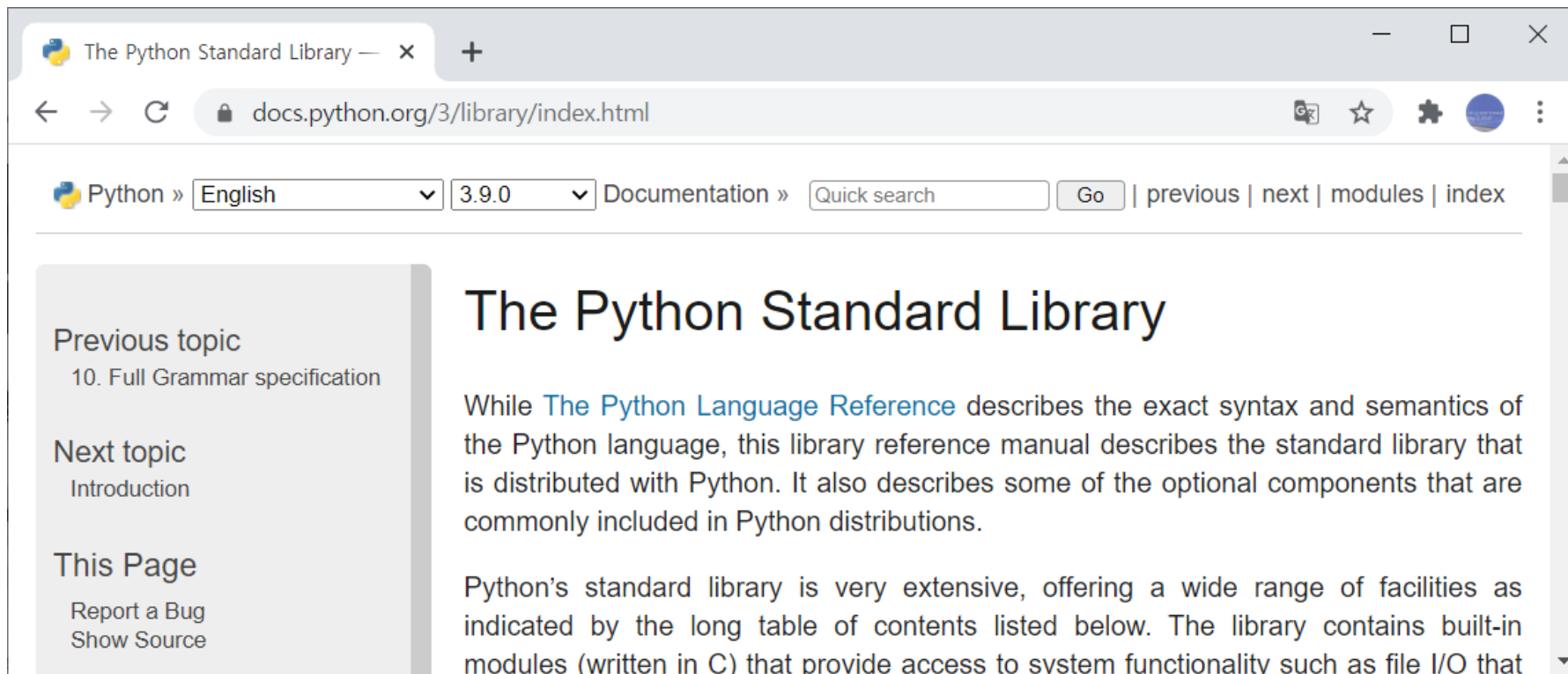
3절 패키지

- 패키지 이름으로 설치
 - `pip install 패키지명`
 - `conda install 패키지명`
- whl 파일을 이용한 설치
 - `python -m pip install whl파일명`
- 패키지 관리
 - `conda list 패키지명`
 - `conda remove 패키지명`
 - `pip uninstall 패키지명`

4절. 파이썬 표준 라이브러리

많은 문제를 해결하기 위한 표준화 된 솔루션을 제공

<https://docs.python.org/3/library/index.html>



많은 문제를 해결하기 위한 표준화 된 솔루션을 제공

<https://pypi.org/>

