



파이썬기초

5주차 - 1교시
모듈





학습내용

- 모듈



학습목표

- 파이썬 모듈에 대해 이해하고,
모듈을 이용한 기본적인 코딩을 할 수 있다.



함수

- 값을 전달하면 전달 받은 값을 이용하여 **약속된 처리**를 한 후 처리된 결과 또는 약속된 값을 반환해주는 기능임
- 함수를 사용하는 이유
 - 하나의 프로그램에서 반복 사용되는 부분을 함수로 작성함으로써, **코드의 크기를 줄이고 재사용**할 수 있음
 - 프로그램을 기능 단위로 분리하여 구조화 함으로서, 코드의 **가독성**을 높이고 **유지보수**가 용이함



함수

- 사용자 정의함수, 람다함수, 내장함수, 외장함수로 구분됨
- **def** 키워드를 이용하고 함수의 이름, 매개변수, return 키워드를 이용해서 정의함

```
def 함수이름 (매개변수 1, 매개변수 2, ... ) :
```



(tap) 실행 코드

실행 코드

실행 코드

```
return(반환 값 또는 연산 식 또는 공백)
```



함수

- 함수의 변수 사용 시 **지역변수**와 **전역변수**로 구분하여 사용함



람다함수

- 익명 함수 형태로 정의하는 람다함수는 함수를 전달 받아 사용하는 함수와 함께 사용됨
- map(), filter()** 함수는 전달된 리스트를 전달받은 함수를 이용해서 새로운 리스트를 생성함



예외처리

- **예외 상황을 예측**하고 예외 발생시 비정상적인 종료를 하지 않고 프로그램이 정상적으로 실행되도록 하는 것을 말함
- 예외처리 구문
 - > try-except
 - > try-except-finally



파이썬의 내장함수와 외장함수

- 파이썬의 내장함수
 - 프로그래밍의 편의를 위해 **프로그래밍에서 많이 사용되는 기능**들을 내장함수로 제공하는 함수
- 파이썬의 외장함수
 - **import** 문을 사용하여 외부의 라이브러리에서 제공하는 함수



생각해 봅시다

자주 사용되는 코드를 미리 정의해 놓고
필요한 시점에 사용할 수 있는
방법에 대해 생각해봅시다.



01



모듈



1 | 모듈

01 모듈(module)이란

모듈

파이썬 코드 파일(확장자 .py)이자, 파이썬 코드들을 저장하는 기본단위로서 서로 관련 있는 작은 프로그램 조각들을 모아 놓은 것

→ 함수의 집합

모듈

하나하나 연결해 어떤 목적을 가진 프로그램을 만들기 위한 작은 프로그램

1 | 모듈

01 모듈(module)이란

- 코드를 **분리하고 공유**하는 기능
- 프로그램을 만들 때 자주 사용하는 코드를 별도의 파일로 만들어서 **필요할 때마다 재사용**할 수 있게 함
 - 파이썬에서는 모든 모듈이 **독립적**으로 실행됨

1 | 모듈

02 모듈의 장점



효율적 프로그램 개발

: 자주 사용되는 기능 재사용 가능



복잡성 감소 및 프로그램 유지보수 용이성

: 기능의 분리



효율적 메모리 사용

: 필요한 부분만 불러 사용

2 | 모듈의 구분

01 표준 모듈



➤ 파이썬을 설치하면 파이썬 패키지 안에 **기본**으로 포함되어 제공되는 모듈

2 | 모듈의 구분

02 사용자 정의 모듈



- > 사용자가 직접 생성하여 사용하는 모듈
- > 파이썬 표준 모듈이 모든 기능을 제공하지 않음

2 | 모듈의 구분

03 외부 모듈(3rd Party)



- 파이썬 이외의 다른 기관이나 SW회사 등에서 배포한 모듈로 **특정 기능을 제공하는 모듈**
- ✓ 전문분야별 또는 특수한 프로그래밍 기능별로 모듈을 제작해서 배포함
- ✓ 고급 프로그래밍 가능

3 | 모듈의 사용

01 import 문

- **import 문**을 이용하면 다른 모듈을 가져와서 사용할 수 있음

- ✓ as(alias)

- 긴 모듈명을 줄여 사용할 수 있음

- ✓ 'from 모듈 import 함수1, 함수2...'

- 모듈 내 특정 함수만 가져올 수 있음

3 | 모듈의 사용

01 import 문

- 모듈 **module.py**와 이것을 import 문으로 가져가는 모듈은 **같은 폴더**에 있어야 함
- 실행코드를 포함하는 모듈을 import하면 그 실행코드도 실행됨

다른 모듈에 의해 import 되는 모듈들

함수와 변수정의들만 사용 가능하고 실행코드를 포함하지 않아야 함

3 | 모듈의 사용

01 import 문

- **main() 함수**의 개념이 없음
→ 제일 먼저 실행하는 모듈이 main() 함수 역할을 함



3 | 모듈의 사용

02 모듈 검색 순서

이미 메모리에 로딩된 모듈



현재 디렉토리에 있는 .py 파일



환경변수(PYTHONPATH)에 등록된 경로에 있는 파일들



표준 모듈 목록

3 | 모듈의 사용

03 표준 모듈 목록

```
In [6]: import sys  
print(sys.builtin_module_names)
```

결과

```
(' _abc', ' _ast', ' _bisect', ' _blake2', ' _codecs', ' _codecs_cn', ' _codecs_hk',  
' _codecs_iso2022', ' _codecs_jp', ' _codecs_kr', ' _codecs_tw',  
' _collections', ' _contextvars', ' _csv', ' _datetime', ' _functools', ' _heapq',  
' _imp', ' _io', ' _json', ' _locale', ' _lsp rof', ' _md5', ' _multibytecodec', ' _opcode',  
' _operator', ' _pickle', ' _random', ' _sha1', ' _sha256', ' _sha3', ' _sha512',  
' _signal', ' _sre', ' _stat', ' _statistics', ' _string', ' _struct', ' _symtable',  
' _thread', ' _tracemalloc', ' _warnings', ' _weakref', ' _winapi',  
' _xxsubinterpreters', ' array', ' atexit', ' audioop', ' binascii', ' builtins',  
' cmath', ' errno', ' faulthandler', ' gc', ' itertools', ' marshal', ' math', ' mmap',  
' msvcrt', ' nt', ' sys', ' time', ' winreg', ' xxsubtype', ' zlib')
```

3 | 모듈의 사용

04 외부(서드파티) 모듈 설치

- 서드파티 모듈을 이용하려면,
해당 모듈을 **다운받아 설치**해야 함

➤ pip(the package installer for Python)

- **파이썬 패키지를 설치하는 프로그램**은
파이썬을 설치하면 기본으로 설치됨

➤ 다운로드 설치 가능: <https://pypi.python.org/>

3 | 모듈의 사용

05 Pipy 모듈 설치 명령어

| 명령어 | 설명 |
|--|---------------------------------|
| <code>pip install {moduleName}</code> | 이름이 moduleName인 모듈 설치 |
| <code>pip install --update {moduleName}</code> | 이름이 moduleName인 모듈의 최신 버전 업그레이드 |
| <code>pip uninstall {moduleName}</code> | 설치된 이름이 moduleName인 모듈 삭제 |



파이썬기초

NEXT

표준 모듈과
사용자 정의 모듈

