공학적 문제해결 기법

9주차 2차시

코드 재사용하기 2

9주차 1차시 : 코드 재사용하기 1

I. 개요

모듈을 이용한 코드 재사용하기

II. 학습 개요

1) 학습 목표

하나의 파일은 하나의 모듈이다. 여러 개의 모듈로 작성된 프로그램들이 상호 호출하면서 실행되도록 한다.

2) 학습 목차(세부 목차)

- 모듈이란
- 내장된 모듈 사용하기
- 이름 없는 함수

9주차 2차시 : 코드 재사용하기 2

Ⅲ. 학습 1 - 모듈이란

모듈

- "파이썬 프로그램 파일 혹은 C 확장 파일로써 프로그램과 데이터를 정의하고 있으며, 고객이 모듈에 정의된 함수나 변수의 이름을 사용하도록 허용하는 것"
- 정의한 내용은 다른 모듈에 의해서 호출되고 활용
- 코드들을 묶어 재사용 가능하게 만드는 하나의 단위
- 서로 연관된 작업을 하는 코드들의 모임으로 구성
- 작성 중인 모듈의 크기가 어느 정도 커지게 되면 일반적으로 관리 가능한 작은 단위로 다시 분할
- 분리된 모듈은 코드의 독립성을 유지하여 나중에 재사용할 수 있도록 만들어야 함

9주차 2차시 : 코드 재사용하기 2

Ⅲ. 학습 1 - 모듈이란

모듈의 종류

- 표준 모듈 : 파이썬 언어 패키지 안에 포함된 모듈
- 사용자 생성 모듈 : 여러분이 만드는 모듈
- 써드 파티 모듈 : 다른 업체나 개인이 만들어서 제공하는 모듈

모듈은 어디에 저장되는가?

- 모듈이 만들어지는 곳은 파일
- 파일은 코드를 저장하는 물리적인 단위
- 모듈은 하나의 단위로 조직된 코드의 모임으로 논리적 개념
- 파일은 상자이고 모듈은 그 안에 있는 파이썬 코드라고 할 수 있음
- 파이썬 모듈은 .py 파일 확장자를 가짐

9주차 2차시 : 코드 재사용하기 2

Ⅲ. 학습 1 - 모듈이란

함수 사용하기

- 모듈 만들기는 쉽다.
- 여러분이 필요로 하는 변수나 함수를 정의한 파이선 파일을 만드는 것이 전부
- 확장자는 .py

```
#FILE : mymath.py
mypi =3.14

def add(a, b):
    return a+b

def area(r):
    return mypi *r*r
```

9주차 2차시 : 코드 재사용하기 2

Ⅲ. 학습 1 - 모듈이란

사용자 모듈 만들기와 호출하기

- 우리가 배워온 일반 파이썬 파일과 다르지 않다.
- 변수 mypi와 함수 add, area를 정의
- 이 모듈을 대화식 모드에서 호출해 보자.

```
>>> import mymath
>>> dir(mymath)
['__builtins__', '__cached__', '__doc__', '__file__', '__initializing__',
'__loader__', '__name__', '__package__', 'add', 'area', 'mypi']
>>> mymath.mypi
3.14
>>> mymath.area(5)
78.5
```

9주차 2차시 : 코드 재사용하기 2

Ⅲ. 학습 1 - 모듈이란

사용자 모듈 만들기와 호출하기

- 첫 번째 명령 import는 mymath를 현재의 모듈로 가져온다.
- 모듈 이름은 원래 파일 이름에서 .py를 제외한 것과 동일
- 두 번째 명령 dir(mymath) 함수 mymath에 정의된 각종 이름들을 보여 줌
- 앞의 8개는 시스템에서 자동적으로 설정해 주는 이름들
- 마지막 3개('add', 'area', 'mypi')가 우리가 정의한 이름들
- 세 번째 명령 mymath.mypi는 mymath 이름 공간 안의 mypi를 참조
- 네 번째 명령 mymath.area(5)는 mymath 이름 공간 안의 이름 area를 호출
- 모듈 내의 이름을 부를 때 '모듈이름.이름' 형식을 이용

9주차 2차시 : 코드 재사용하기 2

Ⅲ. 학습 1 - 모듈이란

모듈을 왜 사용하는가?

- 모듈은 파일 단위로 함수를 그 영역별로 다시 묶음
- 즉, 비슷한 혹은 관련된 일을 하는 함수나 상수들을 모아서 하나의 파일에 저장
- 나중에 필요할 때 재사용

코드를 재사용할 수 있다.
- 대화식 모드에서 입력된 코드를 나중에 다시 실행하려면 같은 코드를 다시 입력해야 하지만, 모듈을 이용하면 재사용이 가능

모듈은 시스템 구성 요소의 기본 단위
- 모든 것은 모듈 단위로 분리되어 있음(단 하나의 프로그램 파일도 모듈)
- 다른 사람과 함께 프로젝트를 진행한다면 모듈을 최대로 활용해야 할 것

별도의 이름 공간을 제공
- 다른 모듈과 겹치지 않는 공간으로 독립적으로 작업 가능

9주차 2차시 : 코드 재사용하기 2

Ⅲ. 학습 2 - 내장된 모듈 사용하기

내장 모듈

- 모듈(Module)은 함수와 변수 그리고 다른 것들을 더 크고 강력한 프로그램에 넣기 위해서 사용됨
- 어떤 모듈은 내장되어 있으며, 어떤 것들은 별도로 다운로드 받아야 함
- time 이라는 내장 모듈을 이용하여 현재 날짜와 시간을 계산

```
>>> import time
```

- import 명령은 time 모듈을 사용하고 싶다고 알려주는 것
- 점(dot) 기호를 사용하여 이 모듈에 있는 함수들을 호출할 수 있음
- time 모듈로 asctime 함수를 호출해 보면

```
>>> print(time.asctime())
```

- asctime 함수는 time 모듈의 일부이며 현재 날짜와 시간을 문자열로 반환

9주차 2차시 : 코드 재사용하기 2

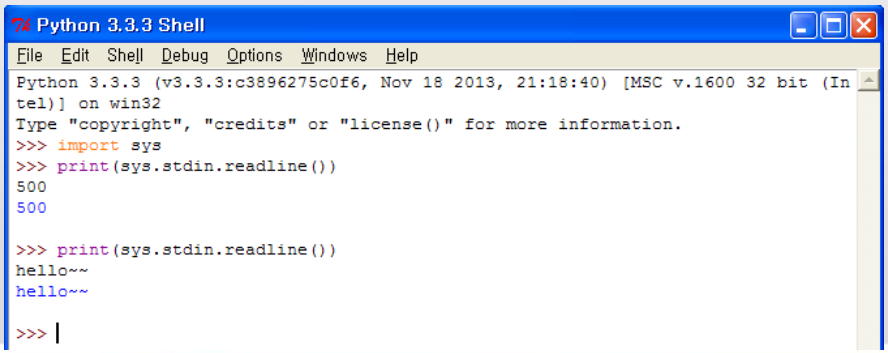
Ⅲ. 학습 2 - 내장된 모듈 사용하기

내장 모듈 - sys(system) 모듈

- 파이썬 시스템이 사용자와 상호연결을 하는데 도움을 주는 모듈
- sys 모듈에는 readline 이라는 stdin(standard input)의 특별한 객체가 있음
- readline 함수는 엔터를 누르기 전까지 키보드로 입력한 텍스트를 읽는데 사용

```
〉〉〉 import sys
〉〉〉 print(sys.stdin.readline())
```

- 어떤 단어를 입력하고 엔터를 누른다면 다음과 같이 그 단어가 출력



9주차 2차시 : 코드 재사용하기 2

Ⅲ. 학습 2 - 내장된 모듈 사용하기

readline 함수를 사용하여 나이를 묻는 함수 만들기

- readline 함수는 사용자가 입력한 것을 문자열로 반환
- int 함수를 이용하여 문자열을 숫자로 변환

```
def your_age():
    print('How old are you? ')
    age = int(sys.stdin.readline())
    if age >= 11 and age <= 19:
        print('teenager? ')
    else:
        print('Huh? ')
```

- 매개변수 없이 이 함수를 호출해 본다.
- How old are you? 라고 나오면 숫자를 입력한다.

9주차 2차시 : 코드 재사용하기 2

Ⅲ. 학습 3 - 이름 없는 함수

람다 함수

- 이름이 없는 한 줄짜리 함수
- 람다 함수를 정의하는 법
  - V lambda 콤마로 구분된 인수들 : 식
  - V 받는 인수가 없고 언제나 1을 리턴

```
>>> f=lambda : 1
>>> f()
1
```

- 람다 함수의 몸체는 문이 아닌 하나의 식
- 값을 리턴하기 위하여 return 문을 사용하지 않음

```
>>> g = lambda x, y : x+y
>>> g(1, 2)
3
```

9주차 2차시 : 코드 재사용하기 2

Ⅲ. 학습 3 - 이름 없는 함수

람다 함수

- 기본값을 갖는 인수

```
>>> incr = lambda x, inc=1 : x+inc
>>> incr(10)
11
>>> incr(10, 5)
15
```

- 값을 리턴하기 위하여 return 문을 사용하지 않음

```
>>> vargs = lambda x, *args: args
>>> vargs(1, 2, 3, 4, 5)
(2, 3, 4, 5)
```

9주차 2차시 : 코드 재사용하기 2

Ⅲ. 학습 3 - 이름 없는 함수

람다 함수

- 람다 함수는 정의하는 즉시 인수로 전달
- 이와 같은 차이는 문(statement)과 식(expression)에서 나옴
- if, for, while, class, def 등 같음 문은 리턴 값이 없으며, 식의 일부분으로 사용될 수 없다.
- 식은 결과 값이 존재하며 다른 식의 일부로 사용
- 람다 함수는 식이기 때문에 정의와 함께 함수 인수로 전달하는 것이 가능

TONGMYONG UNIVERSITY
동명대학교

『 이 콘텐츠는 2014학년도 학부교육 선도대학 육성사업(ACE)에 의하여 개발한 것임 』