

Stage 1 - 모듈과 패키지

모듈과 패키지의 개념에 대해 배워봅시다. 우리가 만든 프로젝트에서 `calculators` 디렉토리를 이용해 모듈과 패키지를 만들어봅시다.

Calculators 패키지 만들기

나만의 프로그램을 만들기 위해서는 모듈과 패키지의 개념을 알아야합니다. 우리가 지금까지 만들었던 여러 프로그램을 하나의 파이썬 파일에서 실행시키기 위한 작업을 합니다. 마치 주머니 여러 개에 핸드폰, 지갑, 손수건 등등의 것을 넣어서 관리하다가 가방을 사서 한 군데에서 물건들을 관리하는 것과 비슷합니다.

`main.py`

```
1 import calculators.grade_calculator
2 import calculators.normal_calculator
3 from calculators import obesity_calculator
4 from calculators.pay_calculator import calculate_pay
5
6
7 print("Hello World!")
8
9 calculators.grade_calculator.calculate_grade()
10 calculators.normal_calculator.calculate_two_numbers()
11 obesity_calculator.calculate_obesity()
12 calculate_pay()
```

모듈과 패키지

모듈이란 무엇인가

레고를 만들어보셨나요? 어린 시절 한번 쯤은 레고 블럭을 맞추며 무언가를 만들어보신 기억이 있습니다. 자동차, 비행기, 보물섬 등 다양한 주제를 가지고 그것들을 만들 수 있도록 알맞은 레고 부품들이 제공이 되죠. 바로 이러한 레고 부품 하나하나를 코딩의 모듈이라 볼 수 있습니다.

모듈이란 구성단위, 구성부품의 뜻이 있습니다. 코딩에서도 모듈이란 하나의 구성단위, 부품을 의미합니다. 우리가 지금까지 만든 파이썬 파일들은 다른 것의 구성부품으로 들어갈 수 없습니다. 각자 파일을 따로따로 실행했던 것 처럼요. 다른 파일에서 부품처럼 우리가 만든 파이썬 파일을 사용할 수 있도록 개선해봅시다.



레고의 모듈

모듈 사용방법

모듈은 어떻게 만드는 것 일까요? 지난 시간에 배웠던 함수 기억나시나요? 바로 함수가 답입니다. 사실 우리는 이미 기존에 잘 만들어진 모듈을 계속해서 사용하고 있었습니다. 예를 들어 랜덤 라이브러리도 모듈화가 되어서 우리가 프로그램을 만들 때 구성부품으로 사용한거죠. 이처럼 우리도 우리가 만든 `calculators` 디렉터리 안의 파일들을 함수로 만들어줍니다. 그리고 그렇게 만든 함수를 사용하고자 하는 파일에서 `'import'`란 명령을 통해 부품으로 가져와 사용합니다.

example

```
1 import random
2 import calculators.pay_calculator
3 import calculators.obesity_calculator
4 import calculators.normal_calculator
```

패키지란 무엇인가

레고 부품이 모듈을 의미한다면, 패키지는 자동차, 비행기, 보물섬 등 이러한 주제를 가지고 레고 부품을 하나의 상품으로 묶은 것입니다.

수능 패키지, 사장님 패키지, 럭셔리 패키지 등 패키지는 어떤 주제를 가지고 사용자들이 이해하기 쉽게 만들어주는 역할을 합니다. 우리도 프로그램을 만들면서 이와 비슷한 기능을 하는 것에 대해 배웠습니다. 바로 디렉터리 입니다. 디렉터리는 파일들을 정리하기 위해 특정한 주제로 파일들을 묶었습니다. 패키지는 디렉터리를 통해 묶인 파이썬 파일들을 디렉터리 단위에서 부터 접근하게 합니다.



레고 배트맨 패키지

패키지 사용방법

모듈화 된 파이썬 파일들을 디렉터리에 정리합니다. 이로써 패키지를 만드는 것은 끝났습니다. 단, 주의할 점은 파이썬 2에서는 디렉터리마다 `'__init__.py'`라는 파일을 생성해주어야 파이썬 패키지 역할을 하는 것을 인지하게 됩니다. 그러나 파이썬 3부터는 단순히 디렉터리만 구성해주어도 패키지의 역할을 할 수 있습니다. 모듈을 `'import'` 사용하여 불러왔다면, 패키지는 `'.'`와 `'from'`을 사용하여 패키지로부터 모듈을 불러올 수 있습니다.

example

```
1 from random import randrange, randint
2 from calculators import pay_calculator.calculate_pay
3 from calculators import obesity_calculator.calculate_obesity
4 from calculators import normal_calculator.calculate
```

Calculators 패키지 만들기

함수로 만들기

각 파일 내의 함수로 처리되지 않은 부분들을 아래와 같이 함수로 처리합니다.

```
calculators/grade_calculator
```

```
1 def calculate_grade():
```

```
calculators/normal_calculator
```

```
1 def calculate_two_numbers():
```

```
calculators/obesity_calculator
```

```
1 def calculate_obesity():
```

```
calculators/pay_calculator
```

```
1 def calculate_pay():
```

import 하기


여러가지 방법으로 import 해보겠습니다. 'main.py' 함수에서 아래와 같이 'import'하고 각 모듈을 사용해봅시다.

main.py

```

1 import calculators.grade_calculator
2 import calculators.normal_calculator
3 from calculators import obesity_calculator
4 from calculators.pay_calculator import calculate_pay

```

 'import calculators' 라고만 선언해주어도 위의 모든 모듈을 '.'를 통해 접근할 수 있습니다. 그러나 사이즈가 큰 패키지의 경우에는 우리가 필요한 모듈만을 가져오기 위해 'from', '.'를 사용하여 특정 모듈만을 'import'하는 연습을 해야합니다.

표기	의
import 모듈이름	불러온 모듈 이름 내부의 모듈을 모두 가져온다.
import 모듈이름.함	불러온 모듈 이름 내부의 함수만 가져온다.
from 모듈이름1 import 모듈이름2.함	모듈이름1로부터 모듈이름2의 함수만 가져온다.
from 모듈이름 import *	모듈이름 안의 모든 모듈을 가져온다.