

codingOn X posco

K-Digital Training 스마트팩토리 개발자 입문

# Python 모듈

# 모듈

# 모듈이란?

- 여러 기능들이 뭉쳐진 하나의 .py 파일
  - 다른 파일에서 import 해서 사용한다
- 코드를 분리하고 공유하는 일
- 함수, 클래스, 변수 등 Python 코드가 저장된 파일
- 표준 모듈: Python 기본 내장
- 외부 모듈: 직접 만든 모듈

# 모듈 생성

- 모듈은 단순 .py 파일을 만들면 됨
- 모듈을 불러올때는 파일명으로 불러오지만 .py를 생략함
- 불러올 파일들은 같은 폴더에 위치해야함

```
#calc.py
def add(a, b):
|     return a + b

def sub(a, b):
|     return a - b

def multiply(a, b):
|     return a * b

def divide(a, b):
|     return a / b
```

```
import calc

# 모듈명.함수명
print(calc.add(5, 2))
print(calc.sub(5, 2))
print(calc.multiply(5, 2))
print(calc.divide(5, 2))
```

# 모듈 불러오기

- import 모듈이름

```
import calc  
import calc as a
```

```
import calc as a  
  
# 모듈명.함수명  
print(a.add(5, 2))  
print(a.sub(5, 2))  
print(a.multiply(5, 2))  
print(a.divide(5, 2))
```

- from 모듈이름 import 시퀀스명

```
from calc import add  
from calc import add as a  
from calc import add, sub  
from calc import *
```

```
from calc import *  
  
# 모듈명.함수명  
print(add(5, 2))  
print(sub(5, 2))  
print(multiply(5, 2))  
print(divide(5, 2))
```

# 표준 모듈

- 파이썬에서 제공하는 모듈 [공식홈페이지방문](#)

모듈	설명
datetime	날짜와 시간의 조작 및 형식화를 제공하는 모듈
time	시간 측정, 대기 및 시스템 시간 관련 기능을 제공하는 모듈
math	수학적 계산을 위한 함수와 상수를 제공하는 모듈
random	난수 생성 및 무작위 작업을 지원하는 모듈
sys	Python 인터프리터와 관련된 정보 및 명령행 인수 처리를 지원하는 모듈
os	운영 체제와 상호작용하며 파일 시스템 조작 및 환경 변수 접근을 지원하는 모듈
json	JSON 데이터를 파싱하거나 Python 객체로 변환하는 기능을 제공하는 모듈

# datetime

```
from datetime import datetime, timedelta

now = datetime.today() # 현재 날짜와 시간(단순한 로컬 시간 계산)
print(now)             # 전체 날짜 및 시간
print(now.year)        # 년
print(now.month)       # 월
print(now.day)         # 일
print(now.hour)        # 시
print(now.minute)      # 분
print(now.second)      # 초

# 날짜 출력
print(f'{now.year}년 {now.month}월 {now.day}일')

# 시간 출력
print(f'{now.hour}시 {now.minute}분 {now.second}초')

# 현재 날짜와 시간 출력
now = datetime.now() # 시간대 처리나 로컬 시간 계산(타임존)
print("현재 날짜와 시간:", now)

# 특정 날짜 계산
next_week = now + timedelta(weeks=1)
print("1주일 후:", next_week)

# 날짜 포맷 변경
formatted_date = now.strftime("%Y-%m-%d %H:%M:%S")
print("포맷팅된 날짜:", formatted_date)
```



# datetime

```
from datetime import date

print("♠ 지금까지 몇 일? ♠")

first_day = date(year=2024, month=11, day=18)
print(first_day) # 시작 날짜 출력

today = date.today()
print(today) # 오늘 날짜 출력
print(date.today().weekday()) # 요일출력 0:월

passed_time = today - first_day
print(passed_time) # 경과 시간 출력 (days 단위)

print(f'개강 이후 {passed_time.days}일 지났습니다.') # 경과 일수 출력
```

# time

```
import time

# 현재 시간 출력
print("현재시간(초):", time.time())
print("년월일시분초: ", time.localtime())

# 2초 동안 대기
print("2초 대기 중...")
time.sleep(2)
print("2초 대기 완료!")

start = time.perf_counter() # 시간 측정
time.sleep(1) # 1초 대기
end = time.perf_counter()
print(f"소요 시간: {end - start:.2f}초")
```

# math

```
import math

# 수학 연산 예제
print(math.pi) # 3.141592653589793
# 제곱근
print(math.sqrt(25)) # 5.0
# 팩토리얼
print(math.factorial(3)) # 6
# 올림
print(math.ceil(2.54)) # 3
# 반올림
print(round(2.6)) # 3
# 버림
print(math.floor(4.9)) # 4
```

# random

```
import random

# randint(a, b): a부터 b까지 임의의 정수 리턴 (a, b 포함)
rand_int = random.randint(1, 10)
print(f"randint: {rand_int}")

# uniform(a, b): a부터 b까지 임의의 실수 리턴 (a, b 포함)
rand_float = random.uniform(1.5, 5.5)
print(f"uniform: {rand_float}")

# random(): 0부터 1 사이의 임의의 실수 리턴 (0 <= x < 1)
rand_between = random.random()
print(f"random(): {rand_between}")

# randrange(a, b): a부터 b 사이의 임의의 정수 리턴 (b 미포함)
rand_range = random.randrange(1000, 10000)
print(f"randrange: {rand_range}")

# random.choice(arr): 리스트에서 무작위로 하나 선택
choices = ['사과', '바나나', '포도', '귤']
rand_choice = random.choice(choices)
print(f"choice: {rand_choice}")
```

# 실습. 로또 번호 뽑기

1. 1~45까지의 수 중에서 랜덤으로 6개의 숫자를 뽑는다
2. 6개의 숫자 중 중복되는 숫자가 없도록 한다.
3. 오름 차순으로 정렬 한다.

실행 결과 📌

```
[1, 12, 16, 26, 29, 44]
```

# sys

```
import sys

# 명령행 인수 출력
print(sys.argv)
print(sys.argv[1:])

# Python 버전 정보 출력
print(sys.version)

# 프로그램 종료
print("프로그램 종료 중")
sys.exit(0)
print("이 문장은 실행되지 않습니다.")
```

```
import os

# 현재 디렉토리 출력
print("현재 디렉토리:", os.getcwd())
file_path = os.chdir(os.getcwd()) # 경로로 이동
dir = os.popen('ls') # 해당폴더 조회
print(dir.read()) # 폴더내용읽기

# 파일/디렉토리 생성
os.mkdir("test")
print("디렉토리 생성: test")

# 파일/디렉토리 삭제
os.rmdir("test")
print("디렉토리 삭제: test")

# 환경 변수 확인
print("PATH 환경 변수:", os.environ.get('PATH'))
```

# json

```
import json

# Python 객체 -> JSON 문자열 변환
data = {
    "name": "홍길동",
    "age": 25,
    "city": "서울"
}
json_str = json.dumps(data)
print("JSON 문자열:", json_str)

# JSON 문자열 -> Python 객체 변환
json_obj = json.loads(json_str)
print("Python 객체:", json_obj)
print(json_obj['name'])
```



# 서드파티 모듈 설치

- 다른 모듈을 설치해서 사용 할 수 있음. 파이썬 커뮤니티에서 다양한 모듈을 제공
- 설치 명령어
  - conda install 모듈명
  - pip install 모듈명

```
conda install numpy  
pip install requests
```

# 실습. 타자연습 게임

## 영어 타자 연습 프로그램

- 게임이 시작되면 영어 단어가 화면에 표시된다.
- 사용자는 최대한 빠르고 정확하게 입력해야 한다.
- 바르게 입력했으면 다음 문제로 넘어가고 “통과”를 출력한다.
- 오타가 있으면 같은 단어가 한 번 더 나온다.
- 종료 조건을 생성하여 종료시 게임시간을 측정한다.

### 출력예시

```
단어: sunlight
입력: sunlighe
오타! 다시 시도하세요.

입력: sunlight
통과!

단어: cloud
입력: exit

게임 종료!
총 2개의 단어를 입력하셨습니다.
총 걸린 시간: 14.31초
평균 단어당 입력 시간: 7.16초
```

복.습.철.저

**수고하셨습니다**