

# JSON을 이용한 직렬화/역직렬화

# JSON을 이용한 직렬화/역직렬화

---

## ❖ JSON

- JavaScript Object Notification
- Javascript Object

```
contact = {  
  name : '홍길동',  
  cell_phone : '010-1111-1111',  
  email : 'hong@naver.com',  
  address : '서울시 강남구',  
  birthday : '2000-12-12'  
}
```

- 파이썬의 dict 자료형과 유사

# JSON을 이용한 직렬화/역직렬화

---

## ❖ JSON 문자열

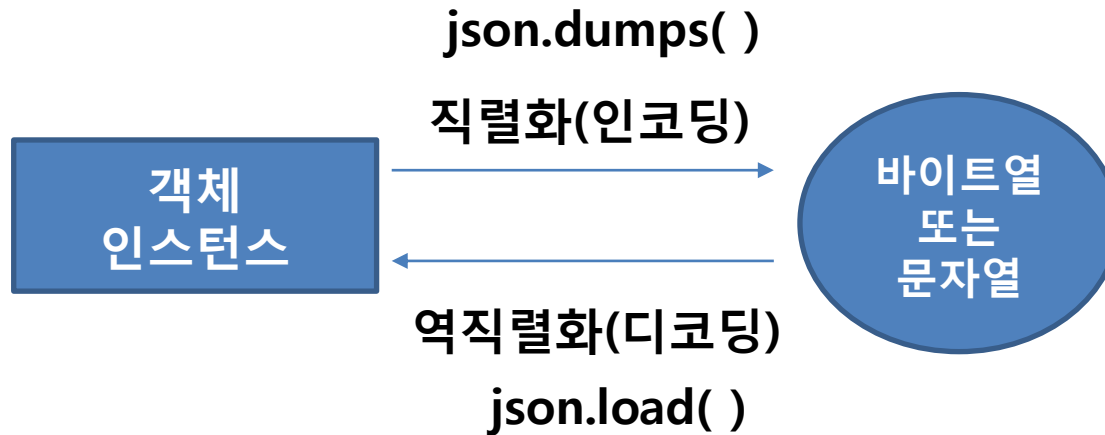
- 인스턴스의 내부 정보(속성 값)을 자바스크립트 객체 표기법으로 문자열로 표현한 것

```
json_str = '''{  
    "name" : "홍길동",  
    "cell_phone" : "010-1111-1111",  
    "email" : "hong@naver.com",  
    "address" : "서울시 강남구",  
    "birthday" : "2000-12-12"  
}'''
```

- 객체(인스턴스) 정보의 저장 및 전송 시 주로 사용

# JSON을 이용한 직렬화/역직렬화

## ❖ 직렬화(Serialize)/역직렬화(Deserialize)



- json 모듈에서 직렬화/역직렬화 지원
  - 파이썬의 기본 자료형에 대해 처리
  - 사용자 정의 자료형(클래스)
    - dict로 변환하여 처리

## JSON을 이용한 직렬화/역직렬화

### ❖ JSON을 이용한 파이썬 클래스의 직렬화 역직렬화



# JSON을 이용한 직렬화/역직렬화

## ❖ 클래스 인스턴스 복원을 위한 JSON 문자열 형식

```
{
  "__class__": "클래스명"    # 클래스명 지정
  "__args__": []             # 클래스 생성자에 전달할 매개변수 리스트
  "__kw__": {                 # 클래스 생성자에 전달할 키워드 매개변수 사전
  }
}
```

- `__class__`
  - 클래스명 지정
- `__args__`
  - 해당 클래스의 생성자에 전달할 위치기반 매개변수로 사용할 값 리스트
- `__kw__`
  - 해당 클래스의 생성자에 전달할 키워드 매개변수로 사용할 값 사전

# JSON을 이용한 직렬화/역직렬화

---

## ❖ Contact의 인코딩 메서드

```
def as_dict(self):  
    return dict(  
        __class__ = "Contact",  
        __args__ = [],  
        __kw__ = dict (  
            name=self.__name,  
            cell_phone=self.__cell_phone,  
            email=self.__email,  
            address=self.__address,  
            birthday=self.__birthday  
        )  
    )
```

## JSON을 이용한 직렬화/역직렬화

### ❖ ContactBook의 인코딩 메서드

```
def as_dict(self):
    return dict(
        __class__ = "ContactBook",
        __args__ = [self.contact_list],
        __kw__ = {}
    )
```

### ❖ ContactBook의 생성자 수정

```
class ContactBook:
    def __init__(self, constact_list=None):
        if constact_list:
            self.contact_list = constact_list
        else:
            self.contact_list = [
                Contact('홍길동', '010-1111-2222', 'hong@naver.com'),
                Contact('고길동', '010-1111-3333', 'gogd@naver.com'),
                ...
            ]

        self.contact_list.sort(key=sort_by_name)
```



# JSON을 이용한 직렬화/역직렬화

---

## ❖ ContactApp의 저장 기능 정의

```
from ContactBook import ContactBook
from Contact import Contact
import json

contact_book = ContactBook()
:

# 객체 인코딩 호출 함수
def contact_encode(obj) :
    return obj.as_dict()

def save():
    json_str = json.dumps(contact_book, indent=4, default=contact_encode)
    with open('contact.dat', 'w') as file:
        try:
            file.write(json_str)
        except Exception as err:
            print('contact.dat 파일 저장 에러 : %s'%err)
```

# JSON을 이용한 직렬화/역직렬화

---

## ❖ ContactApp의 저장 기능 정의

```
def main():
    while True:
        print_menu()
        :
        elif (command == 'X'):
            save()
            print('종료합니다.')
            break
        else:
            print('잘못된 메뉴입니다. 다시 입력하세요')

if __name__ == '__main__':
    main()
```

# JSON을 이용한 직렬화/역직렬화

---

## ❖ ContactApp의 저장 기능 정의

- 람다 사용하기
  - 1줄로 구성되는 함수를 인라인 정의로 사용
  - **lamda (매개변수 리스트):결과값 표현식**
    - 매개변수가 1개인 경우 () 생략 가능
    - return 키워드 생략

```
def save():  
    json_str = json.dumps(contact_book, indent=4,  
                           default=lambda obj: obj.as_dict())  
    with open('contact.dat', 'w') as file:  
        try:  
            file.write(json_str)  
        except Exception as err:  
            print('contact.dat 파일 저장 예러 : %s'%err)
```

# JSON을 이용한 직렬화/역직렬화

## ❖ ContactApp의 읽기 기능 정의

```
:
# contact_book = ContactBook()
contact_book = None
:
def contact_decode(obj):
    if set(obj.keys()) == set(['__class__', '__args__', '__kw__']):
        cls = eval(obj['__class__'])
        return cls(*obj['__args__'], **obj['__kw__'])
    else:
        return obj

def load():
    global contact_book
    with open('contact.dat', 'r') as file:
        try:
            json_str = file.read()
            contact_book = json.loads(json_str, object_hook=contact_decode)
        except Exception as err:
            print('contact.dat 파일 읽기 예러 : %s'%err)
```

# JSON을 이용한 직렬화/역직렬화

---

## ❖ ContactApp의 읽기 기능 정의

```
def main():
    load()
    while True:
        print_menu()
        command = select_menu()
        if(command == 'P'):
            contact_book.print()
            :
        elif (command == 'X'):
            save()
            print('종료합니다.')
            break
        else:
            print('잘못된 메뉴입니다. 다시 입력하세요')

if __name__ == '__main__':
    main()
```