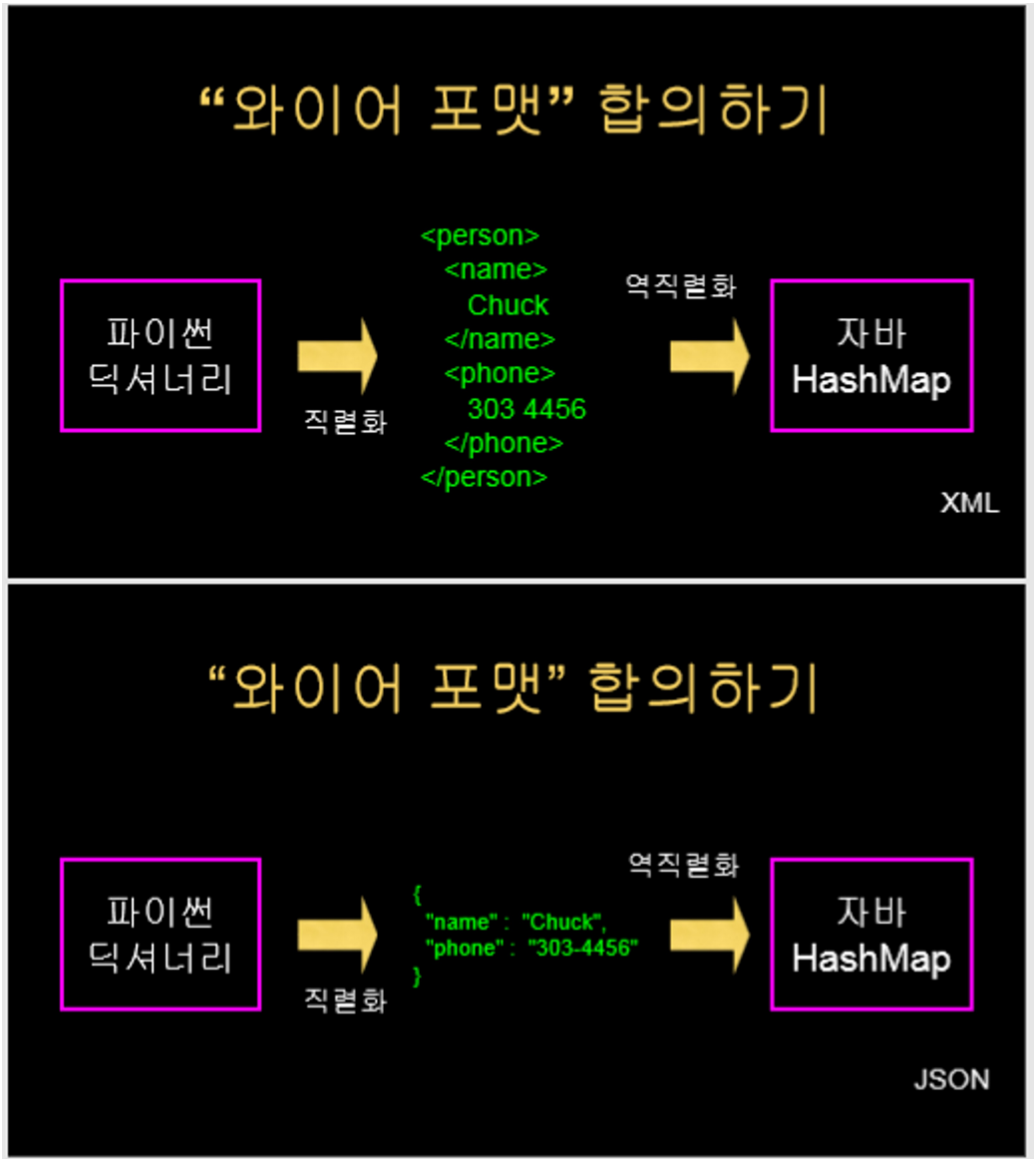
# xml

2019년 6월 5일 수요일 오후 3:27

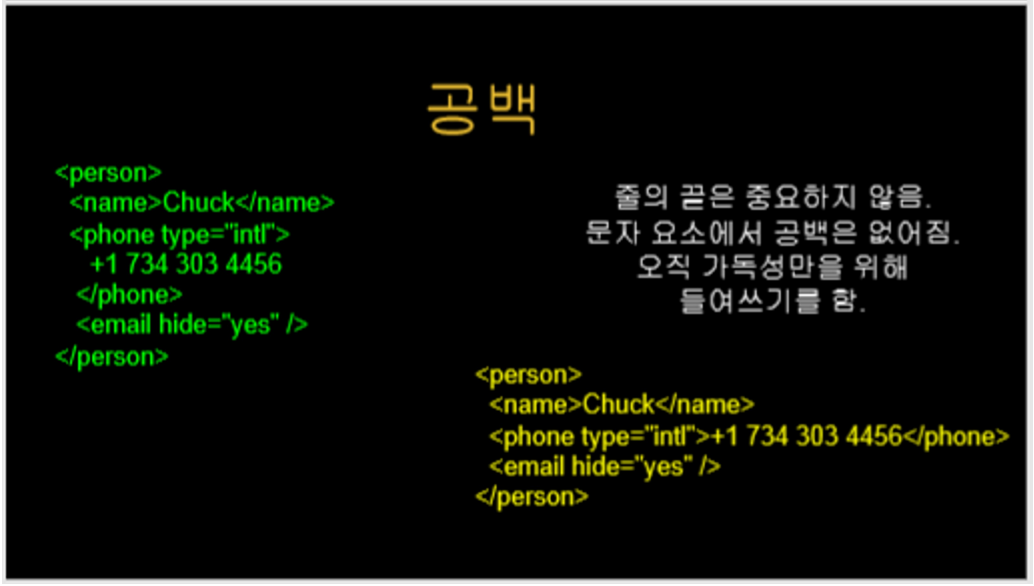
**웹 상의 데이터**

• HTTP 요청과 응답에 대한 이해와 지원을 바탕으로, 이 프로토콜을 이용한 프로그램 간의 정보 교환의 추세
• 네트워크와 응용프로그램 간의 데이터 표현 방식에 있어서 합의가 필요
• 가장 널리 사용되는 두 가지 포맷: XML\과 JSON




**XML**

XML은 eXtensible Markup Language의 약자로, 다음과 같은 계층 구조로 이루어져있습니다. HTML과 비슷하지만 원하는 이름의 태그를 만들 수 있다는 특징이 있고, HTML보다 문법 오류를 더 엄격하게 다룹니다.

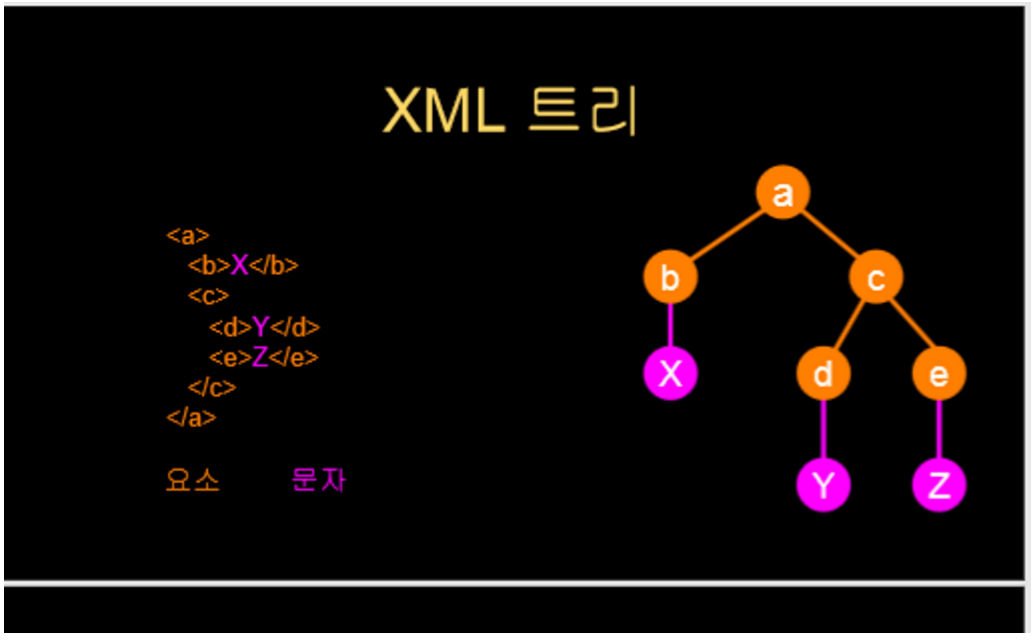



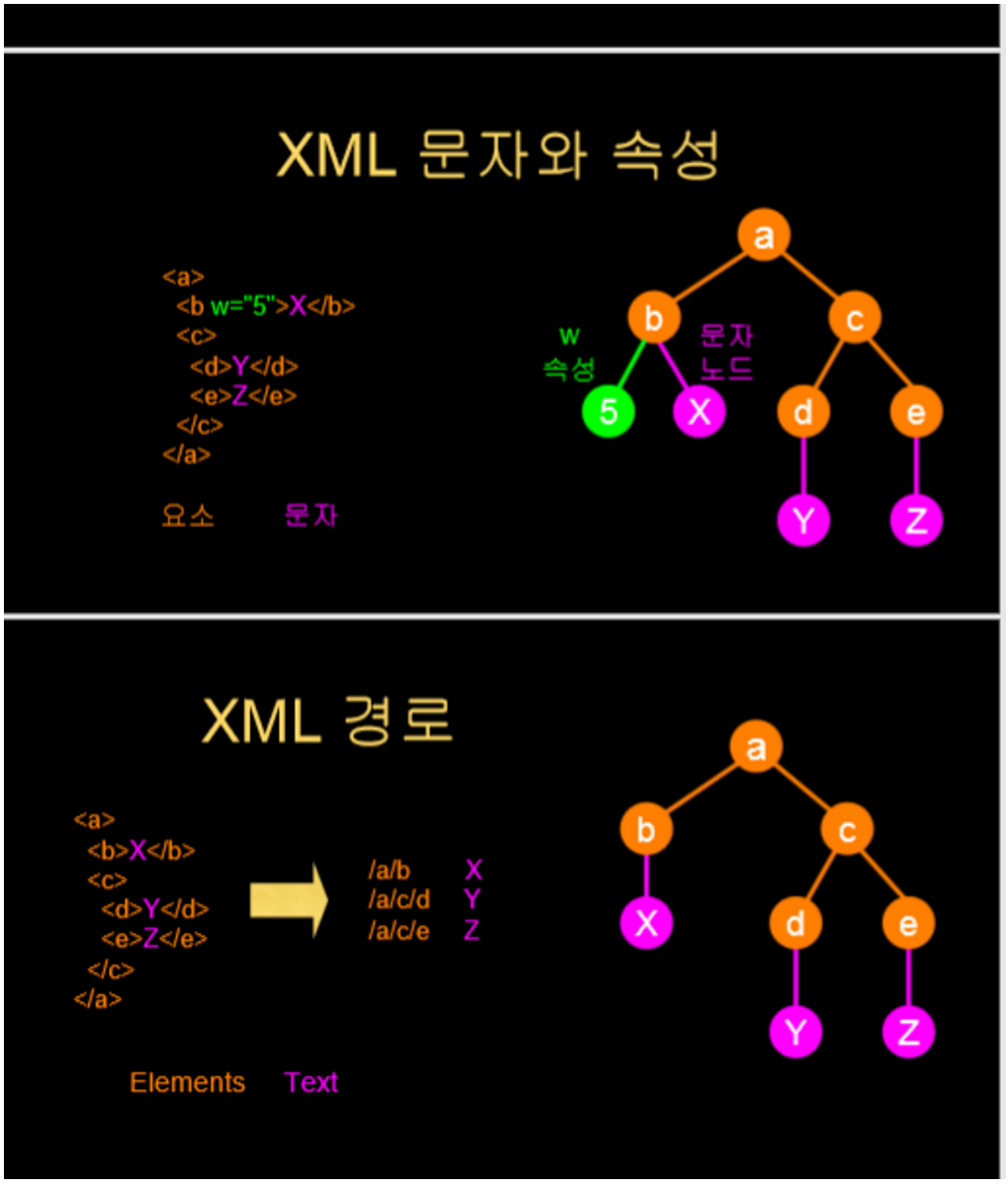

**XML 용어**

• 태그 Tags 는 요소의 시작과 끝을 알려줌

• 속성 Attributes - XML의 여는 태그에 위치한 키-값 쌍

• 직렬화 Serialize / 역직렬화 De-Serialize - 한 프로그램의 데이터를 특정 프로그램 언어에 제한되지 않은 채로 시스템 내에서 저장되고 전달되어질 수 있는 형식으로 변환하는 것






- 실습

## 최종 코드

```
import xml.etree.ElementTree as ET

input = '''
<stuff>
  <users>
    <user x="2">
      <id>001</id>
      <name>Chuck</name>
    </user>
    <user x="7">
      <id>009</id>
      <name>Brent</name>
    </user>
```

```
  </users>
</stuff>'''

stuff = ET.fromstring(input)
lst = stuff.findall('users/user')
print('User count:', len(lst))

for item in lst:
    print('Name', item.find('name').text)
    print('Id', item.find('id').text)
    print('Attribute', item.get("x"))
```

# 학습 내용

**XML**

XML 데이터를 파이썬에 읽어오기 위해서는 xml 모듈이 필요합니다.

다음과 같은 함수를 활용하면 XML에 접근해 원하는 데이터를 추출할 수 있습니다.

```
import xml.etree.ElementTree as ET
data = '''<person>
  <name>Chuck</name>
  <phone type="intl">
     +1 734 303 4456
   </phone>
   <email hide="yes"/>
</person>'''

tree = ET.fromstring(data)
print('Name:',tree.find('name').text)
print('Attr:',tree.find('email').get('hide'))

# Name: Chuck
# Attr: yes
```

조금 더 복잡하지만 XML의 구조를 이해하고 있으면 다음과 같이 반복문을 활용해 XML의 데이터에 접근할 수도 있습니다.

```
import xml.etree.ElementTree as ET
input = '''<stuff>
   <users>
      <user x="2">
         <id>001</id>
         <name>Chuck</name>
      </user>
      <user x="7">
         <id>009</id>
         <name>Brent</name>
      </user>
   </users>
</stuff>'''
```

```
stuff = ET.fromstring(input)
lst = stuff.findall('users/user')
print('User count:', len(lst))
for item in lst:
    print('Name', item.find('name').text)
    print('Id', item.find('id').text)
    print('Attribute', item.get("x"))
```

```
#User count: 2
#Name Chuck
#Id 001
#Attribute 2
#Name Brent
#Id 009
#Attribute 7
```

출처: <https://www.edwith.org/python-network-data/lecture/24468/>

# json

2019년 6월 5일 수요일 오후 3:41

**JavaScript Object Notation**

- Douglas Crockford - JSON을 “발견”
- 자바스크립트의 객체 표현 방식
- JSON 은 데이터를 중첩된 “리스트”와 “딕셔너리” 로 표현

## 학습 내용

**JSON(JavaScript Object Notation)**

JSON은 XML보다 더 자주 사용되는 데이터 포맷입니다. 이 코드는 이전 시간에 XML로 실행했던 것과 정확히 같은 내용의 코드입니다. 데이터가 XML 형식에서 JSON 형식으로 바뀐 것을 제외하면 말입니다.

JSON은 파이썬에서의 딕셔너리와 굉장히 비슷하기 때문에 데이터를 읽어온 후 딕셔너리로 접근할 수 있습니다.

```
import json
data = '''{
  "name" : "Chuck",
  "phone" : {
    "type" : "intl",
    "number" : "+1 734 303 4456"
   },
   "email" : {
     "hide" : "yes"
   }
}'''

info = json.loads(data)
print('Name:',info["name"])
print('Hide:',info["email"]["hide"])

#Name: Chuck
#Hide: yes
```

이와 같이 여러 개의 데이터를 읽어올 경우 리스트에 딕셔너리가 포함된 형태로 읽어집니다.

```
import json
input = '''[
  { "id" : "001",
    "x" : "2",
    "name" : "Chuck"
  } ,
  { "id" : "009",
    "x" : "7",
    "name" : "Chuck"
  }
]'''
```

---

```
info = json.loads(input)
print(info)
print('User count:', len(info))
for item in info:
    print('Name', item['name'])
    print('Id', item['id'])
    print('Attribute', item['x'])
```

```
# [{'id': '001', 'x': '2', 'name': 'Chuck'}, {'id': '009', 'x': '7', 'name': 'Chuck'}]
# User count: 2
# Name Chuck
# Id 001
# Attribute 2
# Name Chuck
# Id 009
# Attribute 7
```

출처: <https://www.edwith.org/python-network-data/lecture/24469/>

# SOA

2019년 6월 5일 수요일 오후 3:42

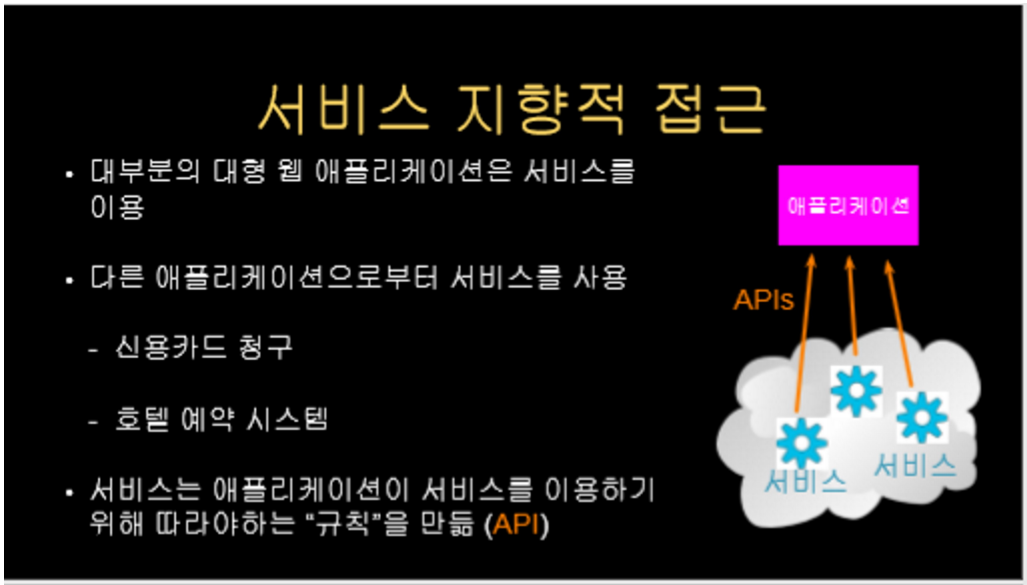
## 학습 내용

**Service Oriented Approach**

**서비스 지향 아키텍처**(Service Oriented Architecture, 약칭 SOA「에스오에이」혹은「소아」로 발음)란 대규모 컴퓨터 시스템을 구축할 때의 개념으로 업무상에 일 처리에 해당하는 소프트웨어 기능을 서비스로 판단하여 그 서비스를 네트워크상에 연동하여 시스템 전체를 구축해 나가는 방법론이다.

* 출처 : 위키백과, '서비스 지향 아키텍처'

출처: <https://www.edwith.org/python-network-data/lecture/24471/>

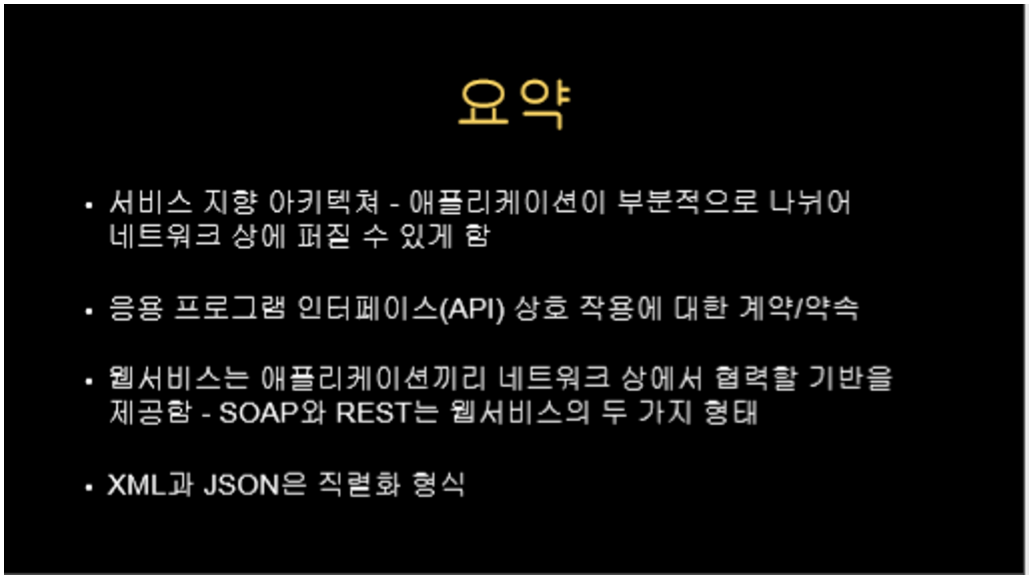



# API

2019년 6월 5일 수요일 오후 3:47

- 응용 프로그램 인터페이스(API)
  http://en.wikipedia.org/wiki/API
  API는 인터페이스를 지정하고 그 인터페이스의 객체의 행동을 제어한다는 점에서 매우 추상적. API에 명시된 기능을 제공하는 소프트웨어를 API 의 "실행"이라고 하며, API는 대체로 애플리케이션을 구성하게 되는 언어로 정의됨.

  API 보안과 비율 제한
  • API를 실행하기 위한 계산 자원은 "무료"가 아님
  • API를 통해 제공된 데이터는 대체로 매우 가치가 높음
  • 데이터의 제공자는 하루 요청량을 제한하여서 API 의 "키"를 요구하거나, 사용료를 부과하기도 함
  • 발전을 거치면서 여러 규칙들이 바뀌기도 함




- 구글지도 api
- 트위터 api 실습 해보기