스크립트 텀프로젝트

국내 / 외 여행 정보 어플리케이션

- ALL IN TRIP -

2013180035 정승혜
2014184007 김예솔

#1. 현재까지 구현 한 함수 - xml

```
from xml.dom.minidom import parse
from xml.etree import ElementTree
from http.client import HTTPConnection

class Functions:
    #이 함수는 파일에서 불러오기용
    def __init__(self,Name,subName,regKey,**items):
        #Name은 "apis.data.go.kr"
        #subName은 그 뒤에 오는
        #/1262000/CountrySafetyService/getCountrySafetyList 이런거
        #regkey는 등록키
        #items는 pageNo = 999, NumOfRaw = 999이런 식으로 넣으면 됨
        self.conn = None
        self.subName = subName
        #두개는 생성자 오버로딩으로 regkey가 있다면
        #reg키를 넣는다.
        self.fileName = Name
        self.regKey = regKey
        #openAPI에서 필요한 URL을 만들기 위해서 diction형태로 초기화할때 넣어준다.
        self.items = items
```

이 함수들을 import 해서 사용

1. Name 서버주소
2. subName 그 외의주소
3. regKey본인 키
4. **items 페이지 수 등 필요한 추가 정보

#1. 현재까지 구현 한 함수 - menu

```python
def printMenu1():
    init()
    print("All In Trip Menu!")
    print("1.국내")
    print("2.국외")
    print("3.종료")
    print("-----------------------------------")
    funcComm = int(input())
    if funcComm == 1:
        printMenuLocal()
    elif funcComm == 2:
        selectContinent()
    elif funcComm == 3:
         QuitBookMgr()
#=============================================================================

#국내1
def printMenuLocal():
    print("All In Trip 국내 메뉴입니다.")
    print("1.서울")
    print("2.경기")
    print("3.인천")
    print("4.강원도")
    print("5.충청도")
    print("6.전라도")
    print("7.경상도")
    localNum = int(input())
    #위에 정보 저장하고 메뉴로 넘어가야할듯한
    if 0<localNum and localNum <8:
        printMenuLocalDetail(localNum-1)
    else:
        printMenuLocal()

#-----------------------------------------------------------------------------
```

While문을 돌려 메뉴를 계속
이용할 수 있도록 함

앞에서 구현한 기능들에 파일 이름 입력
〉〉 자세한부분 영상 참조

#1. 현재까지 구현 한 함수 - xml

```
def extractBookData(self,treename,*itemName):
    tree = ElementTree.fromstring(self.req.read())
    #print(self.req.read())
    # Book 엘리먼트를 가져옵니다.
    itemElements = tree.getiterator(treename)  # return list type

    print(itemElements)
    for item in itemElements:
        ls = []
        for x in itemName:
            ls.append(item.find(x))
       #print (strTitle)
        for x in ls:
            if len(x.text) >0:
                print(x.text,end = " ")
        print("")
```

불러올 트리와 그 밑에 있는
자식 트리들을 읽어 올 수 있음

이름을 불러오는 것이기 때문에
어느 트리라도 읽기 가능

```
def extractcountryData(self,treename,country,*itemName):
    tree = ElementTree.fromstring(self.req.read())
    itemElements = tree.getiterator(treename)  # return list type

    print(itemElements)
    for item in itemElements:
        #cont = item.find("countryName")
        if country == item.find("countryName").text:
            ls = []
            for x in itemName:
                ls.append(item.find(x))
       #print (strTitle)
            for x in ls:
                if len(x.text) >0:
                    print(x.text,end = " ")
            print("")
```

위와 동일한 부분
Country는 국외 부분에서 나라이름을 받아
와 나라이름과 일치하면 그 정보를 출력

#1. 현재까지 구현 한 함수 - xml

```
def extractBookData(self,treename,*itemName):
    tree = ElementTree.fromstring(self.req.read())
    #print(self.req.read())
    # Book 엘리먼트를 가져옵니다.
    itemElements = tree.getiterator(treename)  # return list type

    print(itemElements)
    for item in itemElements:
        ls = []
        for x in itemName:
            ls.append(item.find(x))
       #print (strTitle)
        for x in ls:
            if len(x.text) >0:
                print(x.text,end = " ")
        print("")
```

불러올 트리와 그 밑에 있는
자식 트리들을 읽어 올 수 있음

이름을 불러오는 것이기 때문에
어느 트리라도 읽기 가능

```
def extractcountryData(self,treename,country,*itemName):
    tree = ElementTree.fromstring(self.req.read())
    itemElements = tree.getiterator(treename)  # return list type

    print(itemElements)
    for item in itemElements:
        #cont = item.find("countryName")
        if country == item.find("countryName").text:
            ls = []
            for x in itemName:
                ls.append(item.find(x))
       #print (strTitle)
            for x in ls:
                if len(x.text) >0:
                    print(x.text,end = " ")
            print("")
```

위와 동일한 부분
Country는 국외 부분에서 나라이름을 받아
와 나라이름과 일치하면 그 정보를 출력

#1. 현재까지 구현 한 함수 - xml

```
overseasAsia = []
overseasAsiad = dict()
overseasEU = []
overseasEUd = dict()
overseasAmerica = []
overseasAmericad = dict()
overseasAfrica = []
overseasAfricad = dict()
def saveOverseas(self):
    tree = ElementTree.fromstring(self.req.read())
    itemElements = tree.getiterator("item")
    for item in itemElements:
        continent = item.find("continent")
        country = item.find("countryName")
        if continent.text == "아시아/태평양":
            overseasAsia.append(country.text)
        elif continent.text == "유럽":
            overseasEU.append(country.text)
        elif continent.text == "미주":
            overseasAmerica.append(country.text)
        elif continent.text == "중동/아프리카":
            overseasAfrica.append(country.text)

    for x in range(len(overseasAsia)):
        overseasAsiad[x+1] = overseasAsia[x]

    for x in range(len(overseasEU)):
        overseasEUd[x+1] = overseasEU[x]

    for x in range(len(overseasAmerica)):
        overseasAmericad[x+1] = overseasAmerica[x]

    for x in range(len(overseasAfrica)):
        overseasAfricad[x+1] = overseasAfrica[x]
```

대륙 별 나라 정보를 사전으로 저장

대륙 별 나라 정보를 리스트로 추가해 사전으로 변경하는 작업

초기에 한번만 불러 줌

#2. 현재까지 구현 한 함수 – excel

```
from openpyxl import load_workbook

class ExcelImport:

    def __init__(self, filename): #클래스 생성할 때 이름 넣어주기
        self.filename = filename

    def loadFile(self): #파일 불러오기
        self.wb = load_workbook(self.filename)
        sheetList = self.wb.get_sheet_names()
        self.sheet = self.wb.get_sheet_by_name(sheetList[0])
        self.maxCol = self.sheet.max_column
        self.maxRow = self.sheet.max_row;
        self.keys = []
        self.datas = []
        for x in range(1,self.sheet.max_column+1):
            self.keys.append(self.sheet.cell(row = 1, column = x).value)
        for x in range(2,self.sheet.max_row+1):
            data = []
            for y in range(1,self.sheet.max_column+1):
                data.append(self.sheet.cell(row = x,column = y).value)
            self.datas.append(data)
```

Openyxl 함수 설치해서 이용

파일 이름.형식 으로 값만 넣어주면
모든 엑셀 파일 이용 가능

#2. 현재까지 구현 한 함수 – excel

```
from openpyxl import load_workbook

class ExcelImport:

    def __init__(self, filename): #클래스 생성할 때 이름 넣어주기
        self.filename = filename

    def loadFile(self): #파일 불러오기
        self.wb = load_workbook(self.filename)
        sheetList = self.wb.get_sheet_names()
        self.sheet = self.wb.get_sheet_by_name(sheetList[0])
        self.maxCol = self.sheet.max_column
        self.maxRow = self.sheet.max_row;
        self.keys = []
        self.datas = []
        for x in range(1,self.sheet.max_column+1):
            self.keys.append(self.sheet.cell(row = 1, column = x).value)
        for x in range(2,self.sheet.max_row+1):
            data = []
            for y in range(1,self.sheet.max_column+1):
                data.append(self.sheet.cell(row = x,column = y).value)
            self.datas.append(data)
```

Openyxl 함수 설치해서 이용

파일 이름.형식 으로 값만 넣어주면
모든 엑셀 파일 이용 가능

#2. 현재까지 구현 한 함수 – excel

```
def printInfo(self):#정보 출력하기
    for x in range(0,self.maxRow-1):
        for y in range(0,self.maxCol):
            print("{0}:{1}".format(self.keys[y],self.datas[x][y]), end = ' ')
            print("")

def filterPrint(self,idx,Contents): #정보가 몇번째 정보인지 알고 비교할 내용을 넣어주면 자동으로 필터링하여 출력
    for x in range(0,self.maxRow-1):
        for y in range(0,self.maxCol):
            if self.datas[x][idx] == Contents:
                print("{0}:{1}".format(self.keys[y],self.datas[x][y]), end = ' ')
        if self.datas[x][idx] == Contents:
            print("")
```

파일의 모든 데이터를 출력

원하는 열 정보의 인덱스 값과
해당 열에서 원하는 키워드를 입력하면
그 키워드에 해당되는 행의
데이터를 모두 출력

#3. 보완해야할 점

```
def extractcountryData(self,treename,country,*itemName):
    tree = ElementTree.fromstring(self.req.read())
    itemElements = tree.getiterator(treename)  # return list type

    print(itemElements)
    for item in itemElements:
        #cont = item.find("countryName")
        if country == item.find("countryName").text:
            ls = []
            for x in itemName:
                ls.append(item.find(x))
       #print (strTitle)
            for x in ls:
                if len(x.text) >0:
                    print(x.text,end = " ")
            print("")
```

현재 트리를 읽는 함수가 한번 찾으면

더이상 읽어주지를 않아서 이부분 보완 후

국가 안전 부분에 적용 할 예정입니다.

```
def filterPrint(self,idx,Contents): #정보가 몇번째 정보인지 알고 비교할 내용을 넣어주면 자동으로 필터링하여 출력
    for x in range(0,self.maxRow-1):
        for y in range(0,self.maxCol):
            if self.datas[x][idx] == Contents:
                print("{0}:{1}".format(self.keys[y],self.datas[x][y]), end = ' ')
        if self.datas[x][idx] == Contents:
            print("")
```

키워드가 완전히 똑같지 않아도

그 단어를 포함하고 있으면

정보를 출력하게 하도록 수정할 예정입니다.

#4. 개발 일정

	추가 해야 할 기능
6 주차	보완해야할 점 수정 및 기능들 최종 구현
7 주차	UI 추가
8 주차	디버깅 및 오류 수정

감사합니다