3주

1주차

 Python Programming 리뷰

 Html, css selector , RDB(SQL), Web 에 기본개념

2주차 (1.27, 28 / 2.7~2.9) 5일

Web Scraping - python, html,css, db, web 기본개념

 :  web 상에 있는 데이터를 수집 , 분석, DB에 저장

3주차(2.10, 11 / 2.14 ~ 16) 5일

Django

:  Dynamic 한 Web Application  작성

   static / dynamic

**Mini Project 제출기한은 2월 24일(목)까지**

: Django 를 활용한 Mini Project

: Web scraping 활용(선택)

: 제출은 어디에 하는지에 대한 정보

  e-ncs.kr 사이트에 인터페이스 개발 프로젝트에 응시하셔서  압축 파일 업로드

: source 압축, 실행된 화면 캡쳐된 image 파일,

  프로젝트 설명서 문서양식은 제공

홍길동

   01프로젝트기획안\_요구사항명세서.docx

   화면캡쳐

       실행된 화면 을 캡쳐된 image 파일들

   django 프로젝트 폴더

홍길동 폴더를 압축해서 ncs 에 올리기

2월 21일 ~ 4월14일

: 리눅스, 도커, 쿠버네티스, 퍼블릭클라우드

4월15일 ~ 6월28일

클라우드 프로젝트

Java - 상속, 인터페이스, 다형성 : 개인적으로 선행 필수

javascript  : 개인적으로 선행 필수

MSA

스프링부트, JPA,  스프링클라우드,

Reactjs, Vuejs - ECMAScript

https://www.youtube.com/playlist?list=PLfI752FpVCS8tDT1QEYwcXmkKDz-\_6nm3

Pycharm Professional Edition

: html,css, js, python, sql

MariaDB

설치시 루트 패스워드 maria / maria

**인코딩 방식  utf-8 반드시 체크**

**port 3306**

Pycharm

: Anaconda python.exe

vuejs/ reactjs

: ecmascript6

list\_test.py

list\_test.ipynb (ipython notebook)

python reserved word

제어문

if , elif, else, : 조건문

for, while : 반복문

break, continue,

연산자

: and, not, or

예외처리(Exception 처리)

: try, except, finally, raise

모듈

: import, from

함수와 클래스 선언

: def , return, lambda, class

etc

: as,  is, in, with , pass, del

전역변수

: global,  nonlocal

쓰레드

: yield

단위테스트

: assert

내부함수 데이터 타입 변환

int() : 정수

float() : 실수

str() : 문자열

list() : 리스트  [ ]

dict() : 딕셔너리 { }

tuple() : 리스트(read only)  ( )

set() : 집합(unique value)

html reference

<https://www.w3schools.com/tags/default.asp>

css reference

<https://www.w3schools.com/cssref/default.asp>

Http 프로토콜

GET / POST 방식

GET - 서버의 데이터를 조회할때 사용하는 http method

클라이언트에서 보내는 데이터를 URL에 append해서 보낸다.

viewer?id=101&name=ps

querystring 형식 : key=value&key=value

POST - 서버의 데이터를 변경할때 사용하는 http method

: 클라이언트에서 보내는 데이터를 Request body로 포함시켜서 보낸다.

Http 프로토콜은

* connection-oriented 연결지향 : TCP
* connectionless 비연결지향 : HTTP

    : 사용자의 정보가 계속 유지되도록 해주는 Cookie, Session 이 필요함

SSR(Server Side Rendering)

: 서버사이드 렌더링

: JSP, FreeMarker, Thymeleaf(타임리프), PHP, ASP#NET

=> Server Side Script 기술

2000년대 초반 Ajax 개념등장

Ajax(Asynchronous Javascript and Xml)

: 비동기, 자바스크립트, Xml(JSON)

: 비동기 통신을  javascript XHR(XmlHttpRequest) 객체가 담당한다.

Sync / Async

CSR(Client Side Rendering)

: 클라이언트 사이드 렌더링

: jquery의 $.ajax()

: Reactjs, Vuejs, Angular

REST(Representational State Transfer) 아키텍처

: open api

json, csv, xml 형식으로 내부의 데이터를 누구나 사용할 수 있도록 공개

: http uri + http method

: http uri 에서는 공유하는 Resource(자원)에 대한 정보를 기술하고

: Resource(자원)들을 제어할때 아래와 같은 다양한 http method를 사용

GET (조회),

POST(등록),

PUT(모든 항목 수정),

PATCH(부분수정),

DELETE(삭제) 등 여러가지  http method를 사용함

전체조회 : GET /users

1명의 User 조회 : GET /users/{id}

등록 : POST /users

수정 : PUT /users/{id} , PATCH /users/{id}

1명의 User 삭제 : DELETE /users/{id}

전체삭제 : DELETE /users

데이터 수집, 분석, 저장 할때 사용되는 Python Third-Party(외부)Libraries

<https://pypi.org/>

: python 외부 라이브러리들을 검색, 등록 할 수 있다.

: python package index

1. Requests

https://github.com/psf/requests

<https://docs.python-requests.org/en/latest/>

https://docs.python-requests.org/en/v0.4.1/api/

Http 통신을 할 수 있도록 해주는 Library

UrlLib 라는 내부모듈도 있음

2. BeautifulSoup4

<https://github.com/waylan/beautifulsoup>

<https://www.crummy.com/software/BeautifulSoup/>

https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Html(Xml) Parser 역할(pulling data out of HTML and XML files)을 하는 Library

3. Selenium

<https://selenium-python.readthedocs.io/>

Web Driver 라는 프로그램을 사용해서 http통신과 html parsing을 하는 Library

web driver download

<https://sites.google.com/chromium.org/driver/downloads>

4. Pandas

데이터 분석(Analysis)

<https://github.com/pandas-dev/pandas>

<https://pandas.pydata.org/docs/user_guide/10min.html>

10분 완성 Pandas

<https://dataitgirls2.github.io/10minutes2pandas/>

PyCon Korea 동영상

https://www.youtube.com/c/PyConKRtube

5. Matplotlib 와 Seaborn

데이터 시각화(Visualization)

<https://github.com/matplotlib/matplotlib>

<https://matplotlib.org/>

<https://github.com/mwaskom/seaborn>

<https://seaborn.pydata.org/>

6. Pymysql

MySql(MariaDB)와 연결해서 python 코드상에서 SQL문 작성

<https://github.com/PyMySQL/PyMySQL>

<https://pymysql.readthedocs.io/en/latest/>

7. SqlAlchemy

개발자가 SQL문을 작성하지 않고도 , 객체를 Table과 매핑해줄 수 있음

ORM(Object Relational Mapping) 기능을 제공하는 Library

Object (python, java 객체)

Relational - DB의 Table

: JPA(Hibernate), SqlAlchemy, Django ORM

<https://github.com/sqlalchemy/sqlalchemy>

<https://www.sqlalchemy.org/>

웹스크래핑 / 웹크롤링

1. 네이버

: 뉴스, open api (Papago), 웹툰

1. 기상청 날씨 데이터

http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp

1. Melon 100 Chart - DB저장
2. 행정구역 정보 분석
3. 국회의원 현황
4. Selenium을 사용한 Daum 포털 로그인
5. 건강검진 데이터 시각화
6. 서울 따릉이 데이터 분석

# 41개의 location 정보를 저장할 list

location\_list = []

#1개의 location 정보를 저장할 dict

location\_dict = {}

#1개의 location 안에 포함된 13개의 data 정보를 저장할 list

data\_list = []

#1개의 data 정보를 저장할 dict

data\_dict = {}

<location wl\_ver="3">

<province>서울ㆍ인천ㆍ경기도</province>

<city>서울</city>

<data>

<mode>A02</mode>

<tmEf>2022-01-31 00:00</tmEf>

<wf>맑음</wf>

<tmn>-6</tmn>

<tmx>4</tmx>

<reliability/>

<rnSt>0</rnSt>

</data>

<data>

<mode>A02</mode>

<tmEf>2022-01-31 12:00</tmEf>

<wf>구름많음</wf>

<tmn>-6</tmn>

<tmx>4</tmx>

<reliability/>

<rnSt>40</rnSt>

</data>

</location>

<location>

</location>

# 서울 city의 날씨

{

  “province”:”서울ㆍ인천ㆍ경기도”,

   “city”:”서울”,

   “datas”:[{“mode”:”A02”,”tmEf”:”2022-01-31 00:00”,”wf”:”맑음”},

          {“mode”:”A02”,”tmEf”:”2022-01-31 12:00”,”wf”:”구름많음”}

         ]

}

location\_list 안에 41개의 city가 있음

[

   {},

   {city:””},

]

Spaghetti Code

MVC 패턴 - 아키텍쳐 패턴

: Separation of concerns(responsibility)

: 관심사의 분리 (역할의 분리)

: MVC(Model / View / Controller)

Model - DB연동

View - 화면

Controller - View와 Model사이의 중간자

MTV 패턴

Model - DB연동

Template - 화면

View - Model과 Template사이의 중간자, 로직

ORM

Class ⇔ Table

Object(Instance) ⇔ Row(Record)

Field(변수,속성) ⇔ Column

Django 마이그레이션 명령어

> python manage.py makemigrations blog

> python manage.py migrate blog

> python manage.py showmigrations blog

> python manage.py sqlmigrate blog 0002

{ } : 중괄호, bracelet , dict

[ ] : 대괄호, bracket, list

( ) : 괄호 , tuple

두번째 Post(글)의 title, text, created\_date, User의 username, email 를 출력해 보세요

Post의 title 필드에 ‘글’ 포함된 Post를 찾아서 create\_date 날짜가 내림차순

조회

모델클래스명.objects.all()

          .objects.get()

  .objects.filter()

get\_object\_or\_404(클래스명,pk=1)

Post.objects.all()

User.objects.all()

Post.objects.get(id=3)

Post.objects.get(title=’sample’)

Post.objects.filter(title\_\_contains(‘글’))

Post.objects.filter(created\_date\_\_lte=timezone.now())

get\_object\_or\_404(Post,pk=3)

객체.id / 객체.pk

등록

모델클래스명.objects.create(title=’test’,text=’aa’)

삭제

객체.delete()

**Project**

mydjango/urls.py

urlpatterns =

[ path('admin/', admin.site.urls),

  path('blog', include('blog.urls')),

  path('product', include('product.urls')),

]

**blog app**

blog/urls.py

urlpatterns =

 [

   path('list', views.post\_list, name='post\_list'),

   path('detail/:id', views.post\_detail, name='post\_detail'

 ]

**product app**

product/urls.py

 urlpatterns =

 [ path('lists', views.product\_list,name='prod\_list'), ]

<http://localhost:8000/admin>

[http://localhost:8000](http://localhost:8000/admin)/blog/list

[http://localhost:8000](http://localhost:8000/admin)/blog/detail/1

[http://localhost:8000](http://localhost:8000/admin)/product/lists

https://docs.djangoproject.com/en/3.1/ref/request-response/

mydjango

   blog

     templates

         blog

            post\_list.html

   product

     templates

          product

            product\_list.html

post 목록

<http://localhost:8000/blog>

mydjango/urls.py

    -> blog/urls.py

         ->blog/views.py (post\_list() 함수)

             ->blog/templates/blog/post\_list.html

post 상세정보

<http://localhost:8000/blog>/post/5

post\_list.html {% url ‘post\_detail’ pk=post.pk %}

‘post\_detail’는 blog/urls.py 의 urlpatterns의 path()

path('post/<int:pk>', views.post\_detail, name='post\_detail'),

Django 3.1 Template Engine

<https://docs.djangoproject.com/en/3.1/topics/templates/>

Django 3.1 Template Tag와 Filter

<https://docs.djangoproject.com/en/3.1/ref/templates/builtins/>

서버사이드 렌더링

* 서버에서 DB의 데이터를
* html 과 django template tag를 사용해서 그려주는 방식

클라이언트 사이드 렌더링

* 서버에서 DB의 데이터를 JSON 형태로 제공하고
* 클라이언트(리액트,vuejs) 에서 동적으로 화면을 그려주는 방식

pycharm

alt + enter : import

ctrl + alt + shift + l(엘) : Code 정렬

ctrl + alt + v : 리턴타입과 변수 자동생성

Project Name : customerprj

App : customer

superuser : django / django

Customer(고객)

* 이름 name : CharField
* 생년월일 birthdate : DateField
* 이메일주소 email : EmailField
* 성별 gender : BooleanField

admin 페이지에서 데이터 등록

queryset 사용해서 조회,등록

urls

views

Pagination docs

[**https://docs.djangoproject.com/en/3.1/topics/pagination/**](https://docs.djangoproject.com/en/3.1/topics/pagination/)

https://bootswatch.com/