


# Venn Relation Define/Find


<https://github.com/kellysolow/vennRelation>

 kellysolow / vennRelation

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

master 1 branch 0 tags

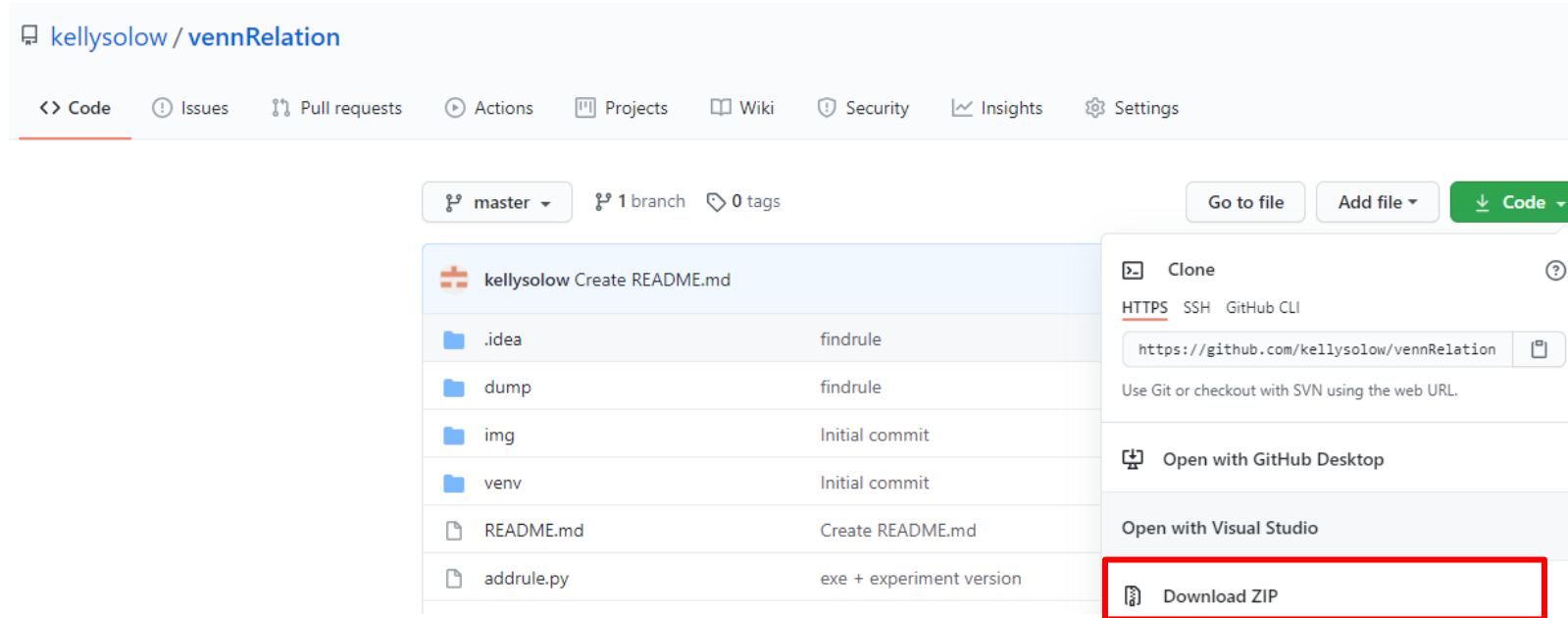
Go to file Add file Code

 kellysolow Create README.md 9a6cf7a 3 minutes ago 4 commits

.idea	findrule	16 hours ago
dump	findrule	16 hours ago
img	Initial commit	3 days ago
venv	Initial commit	3 days ago
README.md	Create README.md	3 minutes ago
addrule.py	exe + experiment version	2 hours ago
addrule2.py	exe + experiment version	2 hours ago
findrule.py	exe + experiment version	2 hours ago
main.exe	exe + experiment version	2 hours ago
main.py	Initial commit	3 days ago
mainUI.py	exe + experiment version	2 hours ago
vennUI.py	exe + experiment version	2 hours ago

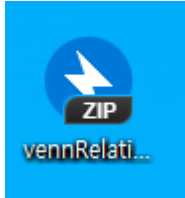
# 프로그램 설치 및 실행과정-1

<https://github.com/kellysolow/vennRelation>



위 링크로 접속 후 우하단의 Download ZIP 클릭

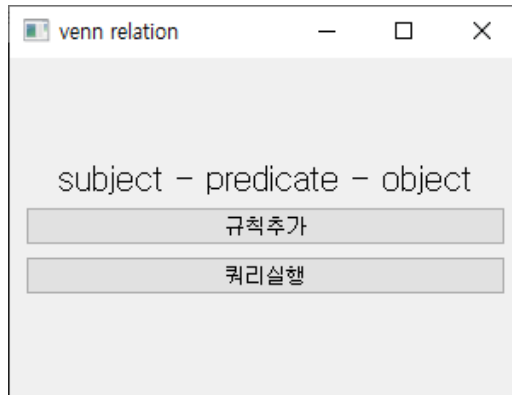
# 프로그램 설치 및 실행과정-2



다운받은 파일의 압축해제

이름	수정된 날짜	유형	크기
.idea	2021-03-24 오후 4:06	파일 폴더	
dump	2021-03-24 오후 4:06	파일 폴더	
img	2021-03-24 오후 4:06	파일 폴더	
venv	2021-03-24 오후 4:06	파일 폴더	
addrule.py	2021-03-24 오후 4:06	Python File	4KB
addrule2.py	2021-03-24 오후 4:06	Python File	3KB
findrule.py	2021-03-24 오후 4:06	Python File	10KB
main.exe	2021-03-24 오후 4:06	응용 프로그램	35,429KB
main.py	2021-03-24 오후 4:06	Python File	1KB
mainUI.py	2021-03-24 오후 4:06	Python File	2KB
README.md	2021-03-24 오후 4:06	MD 파일	1KB
vennUI.py	2021-03-24 오후 4:06	Python File	3KB

main.exe 실행  
(방화벽에 의해 실행 차단될시 추가정보 - 허용)

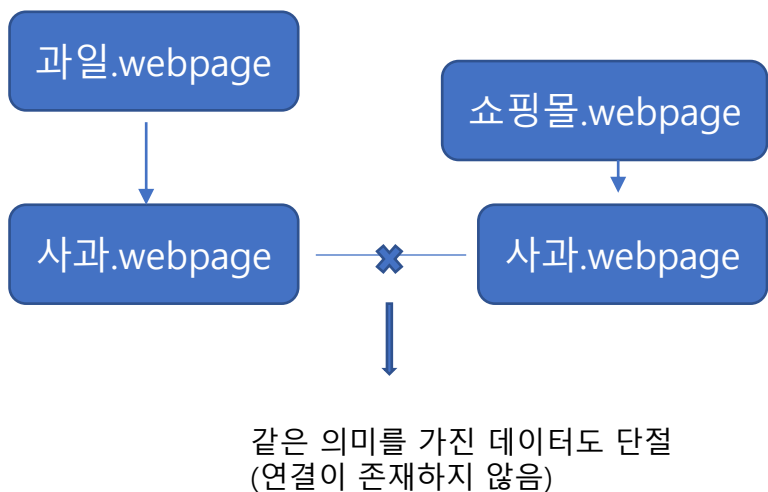


실행성공시 화면

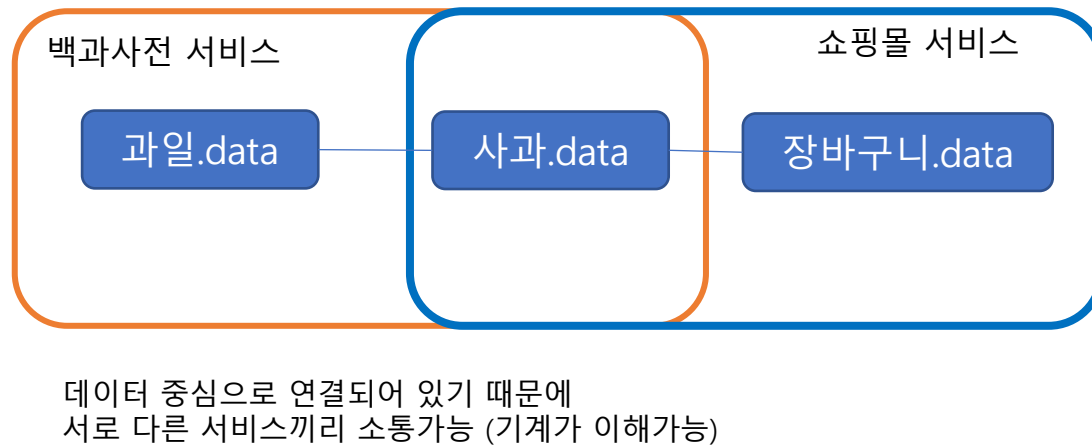
# 본 프로그램의 목적

- '시맨틱 웹'이라는 새로운 웹표준은 기계가 이해하는 웹을 목표로 함
- 기계가 이해하기 위해서 웹의 구조를 문서중심에서 데이터중심으로 변경을 필요로 함

기존 웹의 구조



시맨틱 웹의 구조



웹의 구조에 대한 자세한 이해 필요없이 시맨틱 웹형식의 데이터를 만들어 보는 프로그램

# 프로그램의 목적 -2

- '시맨틱 웹' 식으로 데이터 만들기

- 주어 – 서술어 – 목적어 (subject predicate object)순으로 구성되며 이를 트리플(triple)이라고 함

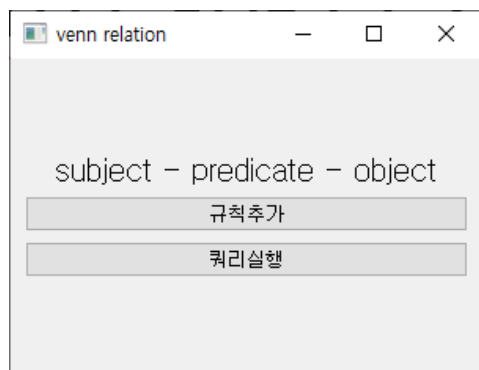
Ex) 쇼핑물 – 상품소개 – 사과

과일 – 종류 – 사과

- 기존 시맨틱웹상에서 위와 같은 관계를 구성할때에는 서술어에 따라 직접 연결방식을 다르게 구성해야함
- 본 프로그램에서는 관계정의할 주어와 목적어를 선택하고 해당 관계에 어울리는 벤다이어그램을 선택하는 것으로 관계정의가 완료됨.

# 프로그램 관계 정의 따라해보기

0. 관계를 만들 subject와 object 정하기.  
(ex. fruit – fruitType – apple)

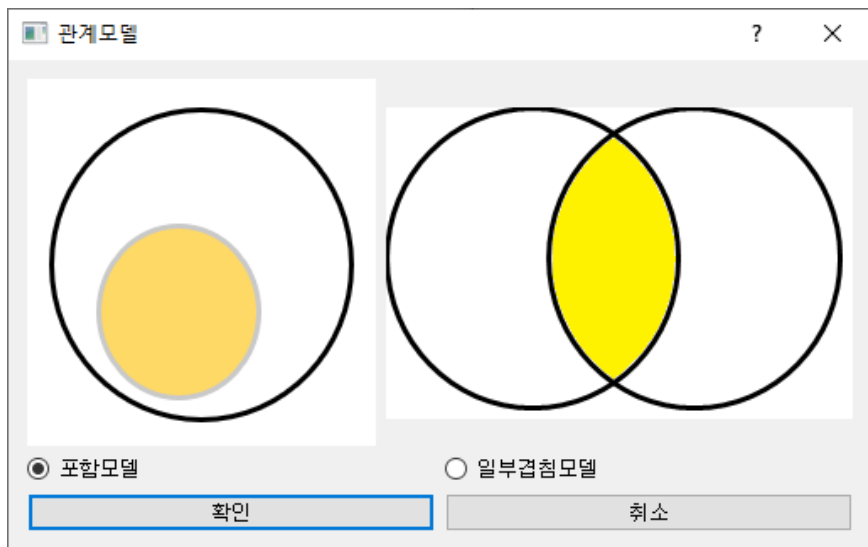


1. 규칙추가 버튼 클릭하기

2. subject와 object 관계에 어울리는 모델 선택하기.

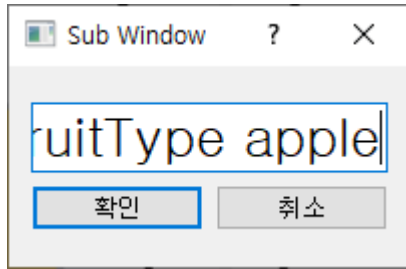
<포함모델>  
큰원 – subject  
작은원 – object

<일부겹침모델>  
좌측원 – subject  
색칠한부분 – object  
우측원 – predicate에 대응하는 object 집단



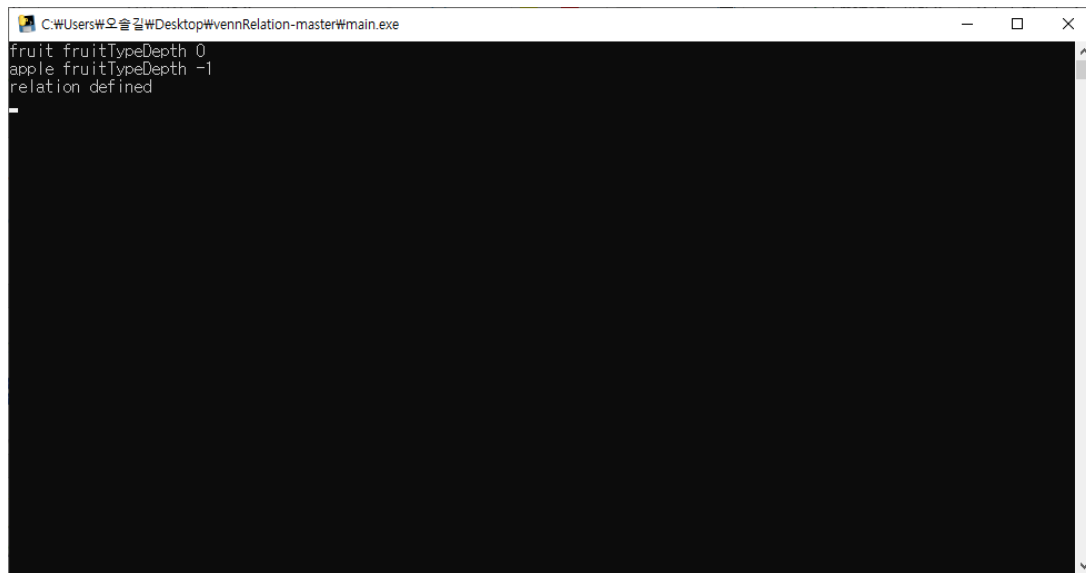
Apple은 fruit에 포함되므로 '포함모델' 선택

# 프로그램 관계 정의 따라해보기-2



## 3. 정의할 관계 작성 (subject predicate object)

단 fruit – fruitType – apple에서 –를 제외하고 띄어쓰기만 사용해서 fruit fruitType apple 작성하기



## 4. 함께 팝업된 console창에서 관계가 정의되었음을 확인 정의한내용은 시맨틱 웹 data dump에 추가됨.

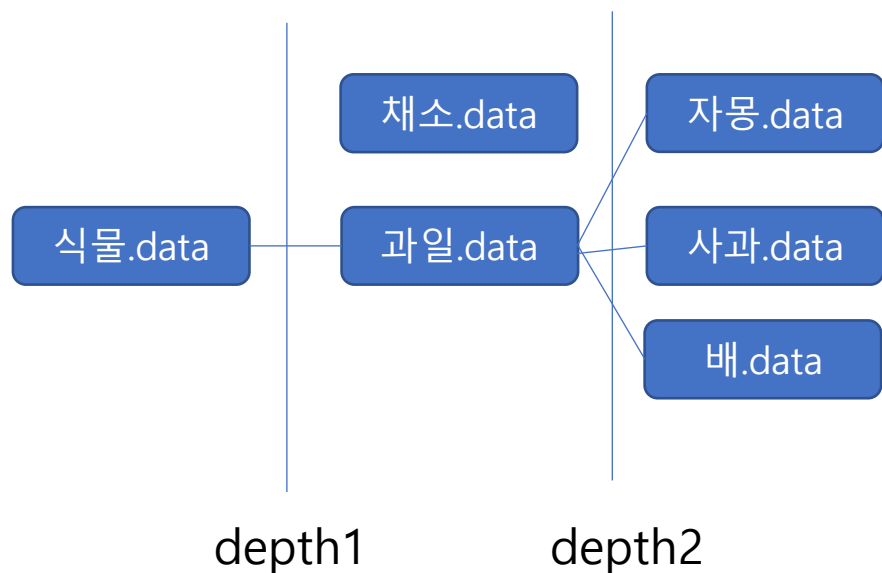
# 프로그램의 목적 -3

- '시맨틱 웹' 식으로 정의된 데이터의 연결관계 검색가능  
ex) 과일, 사과 데이터가 서로 연결되어있는지 확인가능
- Ex) 식물- 하위분류 - 과일 이라는 연결관계가 존재하였을 경우

식물-(하위분류-과일-하위과일)-사과

와 같이 여러단계의 관계도 연결되어있음 확인가능

(트리플 하나당 depth 1단계라고 표현)



```
ArLone_Township,_Pine_County,_Minnesota
ArLone_Township,_Pine_County,_Minnesota <http://www.opengis.net/gml/_Feature>
ArLone_Township,_Pine_County,_Minnesota "46.01888888888889"^^<http://www.w3.org/2001/XMLSchema#float>
ArLone_Township,_Pine_County,_Minnesota "-92.725"^^<http://www.w3.org/2001/XMLSchema#float>
ArLone_Township,_Pine_County,_Minnesota
Arna_Township,_Pine_County,_Minnesota
Arna_Township,_Pine_County,_Minnesota "9.8E7"^^<http://www.w3.org/2001/XMLSchema#double>
Arna_Township,_Pine_County,_Minnesota "9.79015505707008E7"^^<http://www.w3.org/2001/XMLSchema#double>
Arna_Township,_Pine_County,_Minnesota "9.75E7"^^<http://www.w3.org/2001/XMLSchema#double>
Arna_Township,_Pine_County,_Minnesota "9.73835529486336E7"^^<http://www.w3.org/2001/XMLSchema#double>
Arna_Township,_Pine_County,_Minnesota "500000.0"^^<http://www.w3.org/2001/XMLSchema#double>
Arna_Township,_Pine_County,_Minnesota "517997.6220672"^^<http://www.w3.org/2001/XMLSchema#double>
Arna_Township,_Pine_County,_Minnesota "311.0"^^<http://www.w3.org/2001/XMLSchema#double>
Arna_Township,_Pine_County,_Minnesota "310.896"^^<http://www.w3.org/2001/XMLSchema#double>
Arna_Township,_Pine_County,_Minnesota "86"^^<http://www.w3.org/2001/XMLSchema#nonNegativeInteger>
Arna_Township,_Pine_County,_Minnesota "0.9"^^<http://www.w3.org/2001/XMLSchema#double>
Arna_Township,_Pine_County,_Minnesota "0.8880349646476255"^^<http://www.w3.org/2001/XMLSchema#double>
```

(데이터 덤프의 일부분)



# 프로그램 검색 기능 따라해보기 -1

addrule.py	2021-03-24 오후 4:06	Python File	4KB
addrule2.py	2021-03-24 오후 4:06	Python File	3KB
entitiesCount.xlsx	2021-03-24 오후 5:43	Microsoft Excel ...	3,989KB
findrule.py	2021-03-24 오후 4:06	Python File	10KB
main.exe	2021-03-24 오후 4:06	응용 프로그램	35.429KB

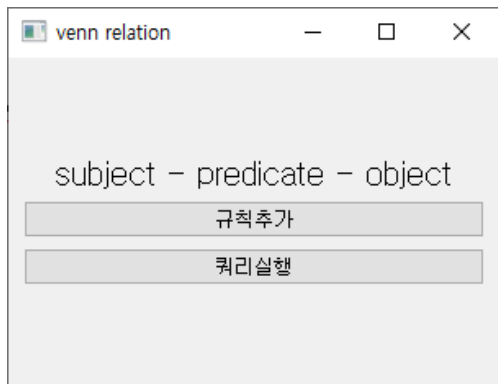
1. 검색하기 위해서 현재 데이터 덤프에 있는 데이터를 출현빈도순으로 정렬한 entitiesCount 파일 열어보기

1	United_States	20821
2	<http://www.opengis.net/gml/_Feature>	20669
3	0.0^^<http://www.w3.org/2001/XMLSchema#double>	16201
4	Central_Time_Zone_(North_America)	10492
5	Eastern_Time_Zone	6726
6	City	5830
7	Township_(United_States)	2820
8	Minnesota	2719
9	Census-designated_place	2510
10	2010^^<http://www.w3.org/2001/XMLSchema#gYear>	2293
11	Michigan	1902
12	Town	1730
13	Mountain_Time_Zone	1599
14	Illinois	1441
15	Central_European_Time	1423
16	Pacific_Time_Zone	1371
17	California	1359
18	258998.8110336^^<http://www.w3.org/2001/XMLSchema#double>	1090
19	Missouri	1049
20	Iowa	1036
21	1.0^^<http://www.w3.org/2001/XMLSchema#double>	978
22	Florida	957

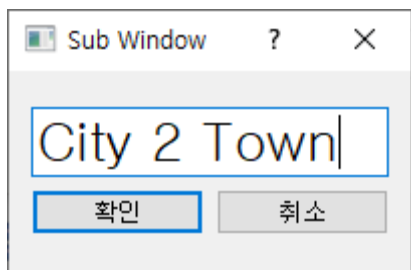
2. 웹상에 관계가 존재하는지 확인하고 싶은 두 개의 entity 선택

(ex City, Town)

# 프로그램 검색 기능 따라해보기 -2



3. 쿼리실행 버튼 클릭하기



4. 관계를 확인하고 싶은 두개의 entity와 depth를 좌측의 그림처럼 기입하기  
(ex. [entity1] [depth] [entity2] 형식으로 기재)  
(depth 1당 1000mb 가량의 메모리 소요를 필요로  
너무 높은 depth에 대한 검색은 수행어려움)

```
depth 1 searching...
depth 2 searching...
```

5. 콘솔창에 별다른 관계가 표현되지않으면 관계 미발견

# 프로그램 응용방안

- 벤다이어그램을 이용해서 직접 정의한 entity도 검색해 볼수 있음

```
apple 는 fruit에 포함됨  
fruit 0  
apple -1
```

(fruit 2 apple 의 검색결과)

- 직접 정의한 entity 와 data dump를 혼용해서 검색가능 (ex fruit 4 City)
- 본 프로그램을 이용해서 정의된 관계는 depth를 활용하기 때문에 기존 dump에 존재하는 것보다 더 빠르게 검색가능.