

#### 공개 소스 라이센스 계약에 따라 개발한 소프트웨어를 다른 사용자가 사용 가능

- 파이썬 : 널리 사용되는 개방형 소스 개발 프로젝트
- 다양한 사용자층과 커뮤니티가 활성화되어 있음
- ➡ 대표적으로 파이썬 패키지 인덱스(PyPI)로 불리는 라이브러리 저장소 관리 기구가 있음



[https://pypi.org/]

# 1 외부 모듈이란?



4차 산업혁명 관련 IT 분야에서 파이썬의 중요성이 높아지고 있는 이유

파이썬이 해당 분야의 프로그래밍에 도움되는
 외부 모듈을 다양하게 제공하기 때문

#### 데이터 분석, 통계

Numpy, Pandas, Matplotlib ···

#### 인공지능

Tensorflow,
PyTorch,
Keras ···

#### 웹 크롤링

BeautifulSoup, Selenium ···

- 2 외부 모듈 설치하기
  - 1 패키지 관리자 사용하기(PIP)
    - 파이썬 3.4 이후 버전은 기본적으로 pip를 포함하고 있으며, 손쉽게 외부 모듈을 설치할 수 있음
  - 2 콘솔창에서 pip install 패키지명 한 줄로 외부 모듈 설치 가능

[zoostar@~\$pip install pandas

Requirement already satisfied: pandas in ./anaconda/li (0.23.0)

Requirement already satisfied: python-dateutil>=2.5.0 6/site-packages (from pandas) (2.6.0)

Requirement already satisfied: pytz>=2011k in ./anacon

# 2 외부 모듈 설치하기

3

Jupyter Notebook 환경에서 명령어 앞 !를 붙여설치 가능

- 느낌표(!)를 붙이면 콘솔창에서 입력하는 것과 같은
   역할
- 삭제는 pip uninstall 모듈명

```
In [2]: !pip install pandas

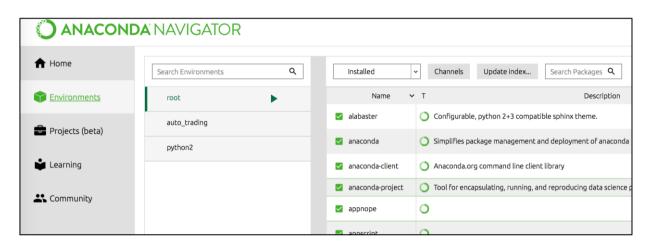
Requirement already satisfied: panda:
Requirement already satisfied: pytho:
ndas) (2.6.0)

Requirement already satisfied: pytz>:
.10)

Requirement already satisfied: numpy:
```

# 2 외부 모듈 설치하기

- 4
- Anaconda를 설치했다면, Anaconda Navigator에서 설치 가능
- 환경을 달리 정의할 수 있고, 환경마다 패키지(외부 모듈)를 설치할 수 있음



- 2 외부 모듈 설치하기
  - 일반 사용자들이 개발한 모듈도 사용 가능
    - 사용자 정의 모듈과 같은 방법으로,
       git 등의 경로에서 다운 받은 파일을 로컬 경로에 저장



1 데이터베이스 연결하기

pymysql

MySQL을 파이썬에서 사용할 수 있는 모듈 중 하나



pip install pymysql로 설치



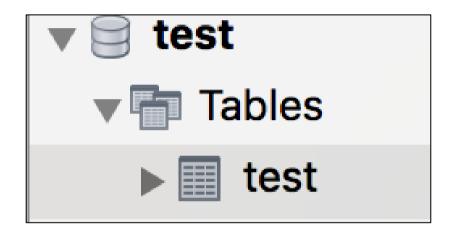
MySQL 설치 (https://dev.mysql.com/downloads/mysql/)

 전 세계에서 가장 많이 사용되는 오픈 소스 관계형 데이터베이스

!pip install pymysql

Requirement already satisfied: pymysql in

- 1 데이터베이스 연결하기
  - 1 데이터베이스 서버 환경
    - test db에 test 테이블 생성



• test 테이블 구성

id	name	age	enable
1	파이썬	10	1
NULL	NULL	NULL	NULL

# 1 데이터베이스 연결하기

1

#### 데이터베이스 서버 환경

#### 스키마 생성 코드(Mysql에 붙여 넣어 실행)

CREATE DATABASE 'test' /\*!40100 DEFAULT CHARACTER SET latin1 \*/;

#### 테이블 생성 코드(Mysql에 붙여 넣어 실행)

CREATE TABLE 'test' (

'id' int(11) NOT NULL AUTO\_INCREMENT,

'name' varchar(500) CHARACTER SET utf8 DEFAULT NULL,

'age' int(11) DEFAULT NULL,

'enable' varchar(45) DEFAULT '0',

PRIMARY KEY ('id')

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

# 1 데이터베이스 연결하기



모듈 import, 연결 정보 정의 (서버 주소, 포트번호, 아이디, 비밀번호)

```
import pymysql

dbURL="localhost"
dbPort=3306
dbUser="root"
dbPass="1234"
```

```
conn = pymysql.connect(
   host=dbURL, port=dbPort, user=dbUser, passwd=dbPass, db='test',
   charset='utf8', use_unicode=True)
```

# 1) 데이터베이스 연결하기

3

데이터 가져오기

```
sql = "SELECT * FROM test.test"
cur = conn.cursor()
cur.execute(sql)
result = cur.fetchall()
conn.commit()
```

id	name	age	enable
1	파이썬	10	1
2	프로그래밍	15	0
NULL	NULL	NULL	NULL

```
print(result)
print(type(result))

((1, '파이썬', 10, '1'), (2, '프로그래밍', 15, '0'))
<class 'tuple'>
```

# 1) 데이터베이스 연결하기



#### 데이터 입력하기

```
sql = "INSERT INTO test (name, age, enable) VALUES (%s, %s, %s);"
cur = conn.cursor()
cur.execute(sql,("홍길동","20","1"))
conn.commit()
```

id	name	age	enable
1	파이썬	10	1
2	프로그래밍	15	0
3	홍길동	20	1
NULL	NULL	NULL	NULL

pymysql 모듈을 가져온 뒤 SQL 쿼리문과 DB 연결, 쿼리 실행 함수만 적어주면 데이터베이스에 입력, 수정, 삭제, 출력 가능

2 웹 크롤링하기

웹 크롤링

컴퓨터 소프트웨어 기술로 웹 사이트들에서 원하는 정보를 추출하는 것



파이썬에서 웹 크롤링을 지원하는 대표 라이브러리 BeautifulSoup 설치

pip install beautifulsoup4로 설치

!pip install beautifulsoup4

Requirement already satisfied: beautifulsoup4

- 2 웹 크롤링하기
  - 포털 사이트 급상승 검색어 가져오기

#### urllib 모듈로 웹 페이지 가져오기

- 파이썬에서 웹과 관련된 작업을 도와주는 모듈
- 데이터를 가져오는 역할을 하는 urllib 중 request 모듈을 활용해 포털 사이트의 메인 페이지 정보를 가져옴

```
import urllib.request
web page = urllib.request.urlopen("https://www.naver.com")
print(web_page.read())
```

b'<!doctype html>\n\n\n\n\n\n\n\n\n\n\n\n\n\n\h\chtml lang="ko

- 2 웹 크롤링하기
  - 포털 사이트 급상승 검색어 가져오기
    - 2 BeautifulSoup 모듈로 웹 페이지 분석
      - 텍스트화된 웹 페이지는 BeautifulSoup 모듈을 활용하면 HTML 태그 단위로 분석 가능

- 2 웹 크롤링하기
  - 포털 사이트 급상승 검색어 가져오기
    - 3 급상승 검색어 부분의 패턴 찾기 (브라우저의 페이지 소스보기)
      - 각각의 순위가 (ul class="ah\_l")에 들어간 것을
         확인

```
446 <h3 class="blind">급상승 검색어 검색어</h3>
447 <div class="ah roll area PM CL realtimeKeyword rolling">
448 
449 
450 <a href="#" class="ah a" data-clk="lve.keyword">
451 <span class="ah r">1</span>
452 <span class="ah k">마인</span>
453 </a>
  455 
456 <a href="#" class="ah a" data-clk="lve.keyword">
457 <span class="ah r">2</span>
458 <span class="ah k">정다은</span>
459 </a>
  461 
462 <a href="#" class="ah a" data-clk="lve.keyword">
  <span class="ah r">3</span>
  <span class="ah k">동국여지승람
```

- 2 웹 크롤링하기
  - 포털 사이트 급상승 검색어 가져오기
    - 4

찿은 패턴으로 내용 추출

■ 〈ul class="ah\_l"〉태그 부분만 추출

- 2 웹 크롤링하기
  - 포털 사이트 급상승 검색어 가져오기
    - 5 필요한 정보만 가공하여 마무리
      - 반복문과 HTML 내 텍스트만 가져오는 get\_text 함수를 활용해 검색어만 추출

```
for i in rank.find_all("li"):
    print(i.get_text("위:", strip=True))

1위: 마인
2위: 홍길동
3위: 동국여지승람
```

2 웹 크롤링하기

전체 실습 코드

○ 포털 사이트 급상승 검색어 가져오기

# import urllib.request import bs4 web\_page = urllib.request.urlopen("https://www.naver.com")

result = bs4.BeautifulSoup(web\_page, "html.parser")
rank = result.find("ul".class\_="ah\_l")

for i in rank.find\_all("li"):

print(i.get\_text("위:", strip=True))

## Run! 프로그래밍

#### Mission

#### BeautifulSoup 모듈 활용

```
import urllib.request import bs4
```

```
web_page = urllib.request.urlopen
("https://movie.naver.com/movie/running/current.nhn")
result = bs4.BeautifulSoup(web_page, "html.parser")
rank = result.find(class_="lst_detail_t1")
movie_list = rank.find_all('li')
for n in range(0, len(movie_list)):
   title = movie_list[n].find(class_="tit").find("a").text
   print(n+1,"위:", title)
```



실행이 안될 경우 urllib 모듈 후 from urllib.request import Request 라인 입력