



# 부동산 금융에 데이터 과학 적용하기

4주차. 데이터 분석 프로그래밍 언어 학습하기, 웹 크롤링



1. open API 맛보기, 미국 Vanguard Real Estate ETF(VNQ) 주식 데이터 가져오기
2. 파이썬/판다스 이론, 실습
3. 국내 주식 데이터 수집하기 (웹 크롤링)
  - 3-1. 네이버 증권 - 국내증시 [최근 영업일 가져오기]
  - 3-2. 한국거래소(KRX 정보데이터시스템) 업종분류 현황 및 개별지표 크롤링
  - 3-3. 개별종목 지표 크롤링, 데이터 전처리하기
  - 3-4. FnGuide Index - 산업 섹터 정보 크롤링
  - 3-5. 네이버 증권 - 개별종목 수정주가 크롤링
  - 3-6. FnGuide - CompanyGuide - 재무제표 크롤링
  - 3-7. 가치지표(PER, PBR, PCR, PSR, DY) 계산

## DATABASE 구축하기

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

stock\_db

Tables

kor\_ticker

Views

Stored Procedures

Functions

Query 1 x

```
1 • use stock_db;
2
3 • select * from kor_ticker;
```

Limit to 1000 rows

Result Grid

종목 코드	종목명	시장 구분	종가	시가총액	기준일	EPS	선형 EPS	BPS	주당배당금	종목구분
005257	녹십자홀딩스2우	KOSPI	31750	26860200000	2023-05-24	NULL	NULL	NULL	300	우선주
005290	동진세미켐	KOSDAQ	36800	1892050000000	2023-05-24	3102	2924	13553	130	보통주
005300	롯데칠성	KOSPI	142900	1325950000000	2023-05-24	12754	16405	150133	3300	보통주
005305	롯데칠성우	KOSPI	68200	52876100000	2023-05-24	NULL	NULL	NULL	3305	우선주
005320	국풍	KOSPI	1102	63223400000	2023-05-24	25	NULL	1997	0	보통주
005360	모나미	KOSPI	3365	63589400000	2023-05-24	166	NULL	5502	70	보통주
005380	현대차	KOSPI	207500	43892800000000	2023-05-24	28521	38574	407147	7000	보통주
005385	현대차우	KOSPI	109600	2642800000000	2023-05-24	NULL	NULL	NULL	7050	우선주
005387	현대차2우B	KOSPI	111300	4020220000000	2023-05-24	NULL	NULL	NULL	7100	우선주
005389	현대차3우B	KOSPI	108500	2608830000000	2023-05-24	NULL	NULL	NULL	7050	우선주
005390	신성통상	KOSPI	2105	3025060000000	2023-05-24	533	NULL	2389	0	보통주
005420	코스모화학	KOSPI	53000	1855440000000	2023-05-24	334	NULL	6118	0	보통주
005430	한국공항	KOSPI	38800	1228550000000	2023-05-24	1888	NULL	110940	0	보통주
005440	현대지에프홀...	KOSPI	3785	2415720000000	2023-05-24	640	289	20860	210	보통주
005490	POSCO홀딩스	KOSPI	371000	313759000000000	2023-05-24	41456	40986	692323	12000	보통주
005500	삼진제약	KOSPI	23200	3224800000000	2023-05-24	1728	NULL	20811	800	보통주
005610	SPC삼립	KOSPI	78700	6791030000000	2023-05-24	6568	7985	48007	1700	보통주
005670	루드웰	KOSDAQ	5690	569000000000	2023-05-24	452	NULL	9493	90	보통주
005680	상영전자	KOSPI	9880	1976000000000	2023-05-24	909	891	26088	300	보통주
005690	파미셀	KOSPI	9090	5455540000000	2023-05-24	182	328	1328	0	보통주
005710	대원산업	KOSDAQ	6590	1320480000000	2023-05-24	1879	NULL	20844	175	보통주
005720	덕선	KOSPI	4320	2313100000000	2023-05-24	317	NULL	22295	110	보통주
005725	낙선우	KOSPI	3075	110718000000	2023-05-24	NULL	NULL	NULL	115	우선주
005740	크라온해태홀...	KOSPI	6560	973986000000	2023-05-24	559	NULL	20510	100	보통주
005745	크라온해태홀...	KOSPI	10590	63616900000	2023-05-24	NULL	NULL	NULL	105	우선주
005750	대원B&Co	KOSPI	3395	566023000000	2023-05-24	NULL	NULL	9514	110	보통주
005800	신영와코루	KOSPI	10880	979200000000	2023-05-24	2611	NULL	40302	150	보통주
005810	웅산홀딩스	KOSPI	36150	3614470000000	2023-05-24	7614	8001	99583	1300	보통주
005820	원티	KOSPI	22350	513596000000	2023-05-24	3500	NULL	55263	350	보통주
005830	DB손해보험	KOSPI	81300	5756040000000	2023-05-24	16373	23173	89508	4600	보통주
005850	에스엘	KOSPI	37150	1725560000000	2023-05-24	3360	5629	35492	600	보통주
005860	한일사료	KOSDAQ	6480	2553360000000	2023-05-24	NULL	NULL	1371	0	보통주
005870	후니드	KOSPI	6610	933069000000	2023-05-24	707	NULL	11026	0	보통주
005880	대한해운	KOSPI	2000	6383550000000	2023-05-24	507	405	4912	0	보통주

kor\_ticker 1 x

Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message
5	07:49:05	use stock_db	0 row(s) affected
6	07:49:20	select * from kor_ticker LIMIT 0, 1000	1000 row(s) returned

Automatic context help is disabled. Use the Context Help icon to manually get help for the current command or to toggle automatic help.



## 팩터 기반 상장 리츠 지수 개발과 리츠 재간접 펀드 운용

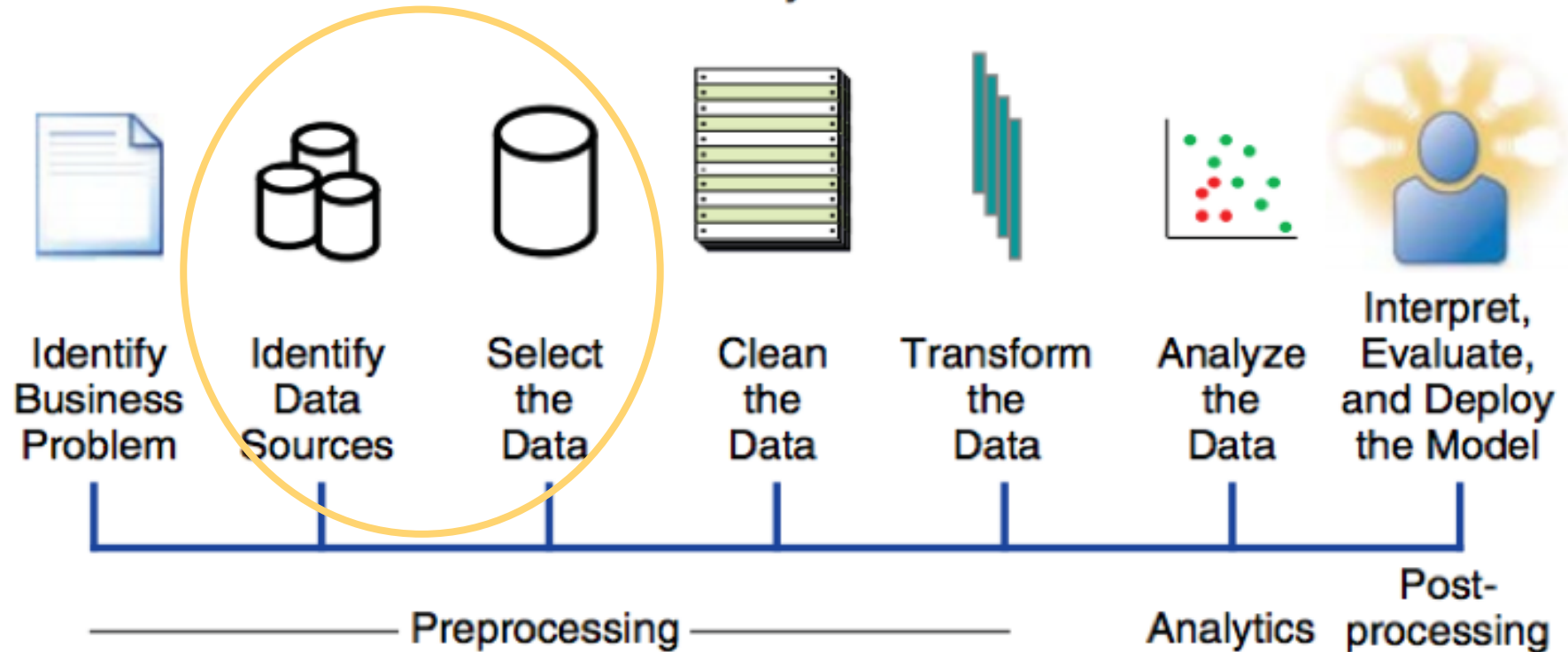
SPI는 글로벌 지수사업자인 MSCI, FTSE와 같이 팩터 기반의 지수(index)를 설계할 수 있는 시스템을 갖추었습니다. MSCI와 FTSE는 [Value, Size, Momentum, High Dividend, Quality, Volatility] 여섯 가지 팩터를 기준으로 지수를 설계합니다. SPI는 여섯 가지 팩터를 기초로 부동산에 특화된 시계열 데이터를 적재하고 있습니다. 예를 들어, Value(가치주) 팩터 평가를 위해 NAV 등의 데이터를 시계열로 데이터베이스에 자동 적재하고 있습니다. Quality(우량주) 팩터 평가를 위해 FFO Growth 등의 데이터를 시계열로 적재하고 있습니다. Dividend(배당주) 팩터 평가를 위해 임대료에 기초한 배당금 데이터를 시계열로 적재하고 있습니다. 이처럼 SPI는 상업용 부동산에 특화된 데이터를 시계열로 적재할 수 있는 서버방법론 및 데이터베이스 아키텍처를 구축하였습니다. 이와 관련한 특허를 출원하였고 싱가포르증권거래소에게 그것의 우수성을 인정받아 파트너십 계약을 체결하였습니다.

자산운용사가 상장 리츠 자금을 운용하는 방식에는 크게 두가지가 있습니다. 지수를 추종하는 Passive Investing 방식의 ETF 펀드와 담당 펀드 매니저의 정성적 리서치에 의존하는 Active Investing 방식이 있습니다. 미래에셋자산운용의 TIGER ETF, 한국투자신탁운용의 ACE ETF가 대표적인 ETF 브랜드입니다. 해외 주식에 투자하는 ETF의 경우 MSCI와 FTSE와 같은 글로벌 사업자가 개발한 지수를 추종하고 국내의 경우 FnGuide 지수 혹은 자체적으로 계산한 단순한 지수를 추종하는 방식으로 운용되고 있습니다. 이와 반대로 Active 펀드는 담당 펀드 매니저의 판단에 근거하여 자금을 운용하고 있습니다. 이지스자산운용과 마스틴투자운용이 Active 방식으로 자금을 운영하고 있습니다. 현재 **국내 상장 리츠 자금 운용 시장은 ETF 운용에 있어 핵심이 되는 지수를 개발할 수 있는 역량을 갖춘 지수개발사업자가 부재하여 시가총액가중지수라는 단순한 방식으로 Passive 펀드가** 운용되고 있고, Active 펀드는 담당 매니저의 정성적 판단에 의존하고 있어 문제가 있습니다.

현재 국내에서 가장 규모가 큰 리츠 ETF는 미래에셋자산운용에서 운용하는 TIGER 리츠부동산인프라(AUM 2,700억원 규모)입니다. 해당 ETF가 추종하는 지수는 FnGuide에서 개발한 리츠부동산인프라지수입니다. FnGuide의 해당 지수는 시가총액가중지수를 추종하는 매우 단순한 방법론에 기초하고 있습니다. 팩터 중 SIZE(시가총액)만을 기준으로 설계된 지수입니다. 단순히 시가총액이 큰 순서대로 투자하도록 설계된 지수입니다. SPI는 FTSE와 MSCI와 같이 팩터 기반의 지수를 개발할 수 있는 시스템을 갖추었고 통계적 분석방법론에 기초한 한국 리츠 지수를 개발 중에 있습니다. SPI 지수가 개발되면 국내 자산운용사 및 증권사와 ETF, ETN 설계가 가능합니다. 이를 통한 지수 제공 수수료 수익이 발생합니다. 해당 지수 설계 방법론을 기초로 싱가포르 등 해외 기관투자자에게 국내 리츠 투자 자문을 제공하며 자문 수익을 발생할 수 있습니다. 2024년 하반기에 사업화 될 예정입니다. 국내 리츠 지수 개발은 시작에 불과합니다. SPI는 MSCI와 FTSE 보다 발전된 글로벌 상장 리츠 지수를 설계할 수 있는 잠재력을 갖고 있습니다. 현재 전 세계에서 가장 규모가 큰 리츠 ETF는 Vanguard Real Estate ETF(약 80조원 규모)입니다. 해당 지수는 MSCI가 개발한 MSCI US IMI Real Estate 25/50 Index를 추종하고 있습니다. 해당 리츠의 지수 방법론은 앞서 언급한 여섯 가지 팩터에 기초하고 있는데 상업용 부동산에 특화된 데이터를 적재하고 분석하지 않고 일반 기업 증권 분석 방식에 기초하고 있습니다. 상업용 부동산에 최적화된 팩터 분석이 아닙니다. 반면에 SPI는 상업용 부동산에 특화된 시계열 데이터를 적재하고 분석하고 있습니다. SPI가 리츠 분야에서 전 세계에서 가장 앞선 데이터 기반 자산운용사가 될 개연성이 여기에 있습니다.

미국, 싱가포르, 일본, 한국 상장 리츠 규모는 약 2,500조원 규모입니다. SPI는 향후 자산운용사를 설립하여 재간접 펀드를 운용할 것입니다. 10년 안에 해당 국가 리츠 시장의 5%인 126조원(시장 성장률을 고려하지 않고 단순 계산) 규모의 자금을 운용 하고자 합니다. 연간 0.5% 운용 수수료 적용 시 연간 6천억원 이상의 수수료를 기대할 수 있습니다.

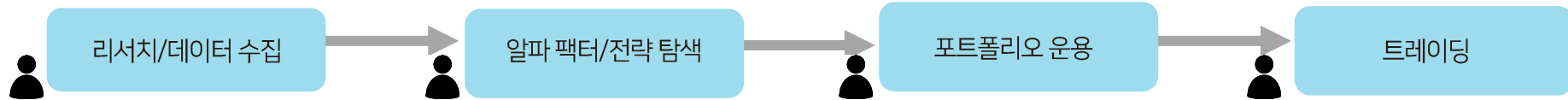
## Overview of the Analytics Process Model



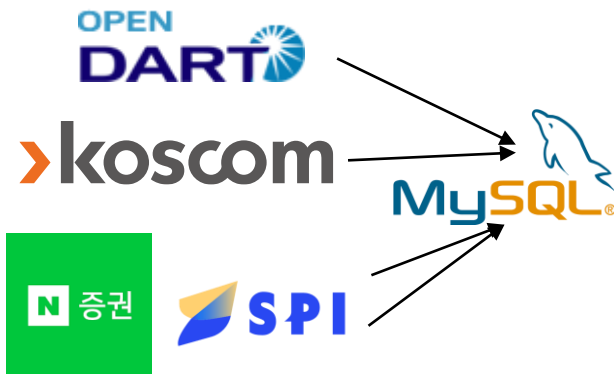
<https://blogs.sas.com/content/sgf/2019/05/14/big-data-in-business-analytics-talking-about-the-analytics-process-model/>

- 기존 자산운용[데이터 수집 - 알파 팩터/전략 설정/포트폴리오 운용- 트레이딩]과정을 데이터베이스 및 IT인프라를 기초로 고도화 및 효율화

기존 자산운용사: 애널리스트 / 펀드매니저



퀀트 포트폴리오 운용: 데이터베이스 / 퀀트 계량분석 / IT인프라



BeautifulSoup

Beautiful Soup is a Python library for pulling data out of HTML and XML files

FACTOR / BETA

자산가격결정모형

가치주 투자

퀄리티 투자

최대샤프지수



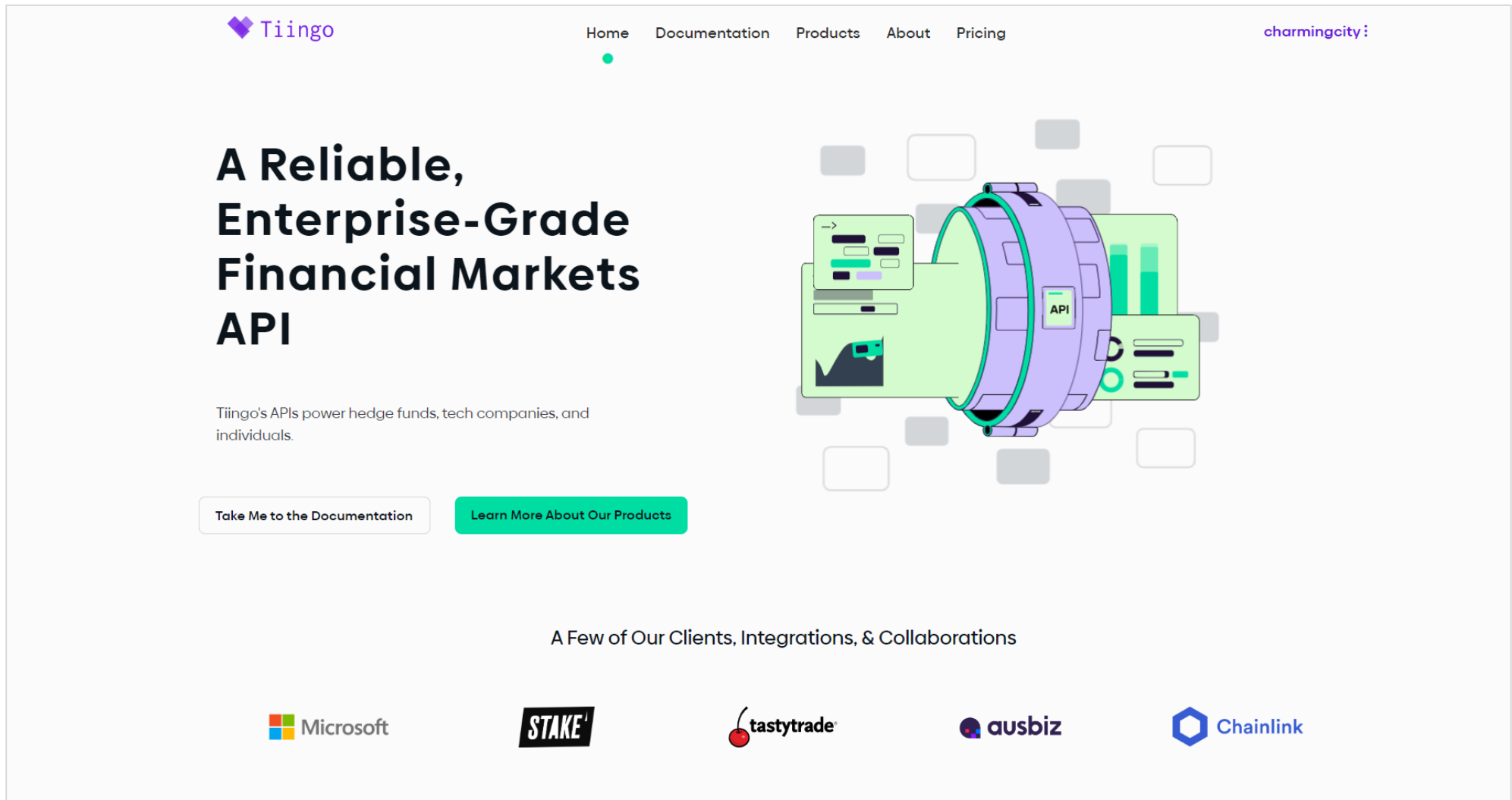
Portfolio Optimization and Quantitative Strategic Asset Allocation in Python



Flexible Backtesting for Python

한국투자증권 API





The screenshot shows the Tiingo website homepage. At the top, there is a navigation bar with links for Home, Documentation, Products, About, and Pricing. The Tiingo logo is on the left, and 'charmingcity:' is on the right. The main heading reads 'A Reliable, Enterprise-Grade Financial Markets API'. Below this, a subtext states 'Tiingo's APIs power hedge funds, tech companies, and individuals.' There are two buttons: 'Take Me to the Documentation' and 'Learn More About Our Products'. To the right of the text is an illustration of a purple vault door with a green 'API' label, surrounded by floating green boxes containing charts and code snippets. Below the main content, a section titled 'A Few of Our Clients, Integrations, & Collaborations' displays logos for Microsoft, STAKE, tastytrade, ausbiz, and Chainlink.

Tiingo

Home Documentation Products About Pricing

charmingcity:

## A Reliable, Enterprise-Grade Financial Markets API

Tiingo's APIs power hedge funds, tech companies, and individuals.

Take Me to the Documentation

Learn More About Our Products

A Few of Our Clients, Integrations, & Collaborations

Microsoft STAKE tastytrade ausbiz Chainlink

<https://api.tiingo.com/>



Tiingo API를 통해 뉴욕증권거래소에 상장된 '뱅크드 리츠 ETF' 주가 정보 가져오기

The screenshot displays the Spyder Python IDE environment. The main editor shows a Python script named `Tiingo.py` that performs the following steps:

- Imports `keyring`, `TiingoClient` from `tiingo`, `pandas` as `pd`, and `keyring`.
- Sets the password for 'tiingo' using `keyring.set_password`.
- Retrieves the API key using `keyring.get_password`.
- Creates a `TiingoClient` instance with the API key.
- Lists stock tickers using `client.list_stock_tickers()`.
- Fetches metadata for the ticker 'AAPL' using `client.get_ticker_metadata("AAPL")`.
- Prints the `ticker_metadata` dictionary.

A variable explorer window titled `ticker_metadata - Dictionary (6 elements)` shows the following data:

Key	Type	Size	Value
description	str	941	Apple Inc. (Apple) designs, manufactures and markets mobile communicat ...
endDate	str	10	2023-04-21
exchangeCode	str	6	NASDAQ
name	str	9	Apple Inc
startDate	str	10	1980-12-12
ticker	str	4	AAPL

The console output shows an `AttributeError` because the script uses `get_ticker_metadatada` (note the typo) instead of `get_ticker_metadata`:

```
AttributeError: 'TiingoClient' object has no attribute 'get_ticker_metadatada'
```

<https://api.tiingo.com/>

<https://api.tiingo.com/documentation/general/overview>

<https://github.com/hydrosquall/tiingo-python>

## 자료형

- 기본 자료형: 숫자형(number) / 문자형 (string)
- 구조 자료형: 배열형 - 리스트(list) / 튜플(tuple)  
집합(Set)  
사전 (Dictionary)  
부울 (Boolean)

자료형은 프로그래밍의 기본이자 핵심 단위.  
프로그래밍을 할 때 쓰이는 숫자, 문자열 등 자료  
형태로 사용되는 모든 것을 의미.

## 제어문

- 조건문(if문)
- 반복문(for문) : 반복 횟수가 정해짐
- 반복문(while문): 무한 루프, 특정 조건을 만족할 때까지

제어문: 자료형을 제어한다.

## 함수, 클래스

함수: 프로그램의 반복적인 입력(Input)과 출력(Output)을 실행

클래스: 인스턴스를 생성하는 틀. 속성(attribute)과 속성의 동작을 수행하는 메소드(method)로 구성.

메소드는 클래스 내에 정의 된 함수이며, 속성은 함수를 통해 할당됨

클래스는 객체를 만드는 틀. 클래스로부터 객체를 생성하는 것을 인스턴스화(instantiation)라고 함.

객체는 임의 변수를 의미하며, 인스턴스는 특정한 클래스를 통해 만든 객체.

생성된 인스턴스가 가지고 있는 속성과 메소드는 닷(.)을 사용해서 접근할 수 있음

클래스 이름은 대문자로 시작

## 패키지, 모듈

패키지: 라이브러리, 특정 기능과 관련된 여러 모듈을 한 그룹으로 묶은 것으로 패키지 안에 패키지가 있을 수도 있다. Import from 패키지

모듈: 클래스와 함수의 집합인 파이썬 파일. 변수, 함수, 클래스 등을 포함

## 클래스

클래스: 인스턴스를 생성하는 틀. 속성(attribute)과 속성의 동작을 수행하는 클래스 내 함수인 메소드(method)로 구성. 속성은 크게 클래스 속성과 인스턴스 속성으로 나뉜다. 속성은 클래스 내부에 포함돼 있는 메서드나 변수를 의미한다. 메소드는 클래스 내에 정의된 함수이며, 속성은 함수를 통해 할당됨

```
In [1]: #클래스(class)

class Monster(): #class는 Monster 객체들을 만들어 내는 붕어빵 기계
    hp = 100
    alive = True

    # damage 메서드, 클래스 내부에 있는 함수를 메서드라고 함
    # 파이썬 메서드의 첫 번째 매개변수 이름은 관행적으로 self를 사용한다. 객체를 호출한 객체 자신이 전달되기 때문에 self라는 이름을 사용.
    def damage(self, attack): #메서드의 매개변수(self, attack)
        self.hp = self.hp - attack #메서드의 수행문
        if self.hp < 0:
            self.alive = False

    #status_check 메서드, 클래스 내부에 있는 함수를 메서드를 하고 함
    def status_check(self):
        if self.alive:
            print('살아있다')
        else:
            print('죽었다')
```

```
In [74]: #m1는 클래스의 인스턴스
#m1 객체는 Monster 클래스의 인스턴스이다. 인스턴스는 특정 객체(m1)가 어떤 클래스인지 관계 위주로 설명할 때 사용
#m1은 객체(object).
m1 = Monster()
m1.damage(120) #m1 객체를 통해 damage 메소드를 호출, def(self, attack) 메서드의 매개변수인 self에 m1 객체가 자동으로 전달됨,
m1.status_check()
```

죽었다

```
In [75]: #m2는 Mosnter 클래스의 인스턴스
```

```
m2 = Monster()
m2.damage(20)
m2.status_check()
```

살아있다

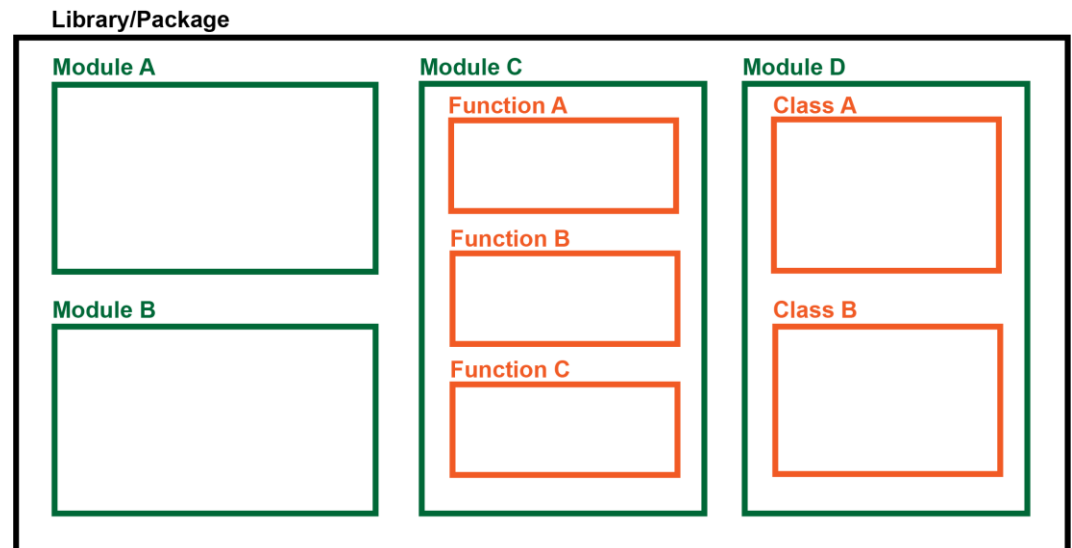
\*객체지향적 프로그래밍: 소프트웨어 개발에 필요한 모든 요소를 객체화하여 프로그래밍하는 기법

### 라이브러리 패키지 > 모듈의 집합 > 변수, 함수, 클래스를 모아 놓은 파일

파이썬 최고의 강력한 장점 중 하나는 오픈소스 커뮤니티가 계속해서 발전시켜 나가는 라이브러리 패키지들을 사용할 수 있다는 것이다.

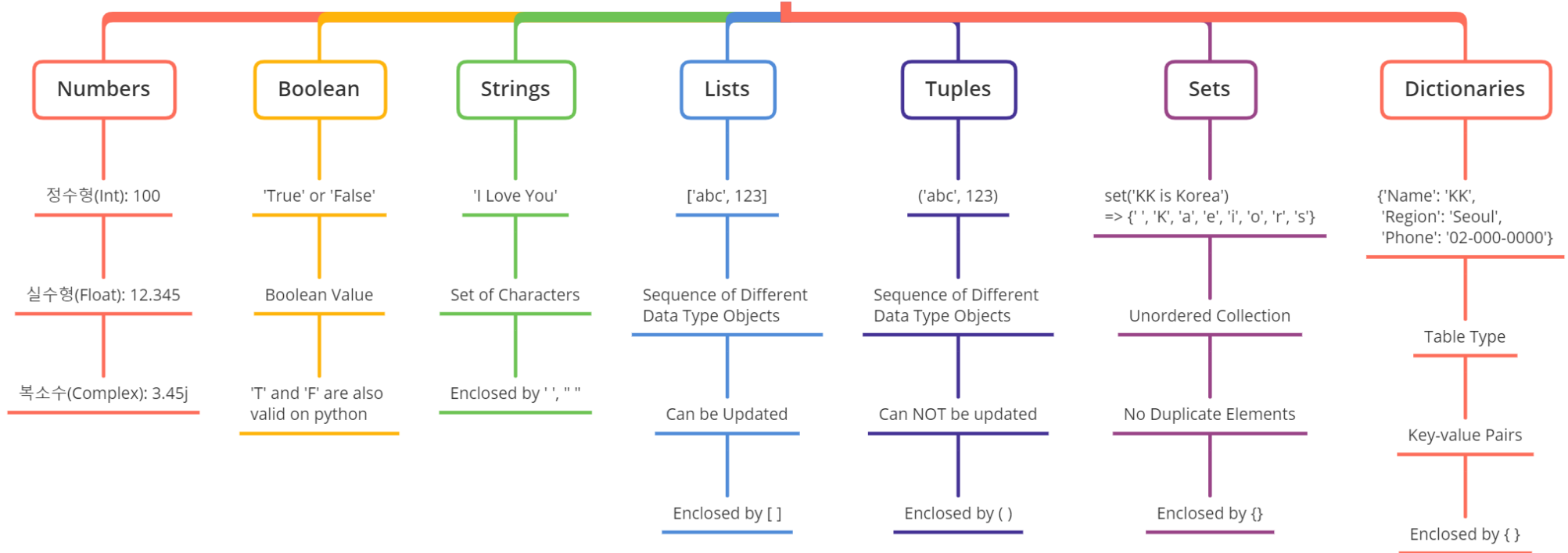
다른 컴퓨터 과학자가 모듈(module) 단위로 작성해 둔 코드를 가져다 쓸 수 있다. 모듈은 미리 만들어 놓은 파이썬(.py) 파일이다.

패키지(package)는 여러 모듈을 특정 디렉토리에 모아 둔다. 닷(.)을 사용하여 파이썬 모듈을 계층적으로 관리할 수 있게 한다. 예를 들어, `scipy.spatial`이 있으면 `scipy`는 패키지 이름이고 `spatial`은 패키지의 모듈이다. 모듈이란 변수, 함수, 클래스를 모아 놓은 파일이다. 다른 파이썬 프로그램에서 불러와 사용할 수 있게 만들어졌다. 본인이 만들어 놓거나 다른 사람이 만들어 놓은 모듈을 가져와 활용할 수 있다.



<https://hyunsitstory.tistory.com/entry/python-%ED%8C%A8%ED%82%A4%EC%A7%80-%EB%AA%A8%EB%93%88-%ED%95%A8%EC%88%98-%ED%81%B4%EB%9E%98%EC%8A%A4-%EA%B0%9C%EB%85%90-%EC%A0%95%EB%A6%AC>

## Python Data Types





## 자료형 객체

자료형 객체	의미	사용
숫자형 int	자연수 객체	자연수
숫자형 float	실수 객체	실수
문자 자료형 str	문자열 객체	글자
리스트형 list	가변 컨테이너	변화하는 레코드 집합
튜플형 tuple	불변 컨테이너	고정된 객체 집합, 레코드
집합형 set	가변 컨테이너	유일한 객체 집합
사전형 dict	가변 컨테이너	키와 값의 저장소
부울형 bool	비교, 논리연산	참 또는 거짓



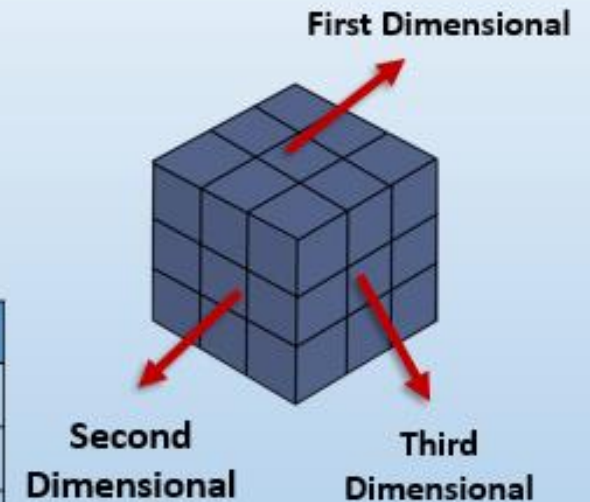
# NumPy Narray

10	15	13	8	25
----	----	----	---	----

**1D-Array**

	Column 0	Column 1	Column 2
Row 0	X[0][0]	X[0][1]	X[0][2]
Row 1	X[1][0]	X[1][1]	X[1][2]
Row 2	X[2][0]	X[2][1]	X[2][2]

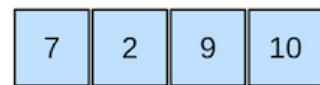
**2D-Array**



**3D-Array**

[www.educba.com](http://www.educba.com)

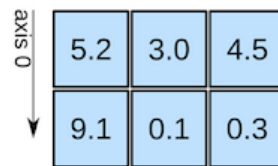
## 1D array



axis 0 →

shape: (4,)

## 2D array

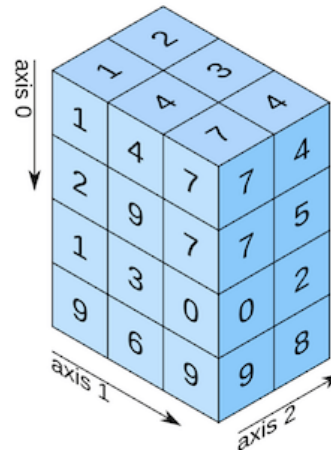


axis 0 ↓

axis 1 →

shape: (2, 3)

## 3D array



axis 0 ↓

axis 1 →

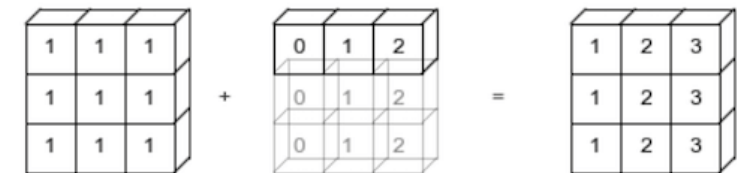
axis 2 →

shape: (4, 3, 2)

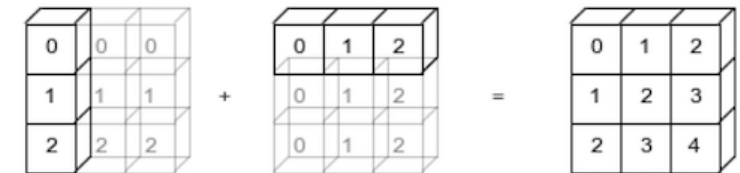
`np.arange(3)+5`



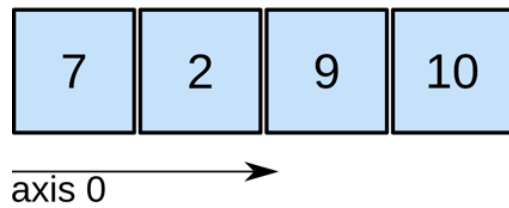
`np.ones((3, 3))+np.arange(3)`



`np.arange(3).reshape((3, 1))+np.arange(3)`

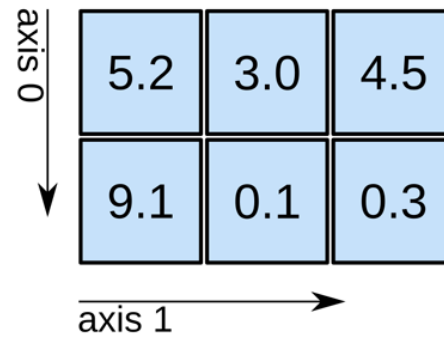


1D array



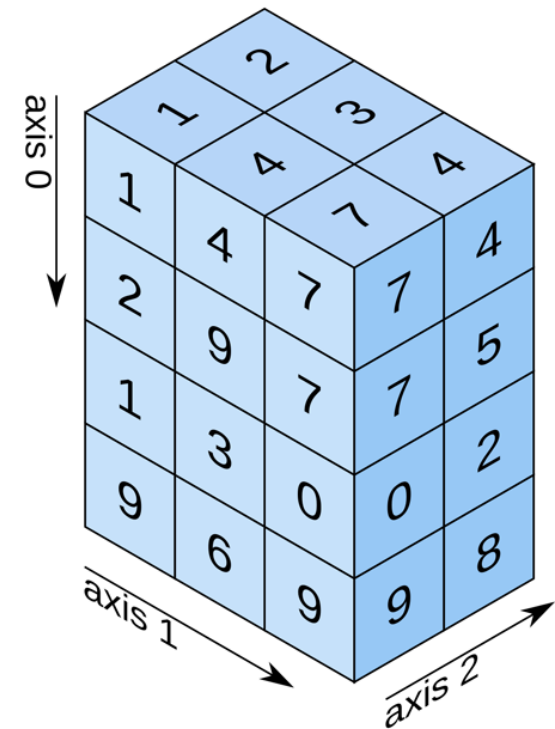
shape: (4,)

2D array



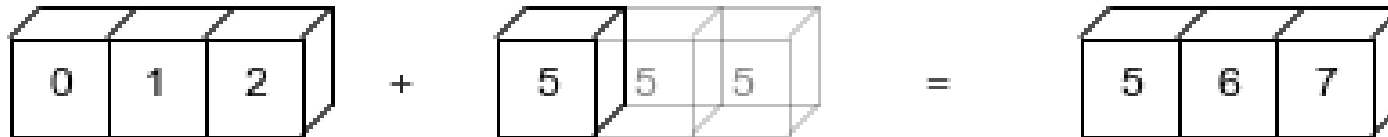
shape: (2, 3)

3D array

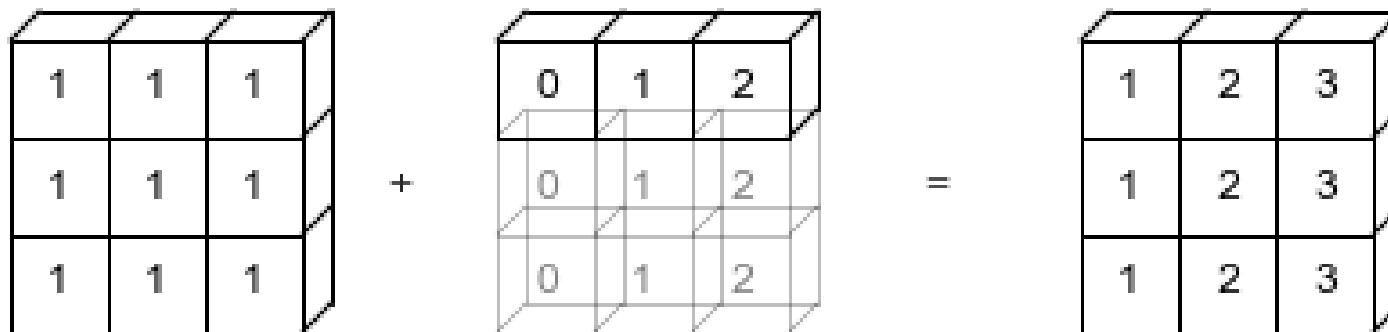


shape: (4, 3, 2)

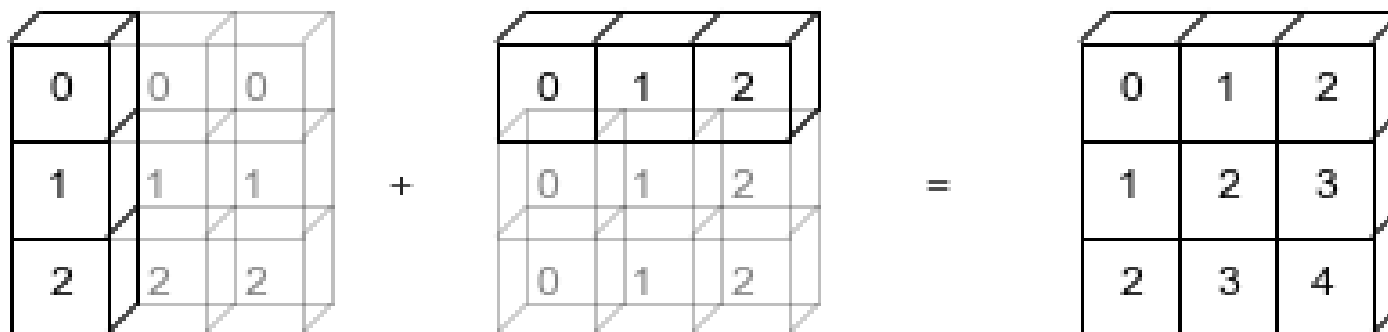
`np.arange(3) + 5`



`np.ones((3, 3)) + np.arange(3)`



`np.arange(3).reshape((3, 1)) + np.arange(3)`







판다스는 2008년 금융데이터 분석용으로 개발됨. 오픈소스 커뮤니티를 통해 계속해서 발전하고 있으며 통계, 데이터과학, 머신러닝 분야에서 가장 중요한 소프트웨어로 성장함

## 판다스 자료형

판다스는 데이터를 효율적으로 다루기 위해 시리즈(Series)와 데이터프레임(DataFrame)이라는 자료형을 사용한다. 데이터프레임은 엑셀에서 볼 수 있는 시트(Sheet)와 동일한 개념이며 시리즈는 시트의 열 1개를 의미한다. 파이썬으로 비유하여 설명하면 데이터프레임은 시리즈들이 각 요소가 되는 딕셔너리(Dictionary)와 유사하다.

판다스는 시리즈와 데이터프레임이라는 구조화된 데이터 형식을 제공한다. 서로 다른 종류의 데이터를 한곳에 담는 그릇(컨테이너)가 된다. 시리즈는 1차원 배열이고, 데이터프레임은 2차원 배열(행렬)이다.

판다스의 1차적인 목적은 서로 다른 유형의 데이터를 공통의 포맷으로 정리하는 것이다. 특히 행과 열로 이루어진 2차원 구조의 데이터프레임은 데이터 분석 실무에서 사용된다.

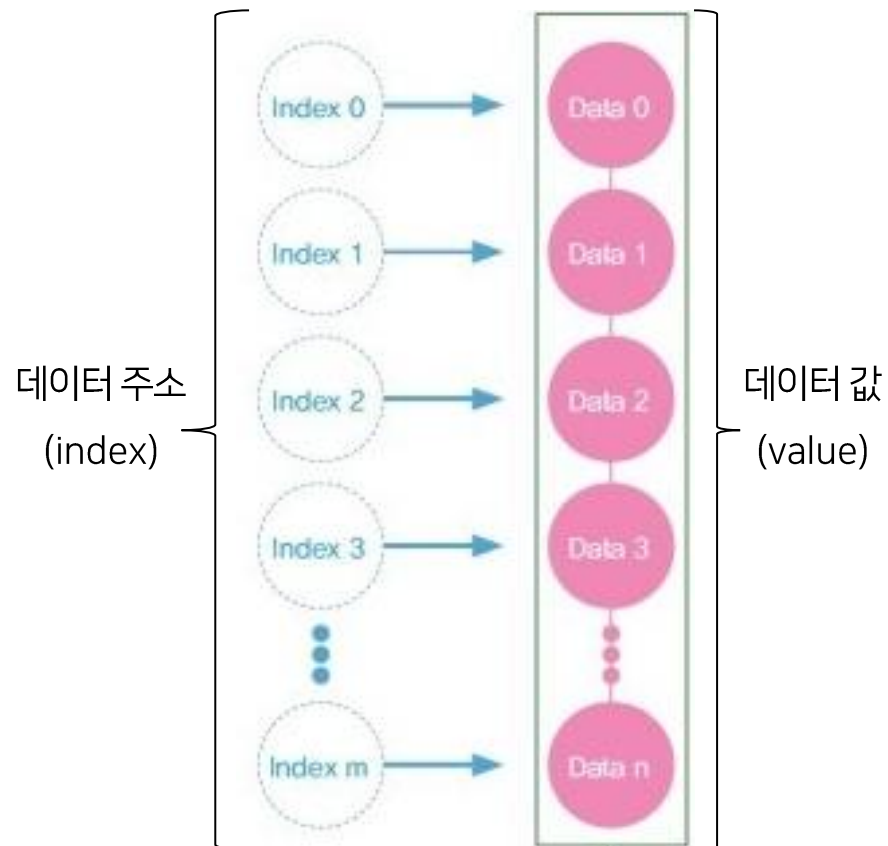
시리즈  
(Series)

데이터프레임  
(DataFrame)

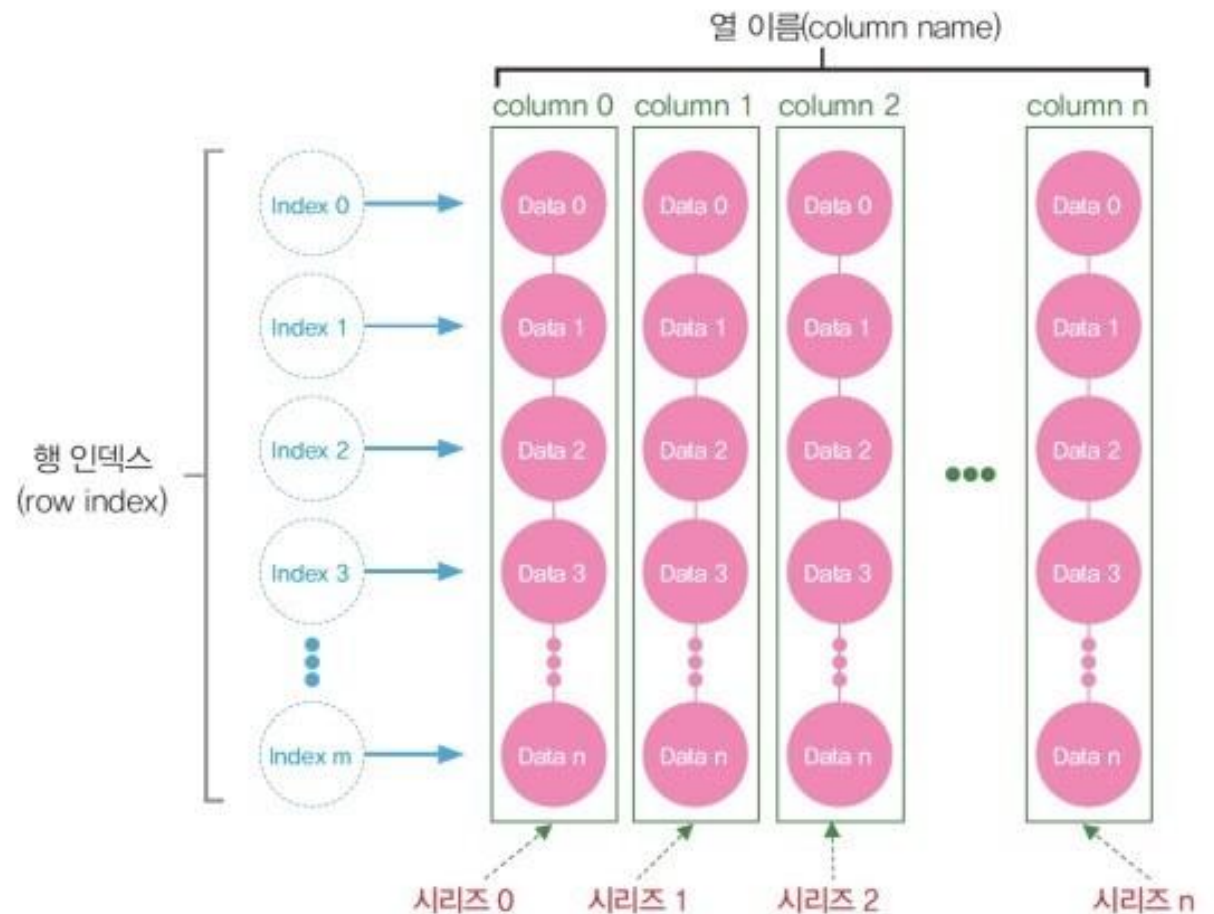
정수형 위치 인덱스(integer position), 인덱스 이름(index name), 인덱스 라벨(index label)

인덱스 이름 (optional)	$i_0$	$i_1$	$i_2$	$i_3$	$i_4$	...	$i_n$
데이터 값 (원소)	$v_0$	$v_1$	$v_2$	$v_3$	$v_4$	...	$v_n$
위치 인덱스	0	1	2	3	4	...	5

시리즈  
(Series)  
1차원 배열



데이터프레임  
(DataFrame)  
2차원 배열



## Python dictionary VS Pandas DataFrame

```
In [1]: import pandas as pd

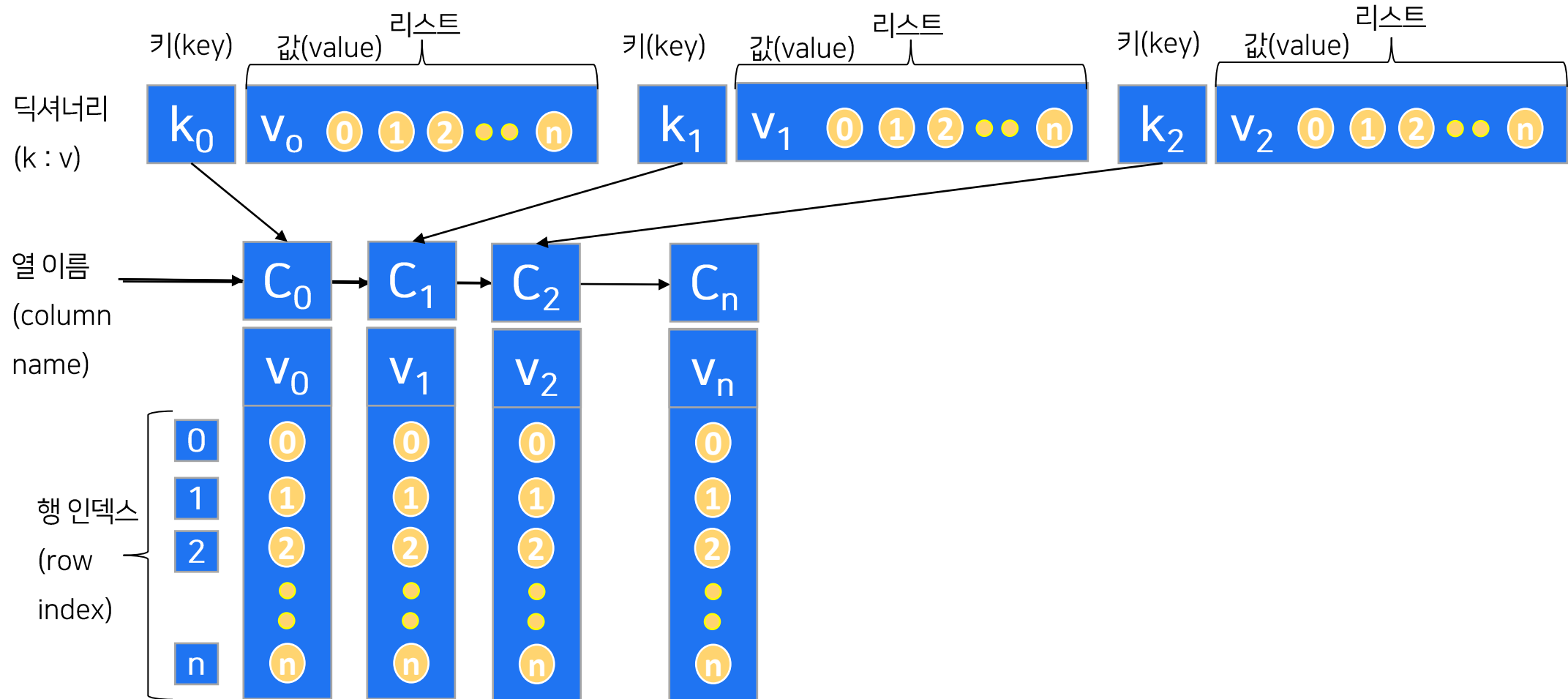
data = {'Key1': [1, 2, 3],
        'Key2': ['Hank', 'Steve', 'Lisa'],
        'Key3': [1.2, 3.1, 3.1]}

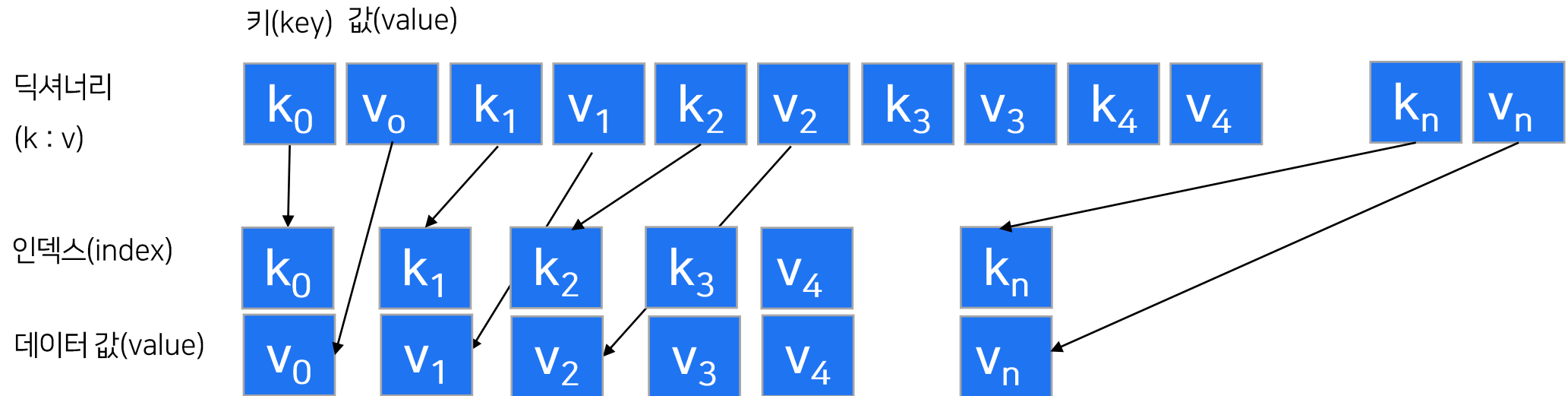
df = pd.DataFrame(data)
df.head()
```

Out [1]:

	Key1	Key2	Key3
0	1	Hank	1.2
1	2	Steve	3.1
2	3	Lisa	3.1







파이썬 딕셔너리 자료형: {Key1 : Value1, Key2: Value 2, Key3: Value3, ...}

딕셔너리 -> 시리즈 변환 : pandas.Series(딕셔너리)

딕셔너리 -> 데이터프레임 변환 : pandas.DataFrame(딕셔너리 객체)

## Series

	apples
0	3
1	2
2	0
3	1

+

## Series

	oranges
0	0
1	3
2	7
3	2

=

## DataFrame

	apples	oranges
0	3	0
1	2	3
2	0	7
3	1	2

Series 1

Mango	
0	4
1	5
2	6
3	3
4	1

+

Series 2

Apple	
0	5
1	4
2	3
3	0
4	2

+

Series 3

Banana	
0	2
1	3
2	5
3	2
4	7

=

DataFrame

Mango		Apple		Banana	
0	4	5		2	
1	5	4		3	
2	6	3		5	
3	3	0		2	
4	1	2		7	

판다스 라이브러리는 여러 종류의 클래스(class)와 다양한 내장 함수(built-in function)으로 구성된다. 시리즈와 데이터프레임은 구조를 표현하는 대표적인 클래스 객체이다. 시리즈와 데이터프레임 클래스의 속성과 메소드를 잘 이해하면 판다스를 자유자재로 다루는 데 어려움이 없다. 내장 함수로는 `Series()`, `DataFrame()`, `read_csv()`, `read_excel()` 등이 있다.



## 웹 크롤링하기



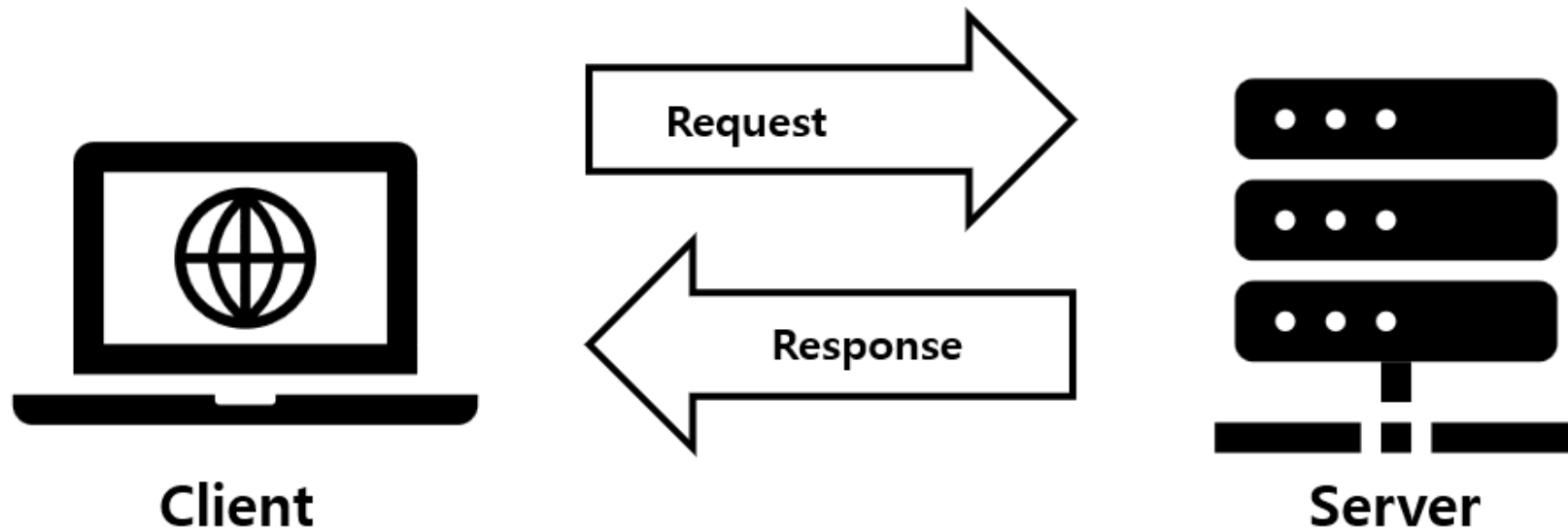
# BeautifulSoup

BeautifulSoup, 해당 페이지의 HTML에서 원하는 데이터를 가져와서 파싱하는 라이브러리이다

## 웹 크롤링하기

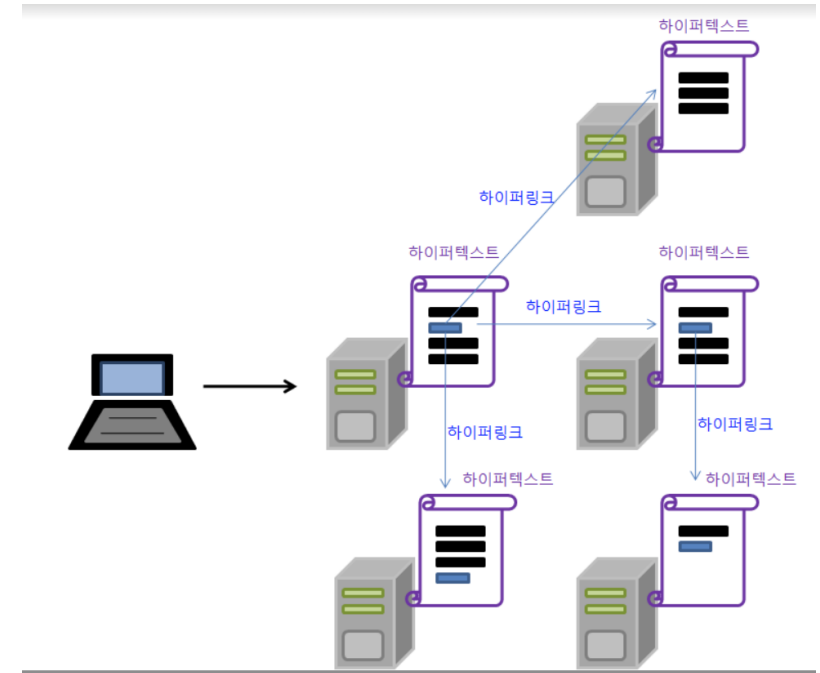
Get 방식: 인터넷 주소를 기준으로 이에 해당하는 데이터나 파일을 요청하는 것  
(웹 브라우저에서 확인 가능)

Post 방식: 사용자가 필요한 값을 추가해서 요청하는 방법,  
GET 방식과 달리 클라이언트가 요청하는 쿼리를 body에 넣어서 전송하므로 요청 내역을 직접 볼 수 없다.  
(개발자 도구 화면에서 확인 가능)



## 일반적인 웹 크롤링 방식

먼저 requests 패키지의 get() 또는 post() 함수를 이용해 데이터를 요청한 후 HTML 정보를 가져오며, bs4 패키지의 함수들을 이용해 원하는 데이터를 찾는 과정으로 이루어진다.



끝없는 해체.재구성이 '이어령표 이야기'를 만든다⑥  
이어령의 '하이퍼텍스트'

<https://www.joongang.co.kr/article/6704227#home>

두 명의 지성인을 기리며

<http://www.latimes.kr/news/articleView.html?idxno=39650>

하이퍼텍스트를 기술하는 언어가 HTML(HyperText Markup Language)이다. HTML에서는 문장 표시 방법이나 하이퍼링크를 태그tag로 표현한다. 이런 언어를 일반적으로 마크업 언어라고 부른다.

HTML에서는 <태크의 종류> 태그의 의미를 적용할 대상이 되는 문장 </태크의 종류>라는 형태로 기술함으로써 문장에 의미를 줄 수 있다. 구체적으로는 해당 문장이 제목임을 나타내거나, 하이퍼링크임을 나타내거나, 테이블임 등을 나타낸다. 예를 들어, <title>웹 기술의 모든 것</title>은 해당 문자의 목이 '웹 기술의 모든 것'이라는 의미이다.

HTML로 기술된 문서를 콘텐츠contents라고 부른다.

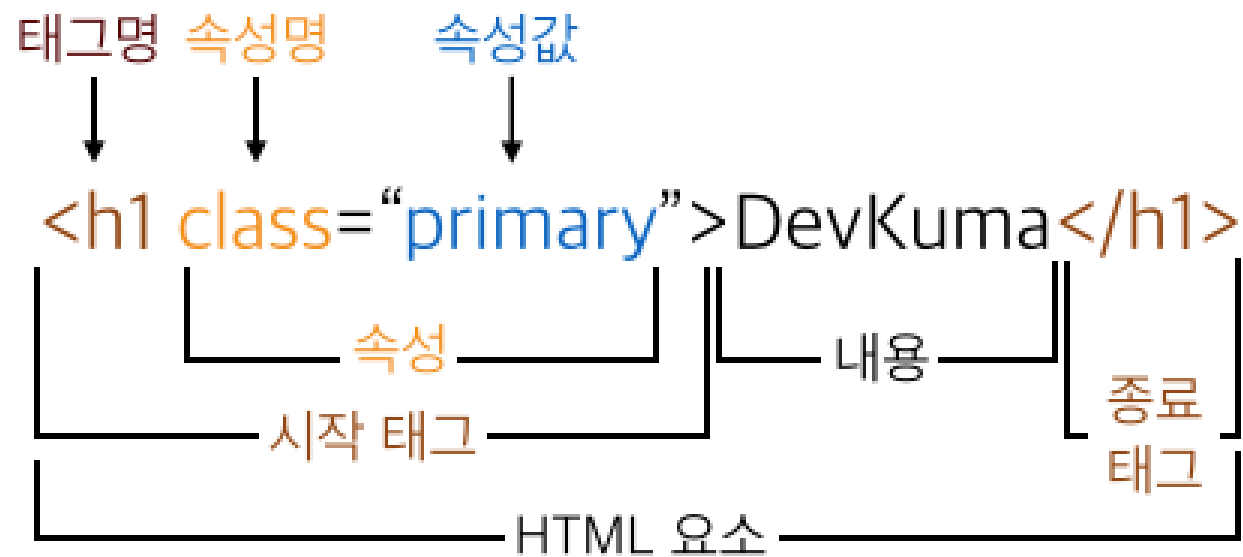
## 표시 프로그램 웹 브라우저

하이퍼텍스트는 문장에 태그로 의미를 붙인 것이며, 사람이 그 상태를 읽기에는 적절하지 않다. 그래서 하이퍼텍스트를 해석해서 사람이 읽기 쉽도록 바꿔주는 것이 웹브라우저라는 프로그램이다. 일반적으로 사용하는 웹브라우저에는 크롬chrome, 에지edge 등이 있다.

웹브라우저의 종류에 따라 표시 방법이 다소 다르기는 하지만 HTML 규칙은 세계 공통이므로, 기본적으로 어떤 웹 브라우저에서도 동일하게 콘텐츠를 볼 수 있다.

출처: 웹의 기초 / 코바야시 교헤이, 사카모토 아키라 지음 / 위키북스

## HTML 구성 요소 분석



한눈에 보는 HTML 요소(Elements & Attributes) 총정리

<https://heropy.blog/2019/05/26/html-elements/>

# What's the Difference?



## HTML

Hypertext Markup Language

### *Create the structure*

- Controls the layout of the content
- Provides structure for the web page design
- The fundamental building block of any web page



## CSS

Cascading Style Sheet

### *Stylize the website*

- Applies style to the web page elements
- Targets various screen sizes to make web pages responsive
- Primarily handles the "look and feel" of a web page



## Javascript

### *Increase interactivity*

- Adds interactivity to a web page
- Handles complex functions and features
- Programmatic code which enhances functionality

<https://seoulpi.io/article/32672>

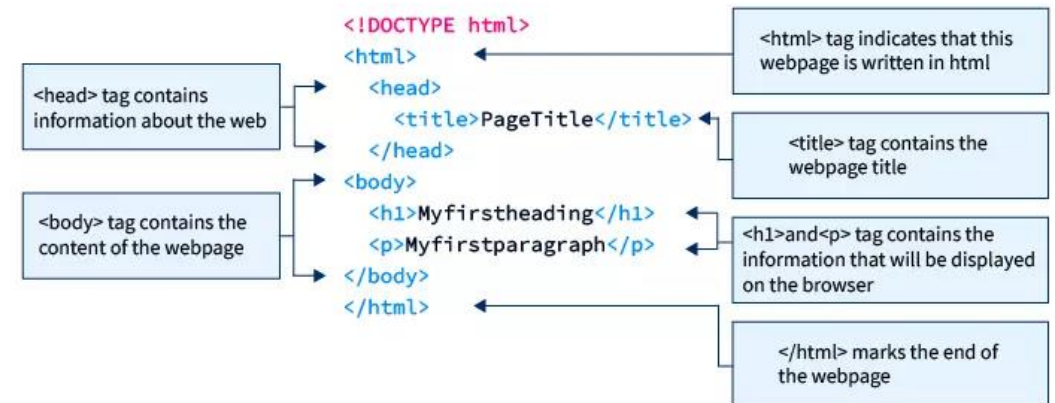
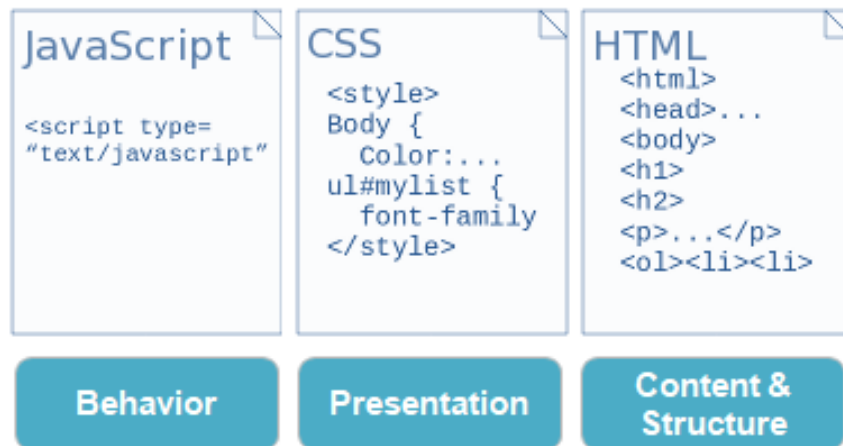
[https://charmingcity.github.io/spi\\_chatgpt/](https://charmingcity.github.io/spi_chatgpt/)

[https://www.w3schools.com/html/html\\_editor.asp](https://www.w3schools.com/html/html_editor.asp)

## HTML, CSS and JavaScript

These 3 web programming languages in conjunction take care of all the information webpage contains (from text to adding special effects).

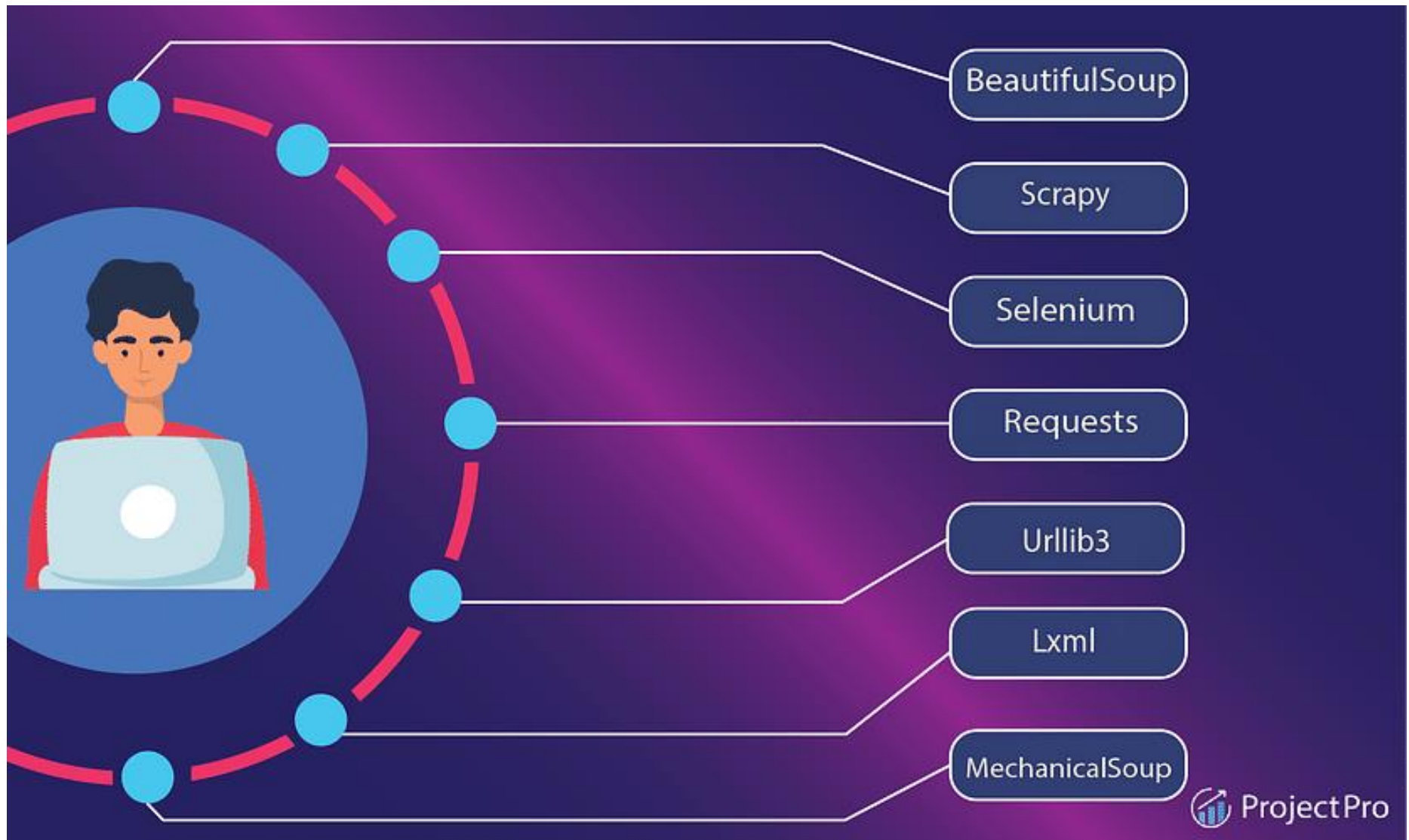
- HTML determines the content and structure of a page (header, paragraph, footer etc.)
- CSS controls how webpage would look like (color, font type, border etc.)
- JavaScript decides advanced behaviors such as pop-up, animation etc.



<https://sunjackson.github.io/2018/12/23/af11fc584cb237cd89ea175e9a713bbd/>

SCALER  
Topics  
<https://www.scaler.com/topics/structure-of-html-document/>





<https://www.projectpro.io/article/python-libraries-for-web-scraping/625>