



# Portfolio

## Park June Woo

## 프로젝트

C++ Gamemanager

---

Pygame을 이용한 게임개발

---

Django를 이용한 웹 사이트

---

ML을 이용한 가상화폐 시장 예측

---

# Content

1	GameManager .....	1
2	Finding the rabbit .....	6
3	Django web site .....	12
4	Machine Learning .....	18

# GameManager 개발 동기

- 평소에 게임에 관심이 많아 스스로 게임을 직접 개발해보고 싶었고, 같은 목적을 가진 친구들과 함께 오락실, 아날로그 느낌의 프로그램을 개발 해보기로 함.
- 객체지향 프로그래밍을 주제로 진행되는 과목에서 진행된 강의의 프로젝트로, 객체지향 패러다임을 받아들여 높은 수준은 아니지만 구도 만큼은 최대한 객체지향 방식으로 개발 해보기로 하였음.
- 각자가 해보고 싶은 요소들(기능들)을 최대한 구현해보기로 하였고, 각자가 만든 소스들 끼리 호환이 가능하도록 만들어 보고 싶었음.

# GameManager

01

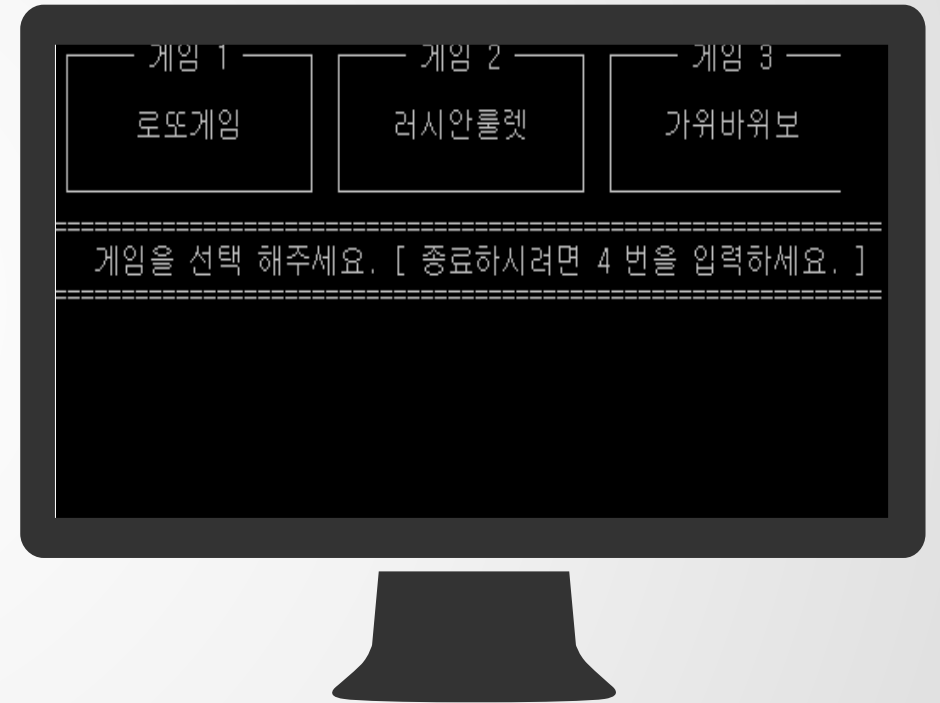
협업을 통해  
개발

02

OOP의 특성을  
이용한 개발

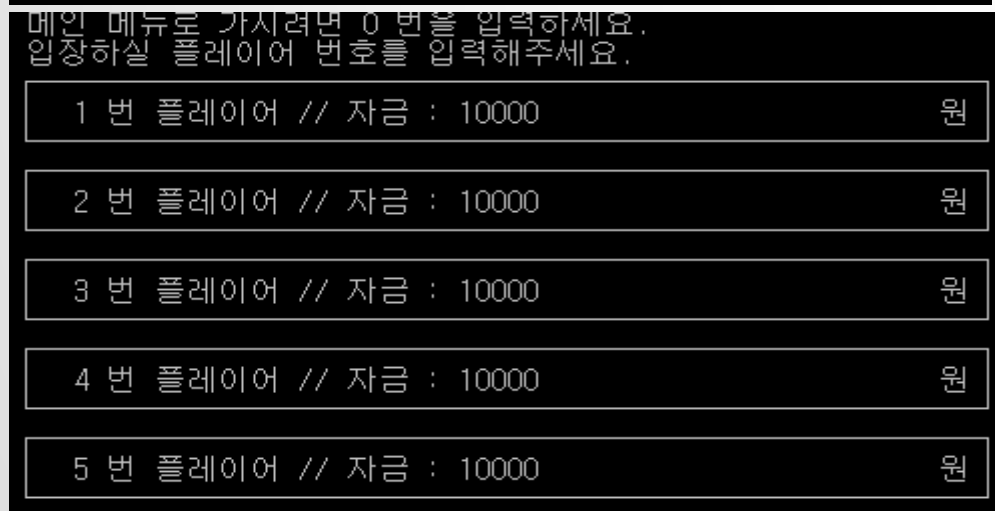
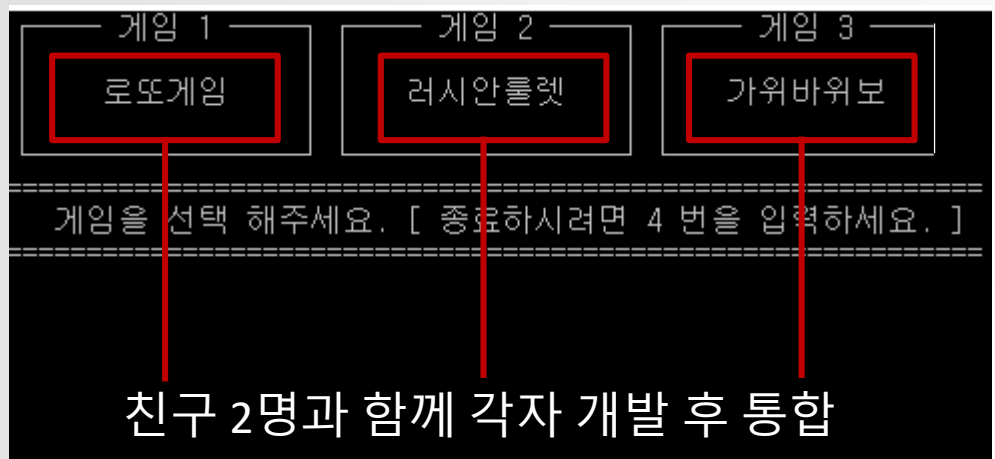
03

모듈화를 통한  
개발



# 01

## 협업을 통해 개발



- 각자가 만들고 싶은 간단한 게임을 기초로 만든 후에 소스를 통합하여 협업 방식의 기초로 개발.
- 본인은 '가위바위보' 게임과 게임의 전체적인 틀인 Main에 대하여 개발하였음.
- 각자 게임 이외의 기능들에 대해서는 서로 토론하고 상의하여 가장 최적화된 코드를 이용하였음.
- 서로의 코드 호환성에 대해 계속적으로 테스트/실행을 반복적으로 수행하며 개발하였음.

공통으로 개발한 부분.

모든 게임은 공통으로 개발한 부분에 대해서 같은 기능을 가지며, 동일한 화면을 출력

## 02

## OOP의 특성을 이용한 개발

몇개의 로또 게임을 생성 하시겠습니까?  
몇번째의 로또 게임으로 시작하시겠습니까?  
-  
각 게임이름이 들어감

원하는 만큼 플레이어의 수를 입력하십시오.  
10

메인 메뉴로 가지려면 0 번을 입력하세요.  
입장하실 플레이어 번호를 입력해주세요.

1 번 플레이어 // 자금 : 10000	원
2 번 플레이어 // 자금 : 10000	원
3 번 플레이어 // 자금 : 10000	원
4 번 플레이어 // 자금 : 10000	원

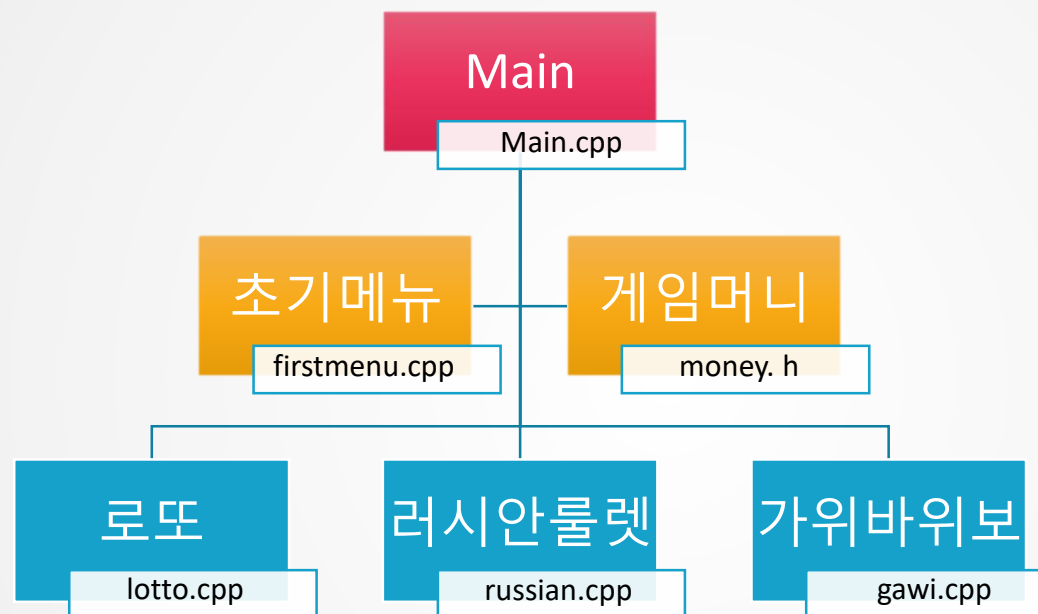
- OOP에 대해서 완벽한 이해는 하지 못하였지만 나름의 이해를 통해서 최대한 특성을 살리고자 하였음.
- ‘다형성’의 특징을 고려해 매 진행시에 사용자가 원하는 만큼의 서로 다른 객체(플레이어, 게임 개수)를 생성 할 수 있게 하여 각 객체마다 다른 정보를 지님.
- ‘캡슐화’의 특징을 고려해 클래스를 기반으로 개발하여 각 메소드가 직접 접근을 하지 않는 이상 서로에게 영향을 주지않게 프로그래밍 하였음.

## 03

모듈화를 통한  
개발

GameManager

- 참조
- 외부 종속성
- Header Files
  - firstmenu.h
  - gawi.h
  - lotto.h
  - money.h
  - russian.h
- Resource Files
- Source Files
  - firstmenu.cpp
  - gawi.cpp
  - lotto.cpp
  - main.cpp
  - russian.cpp



- 각 기능들을 분할(모듈화)을 통해서 효율적으로 다룰 수 있도록 함.
- 재사용성을 위해서 게임머니(money.h)소스를 모듈화.
- 모듈화를 통해서 리팩토링 개념의 기초를 다지고자 함.

# Finding the rabbit 개발 동기

- Python으로 할 수 있는 것들을 찾던 중에 예전에 했던 게임 관련 프로젝트(GameManager)에서 영감을 얻어 좀 더 완성도 있는 게임을 만들어보고 싶었음.
- 다른 게임 제작 툴(Unity 등...)을 많이 사용하지만 python 으로도 게임 제작이 가능하단 것과, 결코 다른 툴들에 비해서 뒤쳐지지 않는다는 것을 보여주고 싶었음.
- 나만의 게임을 만든다는 생각으로 스토리, 디자인, 음향 등을 직접 만들고, 편집하여 사용하여 좀 더 자신의 색이 강하게 드러날 수 있는 게임을 만들어보고 싶었음.



# Finding the rabbit

(pygame을 이용해 게임 만들기)

01

Python의  
범용성을 이해

02

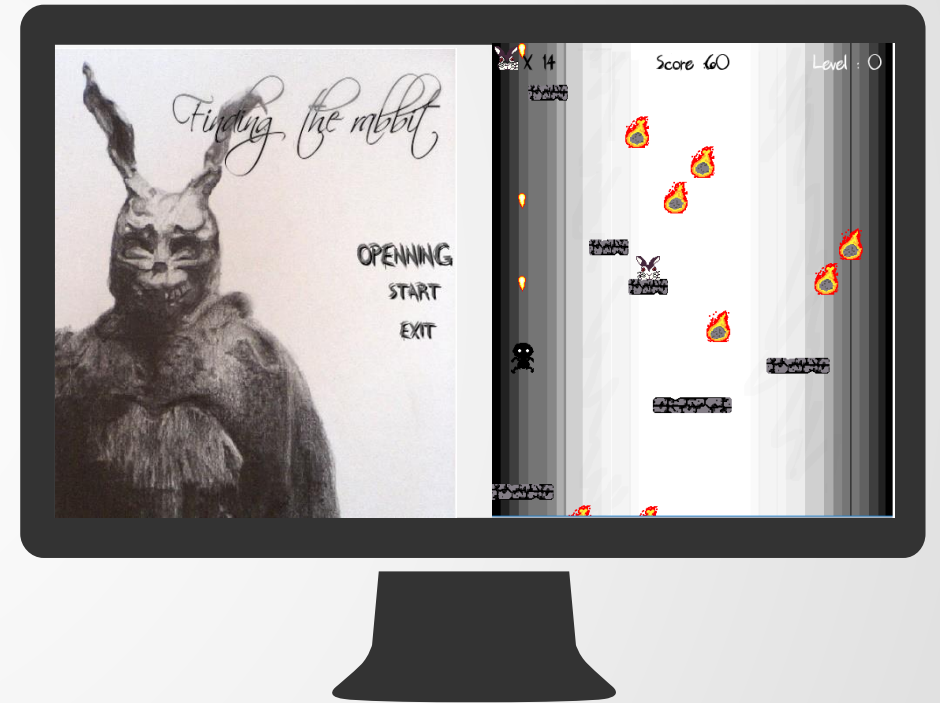
게임프로그래밍  
기본 로직

03

일급객체  
특성을  
이해하며 개발

04

다양한 툴 및  
라이브러리  
사용

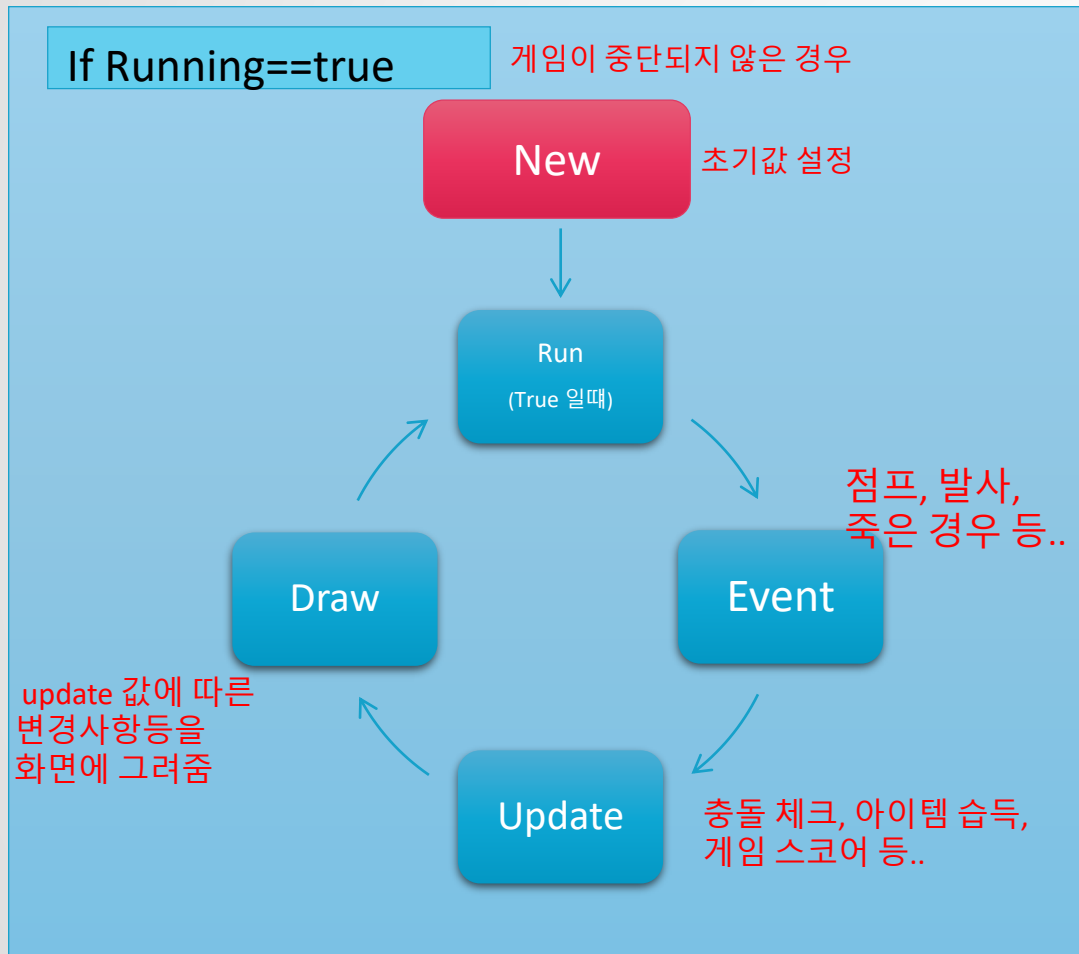


# 01

## Python의 범용성을 이해



- Python을 통해서 할 수 있는 것들을 생각하던 중에 pygame을 통하여 게임을 개발 할 수 있다는 것을 알고 라이브러리를 응용하여 개발.
- 인터프리터 언어이다 보니 게임 개발에 특화된 다른 언어에 비해 성능이 뒤처지지만 넓은 범위에서 사용성이 뛰어난 점과, 많은 라이브러리 등 python장점을 이용하여 개발.



- 게임 프로그래밍의 가장 기본적인 구현 로직을 이용하여 개발 함으로써 복잡하지 않은 형태의 간단한 게임을 만들 수 있음.
- 기본 로직을 응용하여 본 소스에서는 두 가지의 로직(게임기본, 게임시작~메뉴선택)을 사용하여 게임으로의 자연스러운 연출을 시도함.
- 단순히 게임 프로그래밍에만 국한되지 않고 다방면에서 사용이 가능함.

## 03

일급객체  
특성을  
이해하며 개발

```
def __init__(self, game):
    self.groups = game.all_sprites
    pg.sprite.Sprite.__init__(self, self.groups)
    self.game = game

self.player = Player(self) #self.player, Player 객체 생성

def load_images(self):
    self.standing_frames = [self.game.stand.get_image(2, 0, 28, 40),
                           self.game.stand.get_image(32, 0, 29, 40),
                           self.game.stand.get_image(62, 0, 28, 40)]
    for frame in self.standing_frames:
        frame.set_colorkey(WHITE)
    self.walk_frame_l = [self.game.move.get_image(0, 0, 30, 40),
                        self.game.move.get_image(30, 0, 30, 40)]
    self.walk_frame_r = []
    for frame in self.walk_frame_l:
        frame.set_colorkey(WHITE)

    for frame in self.walk_frame_l:
        self.walk_frame_r.append(pg.transform.flip(frame, True, False))
    self.jump_frame = self.game.jump.get_image(0, 0, 30, 40)
```

## Finding the rabbit

Sprite.py  
class(Player, Select,  
Platform ...)

Settings.py  
(상수, 기본값 설정)

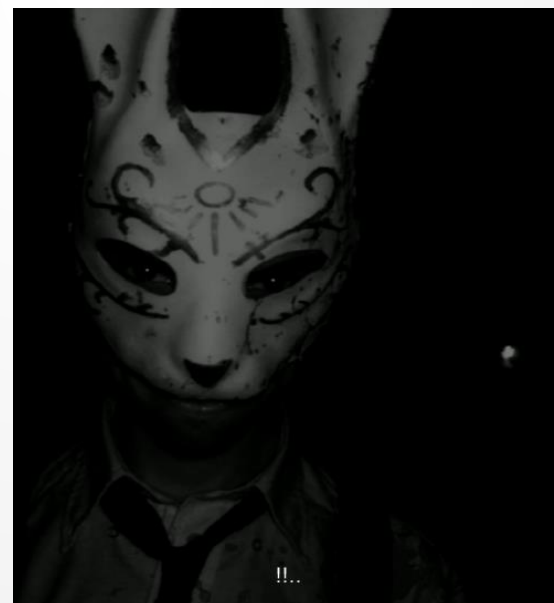
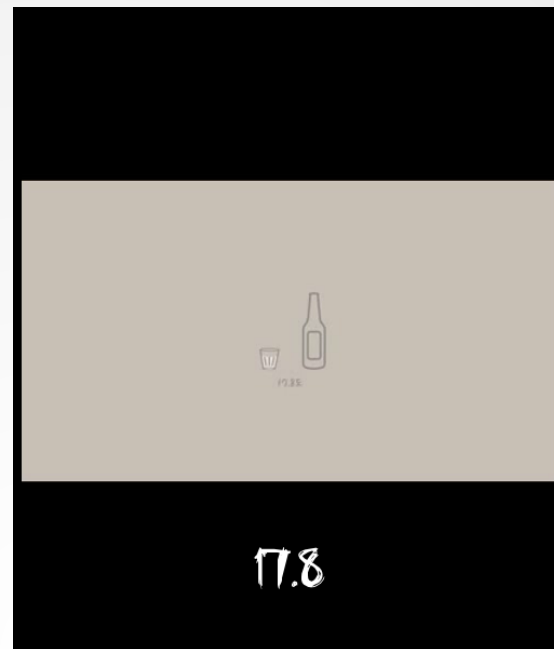
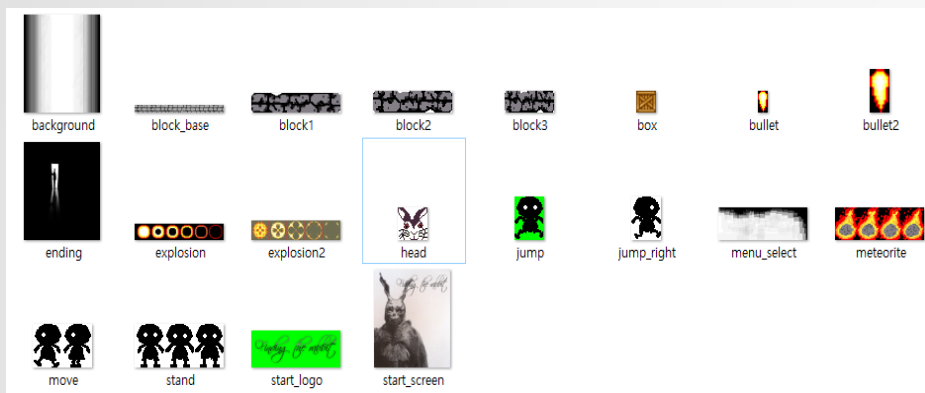
Start.py  
(게임 진행, Game class)

Game class  
내에서  
self로  
매개변수를  
받아 객체  
생성

- Python의 일급객체 특성을 이용, 객체를 매개변수로 받아 다른 클래스 내에서 접근하여 사용.
- 각각의 클래스(sprite.py)는 Game 클래스의 객체를 받아 메서드를 이용하여 변수들의 값을 정의.

# 04

## 다양한 툴 및 라이브러리 사용



- Aseprite(도트 툴)을 이용해 직접 캐릭터, 배경 등 제작.
- Moviepy 라이브러리를 통해 동영상 첨부를 이용, 자연스러운 연출
- AfterEffect(영상 제작)을 이용해 간단한 오프닝영상 제작 및 편집.
- 완성도 높은 제작을 위해서 다양한 툴/라이브러리를 사용함.

# Django Web site 개발 동기

- Django에 대해서 공부한지 얼마 안되었고, 간단한 사이트를 만들어 보고 싶다는 생각에서 개발을 시작하였음.
- Django의 장점 중 하나인 수많은 기능들을 간단하게 사용해보고, 응용하여 나의 생각이 어느정도 반영된 웹 사이트를 만들어 보고 싶었음.
- 예전에 AWS를 사용한 경험이 있었으나, 단순히 개인 서버 구축만을 목표로 하였기에 많은 경험을 하지 못하였고, 이번 기회에 AWS를 적극 사용하여 클라우드 서버를 이용해 효율적으로 구축해보고 싶었음.

# Django Web site

01

장고의 기본적인  
기능들로 개발

02

AWS 스택을  
이용하여 배포

03

Bootstrap3  
템플릿 이용

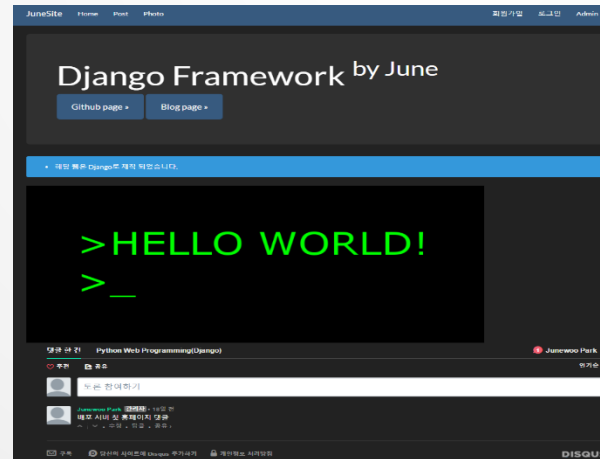
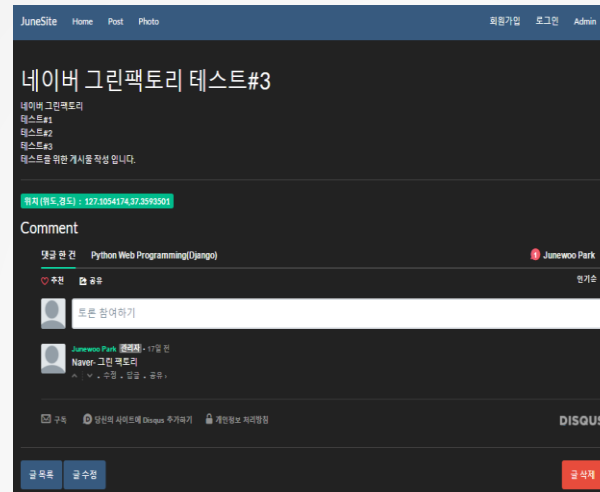
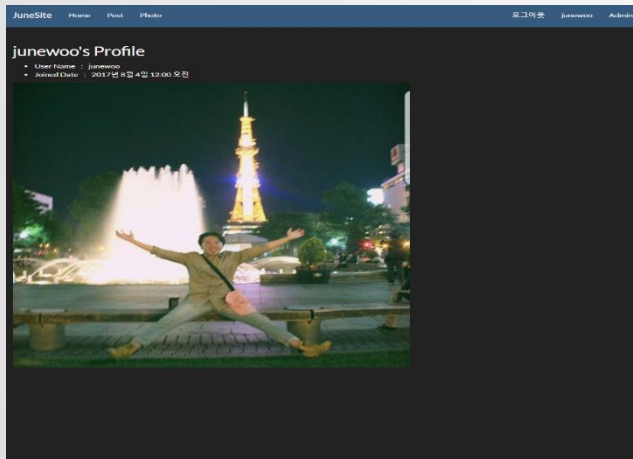
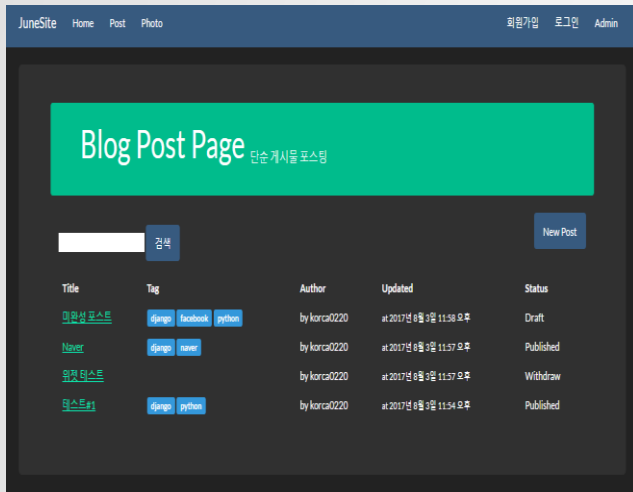
04

API 를 이용하여  
개발



# 01

## 장고의 기본적인 기능들로 개발



- 장고의 가장 기본적인 페이지(list, detail)를 FBV/CBV 방식으로 개발.
- 로그인/회원가입을 지원하는 장고의 내장 라이브러리 auth를 이용하여 개발.
- Template페이지에서 POST방식의 경우 장고 미들웨어에서 지원하는 csrf\_token을 이용하여 CSRF공격에 대해서 방어할 수 있도록 간단하게 개발.



## 02

AWS 스택을  
이용하여 배포

MySQL junesite-v2 available 1.17% 0 Connections

Configuration Details		Security and Network	
ARN	arn:aws:rds:ap-northeast-2:173582888272:db:junesite-v2	Availability Zone	ap-northeast-2c
Engine	MySQL 5.6.35	VPC	vpc-d80a2ab1
License Model	General Public License	Subnet Group	default (Complete)
Created Time	August 3, 2017 at 10:37:00 PM UTC+9	Subnets	subnet-46bf8e2f subnet-78d5bc35
DB Name	junesite_v2	Security Groups	eb-junesite (sg-393d6651) (active)
Username		Publicly Accessible	Yes
Option Group	default:mysql-5-6 (in-sync)	Endpoint	
Parameter Group	junesite-eb-utf-8 (in-sync)	Port	3306
Copy Tags To Snapshots	No	Certificate Authority	rds-ca-2015 (Mar 5, 2020)
Resource ID			

IAM DB Authentication Enabled No

**RDS**

이름 ↑ S3

media

static

## junesite-v2 ElasticBeanstalk

JunesiteV2

Environment tier: Web Server

Platform: 64bit Amazon Linux 2017.03 v2.4.2 running Python 3.4

Running versions: Junesite-widget-v3

Last modified: 2017-08-03 23:51:10 UTC+0900

URL: JunesiteV2.ap-northeast-2.elasticbeanstalk.com

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
JunesiteV2	i-009783a1bbfe6a5ae	t2.micro	ap-northeast-2a	running	2/2 checks ...	None

**EC2 instance**

- RDS(Relational Database Service)를 이용하여 AWS에서 관리하기 쉽도록 이용.
- S3 저장소를 이용하여 AWS 웹 서비스에서 static/media 파일들을 서빙.
- EB(ElasticBeanstalk)를 이용하여 헬스체크, 오토 스케일링 및 모니터링을 쉽게 할 수 있음
- EC2 가상서버를 이용하여 보안 및 네트워크 구성과 스토리지 관리를 효율적으로 함

## 03

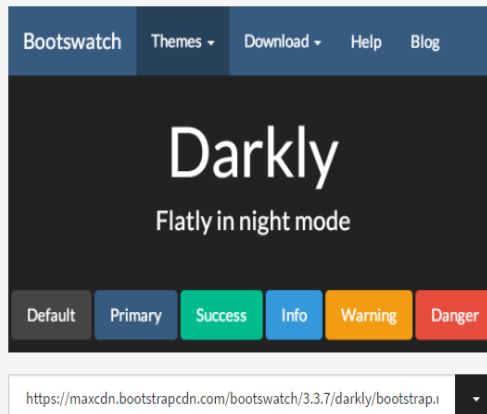
Bootstrap3  
템플릿 이용

Try it!



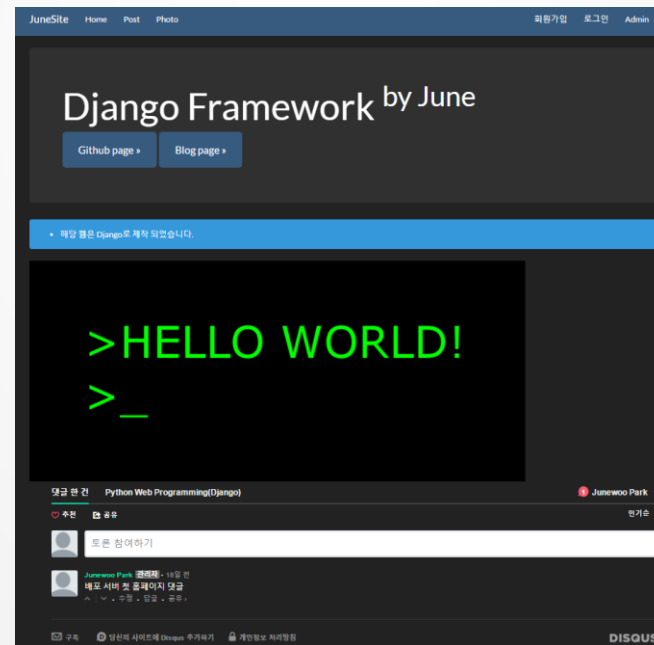
Click to copy

Try it!



```
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootswatch/3.3.7/darkly/bootstrap.min.css">
```

```
<script src="//code.jquery.com/jquery-2.2.4.min.js"></script>
<script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```



- Bootstrap3의 CDN을 템플릿에 삽입하여 간단하게 이용.
- Bootswatch의 CDN을 삽입하여 테마를 간단하게 사용가능.
- Bootstrap3을 이용하여 템플릿 파일의 행/열 구성 및 container 구성.

# 04

## API 를 이용하여 개발

### Signup Form

사용자 이름

비밀번호

4\*3=?

로그인

Facebook아이디로 로그인

Kakao아이디로 로그인

Naver아이디로 로그인

### Post Form

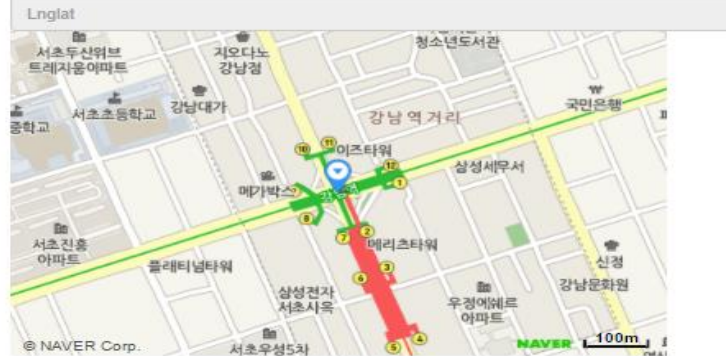
User

제목

포스트의 제목(최대 100자)

내용

Lnglat



Tag set

Status

제출

- 네이버의 맵API를 받아서  
게시글 작성시 현재 위치  
경도/위도를 입력 할 수  
있도록 위젯 개발
- 로그인 시  
Facebook/Kakao/Naver 등  
소셜 정보를 가지고  
로그인이 가능하도록 각  
API ClientID/SercretKey 를  
받아서 개발.

# 가상 화폐 예측 개발 동기

- Machine Learning(ML)에 흥미를 가지게 되고 그룹 스터디를 통해 같이 공부하면서 적당한 난이도의 프로젝트를 개발하고자 함
- Deep Learning 기반 알고리즘 중 RNN(순환 신경망)을 가장 어려워 했고, 프로젝트를 통해 실제 사용을 통해서 알고리즘에 대한 이해도를 높이고 싶었음
- 사전에 주식 시장 예측을 통해 RNN을 공부 했기 때문에 이를 응용하여 최근 이슈가 되고있는 가상 화폐 시장을 예측해 보고 싶었음

# 가상 화폐 예측

(Machine Learning을 이용한)

01

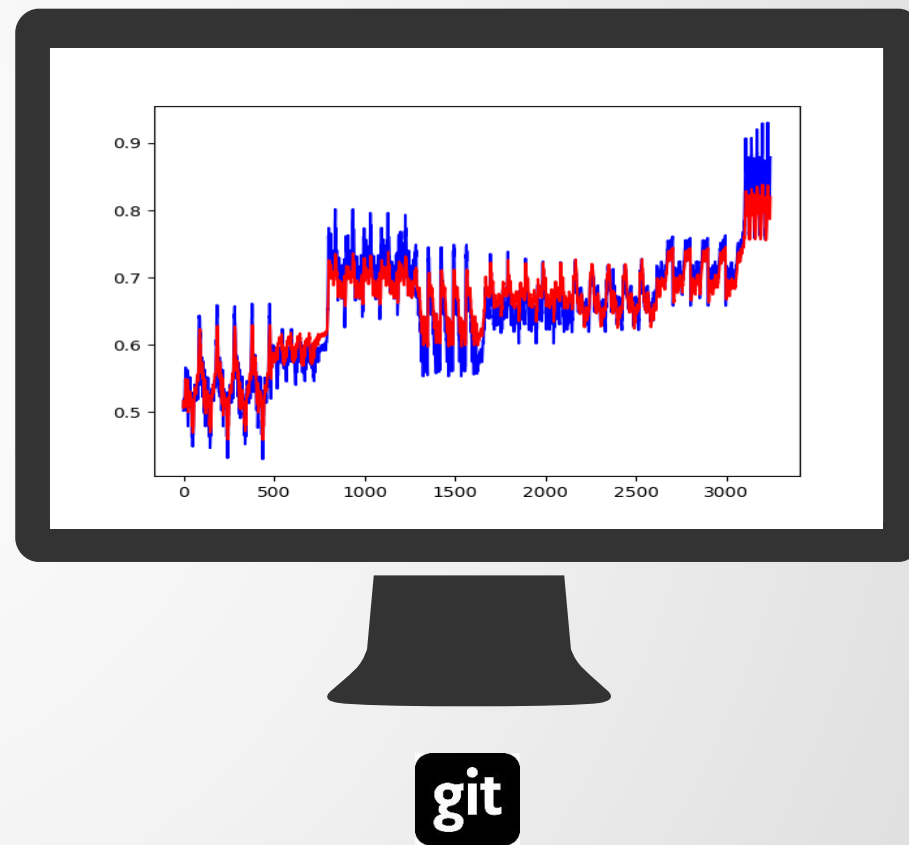
순환  
신경망(RNN)을  
사용하여 예측

02

실시간 데이터  
수집

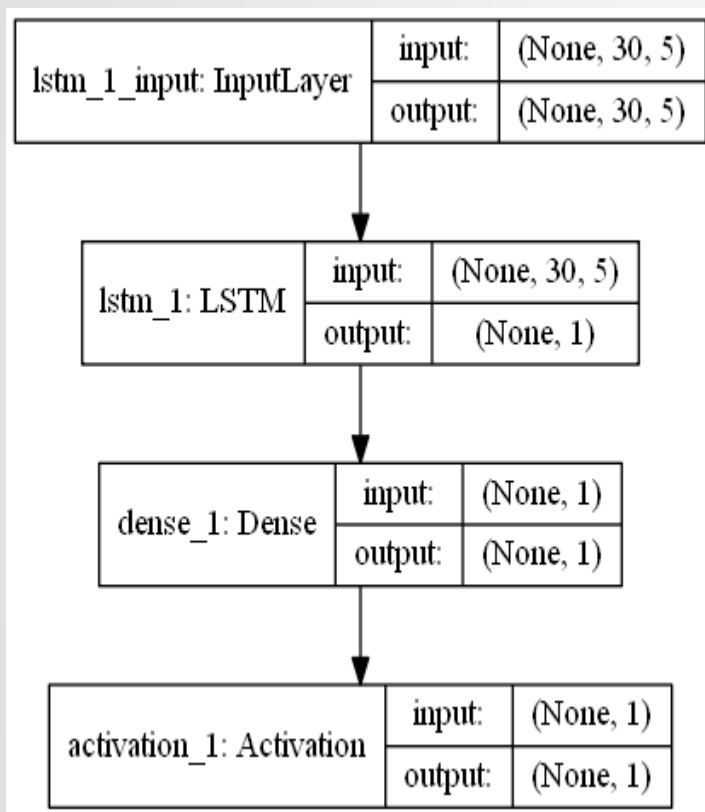
03

그래프를 통하여  
시각적으로 예측  
결과 확인



## 01

순환  
신경망(RNN)을  
사용하여 예측



```

model = Sequential()

# RNNs 의 변형 LSTM
model.add(LSTM(units=1, input_shape=(seq_length, data_dim),
               return_sequences=False))

model.add(Dense(1)) # 뉴런
model.add(Activation('linear')) # 활성화 함수
model.load_weights('model.h5') # model (가중치 값) 불러오기

# loss function, 최적화 함수
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam')

model.summary()

# Layer 구성 방식을 .png로 export
plot_model(model, to_file=os.path.basename(__file__) + '.png', show_shapes=True)

print (trainX.shape, trainY.shape)

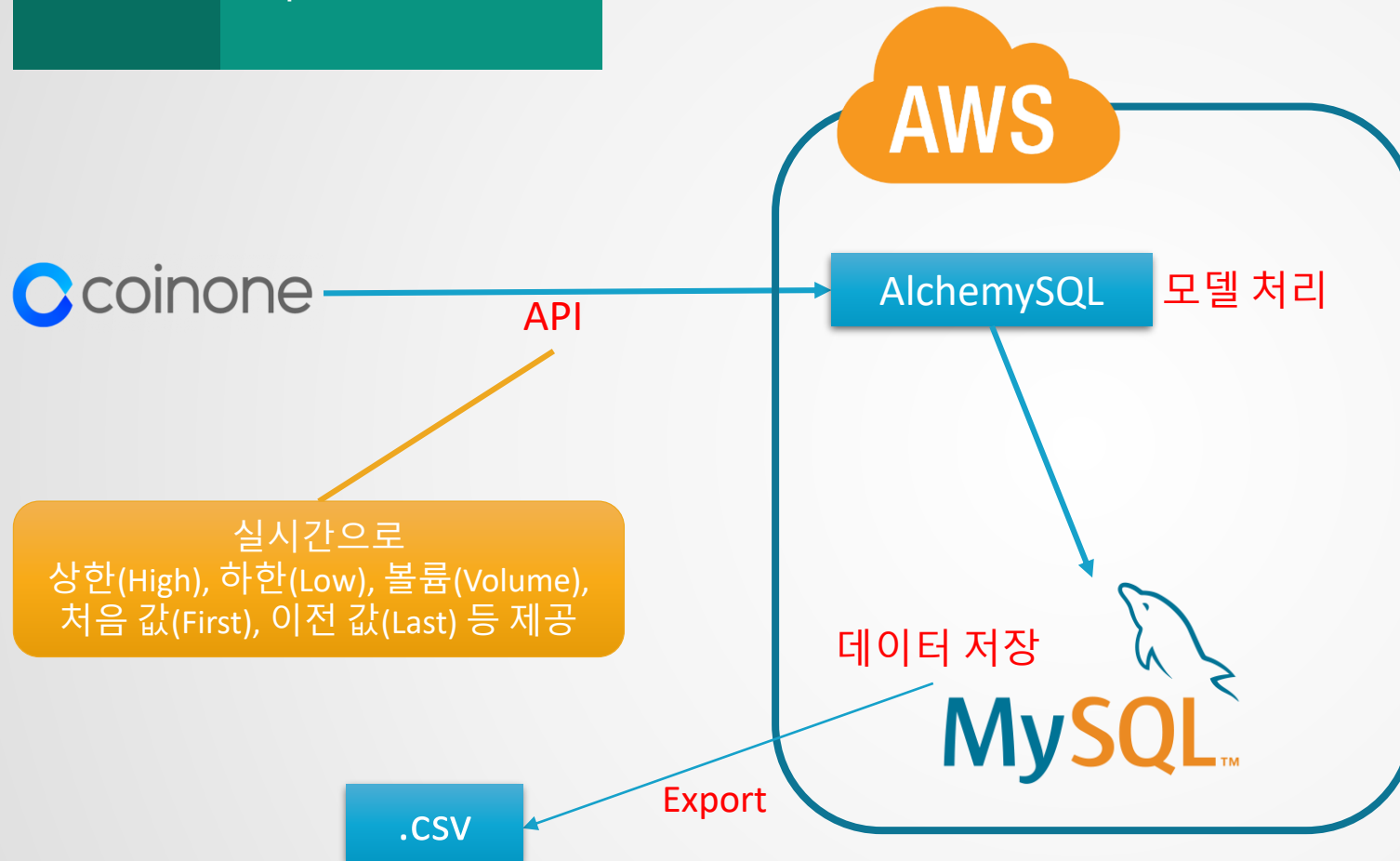
# model.fit(trainX, trainY, epochs=200) # <--- model Load시 사용하지 않아도 됨
# model.save_weights("model.h5") # model 저장

testPredict = model.predict(testX)
# 차이값을 줄이기 위하여 MinMaxScaler 재적용
testY = scaler.fit_transform(testY)
testPredict = scaler.fit_transform(testPredict)
  
```

- 이전의 값들(가상 화폐의 이전 값, 상한 값 등..)을 이용한 다음 값 예측을 위해 RNNs의 변형인 LSTM 알고리즘 사용
- 활성화 함수로 linear(1차 곡선)인 선형 함수를 사용했지만 후에 Layer가 깊어지면 tanh 비선형 함수로 개발
- Train을 끝내고 test한 Y값(실제 값)과 Predict(예측 값)의 차이 값을 줄이기 위하여 한번 더 MinMaxScaler를 통하여 수치 안정화

## 02

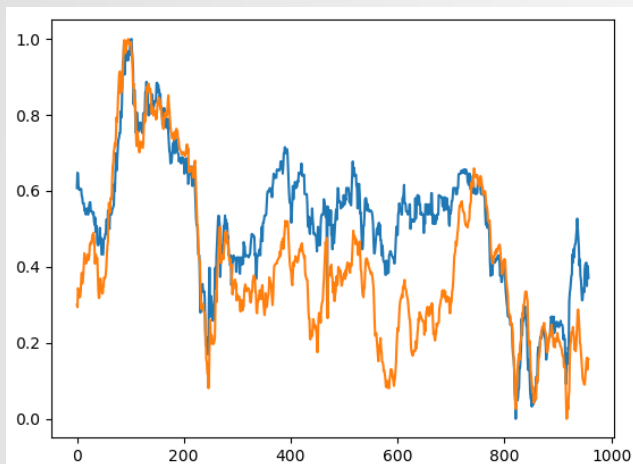
## 실시간 데이터 수집



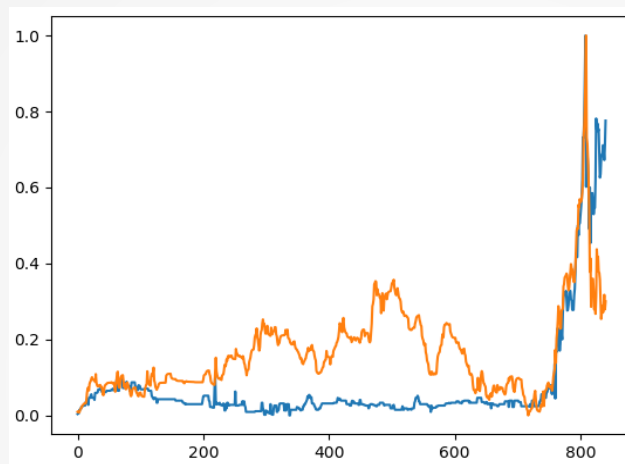
- Coinone(가상 화폐 시장 사이트)에서 주는 객관적인 자료를 API로 받아서 높은 신뢰도로 개발
- AWS를 이용하여 물리적인 사용없이 서버를 계속적으로 동작하기 때문에 실시간 데이터 수집이 가능
- AlchemySQL의 ORM 객체를 통해 받아온 API를 동적으로 테이블 생성
- Mysql DBMS를 이용해 저장 후에 csv로 Export 후 데이터로 사용

## 03

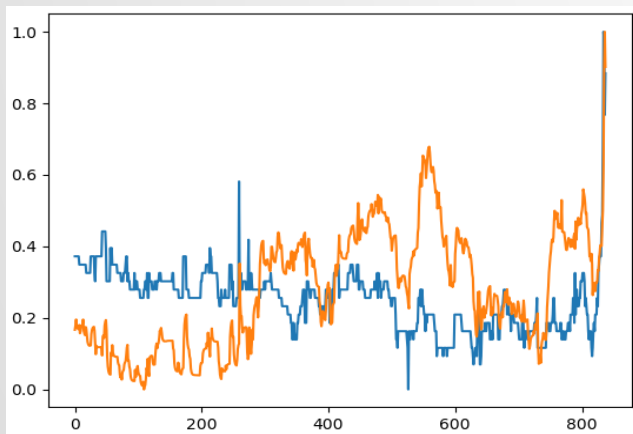
그래프를  
통하여  
시각적으로  
예측 결과 확인



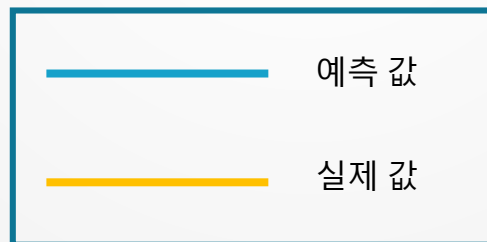
<비트코인>



<비트코인 캐쉬>



<이더리움 클래식>



- Matplotlib 라이브러리를 통해 예측 값을 그래프로 표현
- 실제 값과 비교해 가격의 차이가 있지만 상/하향 곡선의 예측치는 거의 정확함
- 결과적으로 보았을 때, 데이터 수치가 얼마 되지 않는 것에 비해서(약 2800개)높은 정확도를 보이는 것을 알 수 있음





감사합니다