

[시간표]

- \* 09:10 ~ 시작 : (가능한 9시 전에)
  - \* 점심 : 11시30분~오후 1시 (13시)
  - \* 오후 5시 45~**50** : 끝
- 

[학습 주제]

1. 실전정보기술

- 소프트웨어공학 : 개발방법론, 프로젝트, 요구사항, 품질
  - IT 인프라 => 운영체제 (윈도우, 리눅스, 모바일)
  - 최신 트렌드 기술 (AI/인공지능, 파이썬 + 활용)
- => 정보처리기사 + 보안기사

2. 서버 프로그램 구현

- 네트워크 기반 이론
  - MSA, TDD/Test.
  - Front-End / Back-End
  - Java
- => 테스트 전문가 (CSTS) + 보안기사

3. DB/SQL

- Database 기본 이론 / 실습
  - 도커, CI/CD
- => SQLD
- 

\* 결론 : 요구사항은 우리에게 무척!!!! 중요하다. !!!

\* 요구사항이 어떤 것인지 알아보자.

- 고객이 명시적, 암묵적으로 원하는 사항으로 우리 소프트웨어가 제공(달성)해야 할 목표.
- 요구사항의 분류
  - 1) 기능적 요구사항 : ID와 PW를 입력해서 로그인을 해야 한다.
  - 2) 품질적 제약사항 : 500명이 동시에 로그인을 시도해도 1초 이내 응답  
중요한 기능이므로 24x364  
=> 소프트웨어 구조, 동작방식 (SW 아키텍처)
  - 3) 보안 요구사항 : 패스워드를 반복해서 5번 잘못 입력하면 LOCK.
  - 4) (관리적 요구사항)

- 요구사항을 검증하는 방법
  - 1) 완전성 : 요구사항이 빠짐없이 모두 도출/명세.
  - 2) 구체성 (구현 가능성, 테스트 가능성)
  - 3) 명확성 : 용어가 동일한 의미로 해석되는가

\* 요구사항을 잘 관리 !!!

1) 요구사항 도출 (식별) : 찾는다 !!

- 베이스라인 기법 : 과거의 경험 (체크리스트)을 기반으로 요구사항을 도출.
- 시나리오 기법 : 잘 모르는 상황에서 스토리를 전개해서 요구사항을 도출 (페르소나 기법)
- 델파이 기법 : (전문가들이 익명성을 갖고) 의견을 나누어 결정.
- 워크샵, 인터뷰

2) 요구사항 분석 : (이해관계자 마다) 상호 충돌을 해결/조정.

3) 요구사항 명세 : 요구사항 정의서 작성. ==> 개발의 베이스라인 을 선정

4) 요구사항 검증

-----

○ 다양한 이해관계자 있다.

○ 공공

- 공공기관, 정부, 공사, 교육, 의료.
- 목표 : 국민의 행복 증진과 대국민 서비스 향상을 위하여....  
안전한 도시 생활, 국민의 여가활동.  
일자리 창출과 국민의 삶의 질 향상을 위하여....

1) 전자 정부 서비스

- 예전 : 각 부처/기관들이 독자적 시스템, 데이터 형식.... 다 달랐다.
- 개선 : 데이터 표준화/토합, H/W+ S/W 사양 통합 (플랫폼 통합), 인터페이스 통합(웹 기반)  
+ 오픈소스, 정보보안, 모바일...  
==> 전자정부프레임워크 : Spring, Java, ORM.

2) 일자리 창출 (산업 활성화)

- a) 신기술, 미래 유망 산업/기술 발굴 선정 => 4차 산업혁명, AI/빅데이터/클라우드
- b) 교육, 인력 양성
- c) 시범 사업 (정부가 투자해서 사업)
- d) 민간으로 확대, 사업을 고도화.
- e) 인프라 만들도 (기술, 제도, 법률)
- f) 민간에서 경쟁.

○ 회사의 공통

- 목표 : 계속 기업 (회사가 망하지 않고 계속 살아 남는 것)

- 현실 : 사용자의 요구사항은 계속 변경 (개인화, 다양화)  
시장 경쟁음 심화, 변화 가속  
AI 신기술.. 등장.

○ 민간 기업 / 회사 (IT가 주 업이 아닌 회사)

- 고유한 사업 영역이 있고, IT의 지원이 필요 : 금융, 보험, 제조.

\* 우리보다 앞선 회사 (기업, 모델)을 참고, 이기는 전략을 수립.

<지방의 모 은행>

1) 국민은행 (선진사례) 분석 : 경쟁에서 이기고 싶다. (To-Be 모델)

2) 지방의 모은행 : 우리의 현황, 상태를 점검 (As-Is 분석)

3) 갭 분석 : 어떻게 해야야 To-Be가 될 수 있을지 차이점을 분석.

=> 어떻게 일을 할까? 어떤 조직, 어떤 데이터.....

(사업계획 -> RFP -> 제안서 -> 사업을 수주 -> 프로젝트를 수행)

○ 회사 (IT 주업인 회사) : 구글, 네이버

=> 신기술, 시장 경쟁력 있는 서비스/제품을 개발 하고 경쟁 우위

1) 시장조사

2) 제품 기획 / 마케팅 기획.

3) 요구사항 정의 : 스토리 보드, UI 가이드, 기능 명세서.

4) ~~~~ 개발~~~~ 출시 : Agile, DevOps,

[RFP]

- <https://bid.kogas.or.kr:9443/supplier/index.jsp>

- 1 입찰  
공고  
문 [https://bid.kogas.or.kr:9443/supplier/bid/bid\\_download\\_attfile.jsp?notice\\_code=2021082420&seq=10](https://bid.kogas.or.kr:9443/supplier/bid/bid_download_attfile.jsp?notice_code=2021082420&seq=10)
- 2 용역  
계약  
서(안  
) [https://bid.kogas.or.kr:9443/supplier/bid/bid\\_download\\_attfile.jsp?notice\\_code=2021082420&seq=11](https://bid.kogas.or.kr:9443/supplier/bid/bid_download_attfile.jsp?notice_code=2021082420&seq=11)
- 3 과업  
지시  
서 [https://bid.kogas.or.kr:9443/supplier/bid/bid\\_download\\_attfile.jsp?notice\\_code=2021082420&seq=12](https://bid.kogas.or.kr:9443/supplier/bid/bid_download_attfile.jsp?notice_code=2021082420&seq=12)
- 4 제안  
요청  
서 [https://bid.kogas.or.kr:9443/supplier/bid/bid\\_download\\_attfile.jsp?notice\\_code=2021082420&seq=13](https://bid.kogas.or.kr:9443/supplier/bid/bid_download_attfile.jsp?notice_code=2021082420&seq=13)
- 5 (별첨  
)  
공동  
수급  
표준  
협정  
서  
---

-----  
[테스트 레벨]

- 1) 단위 테스트 : 단위 모듈의 (기능성을 중심으로) 테스트.
- 2) 통합 테스트 : 단위 테스트 완료 후 서로 약속한 인터페이스(함수) 중심으로 테스트.
- 3) 시스템 테스트 : ~~~성 (품질적 제약사항)을 중심으로 테스트.
- 4) 인수 테스트 : 고객 관점에서 요구사항이 잘 만족되는지 테스트.
- 5) (설치 테스트)

\* 보안 요구사항:

- 1) 기밀성 : 암호화, 접근통제
- 2) 무결성 : 정보가 변하지 않다는 것을 알수 있는 특징, 정보가 변경되면 변경된 것을 알 수 있는 특징. (해시함수, 해시값)
- 3) 가용성 : 정당한 권한을 가진 사용자가 요청하면 항상 서비스 제공. => 이중화 !!!
- 4) 법적 준거성

-----  
[자격증 vs. 면허증]

○ 자격증 : 하나보다는 연관된 여러가지를 취득하는 것이 좋다.

- \* 기본 : 정보처리기사.
- \* 인프라 : SQLD + 리눅스마스터 + 네트워크관리사.  
클라우드.
- \* 정보보안기사.

○ 기사 취득후 5년~7년이 지난면 시험을 볼 수 있는  
(면허 조건을 가진) 자격증.

---

- \* 정보시스템 감리원 : 5일 교육 + 시험.
- \* 개발보안진단원

---

- \* 기술사 / 감리사.

---

- \* (CPPG : 개인정보 1년 자격인정)
- \* PIA
- \* ISMS

----

○ 빅데이터 (다양한 데이터 + 대량의 데이터 + 증가하는 속도 빠르다)

- 정형 데이터 (틀이 있다, 틀을 갖추고 사용하는 데이터)  
create table (이름, 전화번호)
- 비정형 데이터.  
insert (온도)..... insert(온도, 습도)

```
{ "name" : "chris", "age" : 10 }
```





## [둘째날]

\* 윈도우 운영체제에서 리눅스를 사용할 수 있는 방법?

- VM : VirtualBox (OSS), WMWare, Parallels
- Cloud에 리눅스 머신 생성하고 원격 접속
- WSL2 (Windows Subsystem for Linux v2)  
=> Power Shell을 관리자 모드로 실행 후  
( wsl --install -d Ubuntu)
- Docker

## ○ 정보보안의 기본 이론

\* 정보보안의 요구사항

- 기밀성 (암호화, 접근 통제), 무결성 (해시함수), 가용성 (이중화), 법적 준거성.

\* 암호화

- 양방향 암호화 (암호화, 복호화), 단방향 암호화 (암호화, 복호화X)
- 대표적인 단방향 암호화 기법 : 해시 (해시함수)

<해시함수> : sha-256

- 1) 단방향 : 암호화는 할 수 있으나, (수학적 방법으로는) 복호화 할수 없다.
- 2) 고정길이 출력 : 입력값의 크기는 가변이나 출력값의 크기(길이)는 고정
- 3) 입력값이 다르면 출력값도 다르다.
- 4) 2개의 결과값 (MD, Message Digest, 해시값)이 같으면, 2개의 입력값이 같다.

12345 % 10 ==> 5

123 % 10 ==> 3

345 % 10 ==> 5

==> 무결성을 확보하는 데 사용한다

## [블록체인]

○ 블록체인 = 블록 + 체인

= (Header + Data) + 체인

\* <https://hey.technonia.com/>

\* <https://wikidocs.net/>

[KOSA20240122](#)

<https://colab.research.google.com/>

<https://pypi.org/>

-----

### # 문자열 연습

```
str = 'Hello World'
print(str)
```

```
str = "Hello World"
print(str)
```

```
str = """Hello
World"""
print(str)
```

```
str = """Hello
World"""
print(str)
```

-----

-----

### # 리스트

```
var = [1,'chris',['tommy', 'harry'], ('hans', 2)]
print(var)
print(var[3])
print(var[3][0])
print(var[-1])
```

```
var[0]=7
print(var[:2])
```

```
del var[2:]
print(var)
```

```
print(len(var))
```

-----

-----

## # 딕셔너리

```
var = {'name' : 'chris', 'age' : 23, 1 : ['a', 'b', 'c']}  
print(var)  
print(var['name'])
```

```
var['age'] = 33  
print(var)  
del var[1]  
print(var)  
var[1] = 'Hi'  
print(var)
```

```
var.update({'name' : 'hans', 'age' : 30})  
print(var)
```

## # 얕은 복사와 깊은 복사

```
data_1 = {'1st' : 'chris', '2nd' : 'tommy', '3rd' : 'harry'}  
data_2 = data_1  
print(data_1)  
print(data_2)  
data_1['1st'] = 'hans'  
print(data_1)  
print(data_2)
```

```
data_1 = {'1st' : 'chris', '2nd' : 'tommy', '3rd' : 'harry'}  
data_2 = data_1.copy()  
print(data_1)  
print(data_2)  
data_1['1st'] = 'hans'  
print(data_1)  
print(data_2)
```

-----

-----

### # 입출력 함수의 활용

```
import math

def getArea(length):
    area = length*length*math.pi
    return area

value = input('Input a number: ')
area = getArea(int(value))
print('Input number : {0} => Area : {1}'.format(value, area))
```

-----

-----

### # 함수

#### # 파라미터도 없고 리턴값도 없음

```
def hello():
    print("Hello World")
```

```
hello()
```

#### # 파라미터와 리턴값이 있음

```
def add(x, y):
    print(x)
    print(y)
    return x + y
```

```
val_x = 1
val_y = 2
val_sum = add(val_x, val_y)
print(val_sum)
```

# 여러개의 값을 반환하는 함수

```
def calc(a, b):
    return a + b, a - b
```

```
result = calc(1,3)
print(result)
```

```
x, y = calc(1,3)
print(x)
```

-----

-----

# Lamda

# 일반적인 함수

```
def add(a, b):
    return a + b
result = add(1,2)
print(result)
```

# 간단한 람다함수

```
add = lambda a, b: a + b
result = add(1,3)
print(result)
```

```
result = (lambda a, b: a + b)(1,4)
```

```
print(result)
```

-----

-----

### # File I/O와 예외처리

#### # 새로운 파일에 데이터 쓰기

```
file = open("test.txt", "w")
for data in range(1, 11):
    file.write(f'{data} line\n')
file.close()
```

#### # 기록한 데이터 읽기

```
file = open("test.txt", "r")
while True:
    line = file.readline()
    if not line:
        break
    print(line)
file.close
```

#### # 예외처리 : try ~ except

```
try:
    myFile = open('NotExist.txt', 'r')
except FileNotFoundError as e:
    print('파일이 존재하지 않습니다.')
    print(e)
```

-----

#### ○ 교수님의 숙제

학점관리 시스템을 개발하고 싶다.

국어, 수학 의 평균

70 점 이상 : A

40 점 이상 : B

: C

AI(인공지능)으로 개발.

String cacGrade(KOR, MATH)

```
{
    mean = (KOR + MATH) / 2
    if (70<=mean) return 'A'
    ..
    ...
}
```

1) 학점 데이터를 확보 (2만건)

KOR	MATH	GRADE
90	90	A
10	40	C
..		
..		

2) 데이터 특징 파악, (필요시) 전처리

3) 학습용 데이터 (15000건) + 검증용 데이터 (5000건)

4) 파이썬/구글이 준 머신러닝 라이브러리에 입력 (학습데이터)

==> Weigh 를 찾아줘.

5) 검증용 데이터로 정확도(품질) 검증.

==> 99.9 % 정확도로 맞추고.

6) weight 파일을 드리겠습니다. (모델에 탑재하세요)

==> SW 개발.

7) 다음학기 학생의 점수를 입력 (80, 70) ==> 'A'

(1) 학습을 위한 데이터를 우리가 직접 만들자.

#데이터 준비

import random



# 점수를 계산해서 평균에 따른 학점을 반환

# 편의상 A, B, C로 구분함.

```
def get_grade(kor, math):
```

```
    mean = int((kor + math) / 2)
```

```
    grade = "C"
```

```
    if 70 <= mean:
```

```
        grade = "A"
```

```
    elif 40 <= mean:
```

```
        grade = "B"
```

```
    return grade
```

# 20000개 데이터를 만들어 CSV에 기록한다.

```
fp = open("grade.csv", "w", encoding="utf-8")
```

```
fp.write("KOR,MATH,GRADEWrWn")
```

```
for i in range(20000):
```

```
    kor = random.randint(10, 100)
```

```
    math = random.randint(10, 100)
```

```
    grade = get_grade(kor, math)
```

```
    fp.write("{0},{1},{2}WrWn".format(kor, math, grade))
```

```
fp.close()
```

(2) 만들어진 데이터의 특징을 살펴보자 (그림)

# 특징 보기

```
import matplotlib.pyplot as plt
```

```
import pandas as pd
```

# pandas로 csv 파일 읽기

```
grade = pd.read_csv("grade.csv", index_col=2)
```

**# 그래프로 표시하기**

```
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
```

**# Lable 색상 표시하는 함수**

```
def scatter(label_text, color):
    b = grade.loc[label_text]
    ax.scatter(b["KOR"], b["MATH"], c=color, label=label_text)
```

```
scatter("A", "red")
scatter("B", "blue")
scatter("C", "yellow")
ax.legend()
```

-----

3) SVM (파이썬에서 제공하는 Support Vector Machine)을 이용해 보자.

-----

**# SVM으로 머시러닝**

```
from sklearn import svm, metrics
from sklearn.model_selection import train_test_split
import pandas as pd
```

**# grade가 저장된 CSV 파일 읽기**

```
grade = pd.read_csv("grade.csv")
```

**# 데이터 전처리 (학습을 위하여 데이터를 분리하고 가공-정규화)**

```
label = grade["GRADE"]
kor = grade["KOR"] / 100 # 점수는 최대 100점
math = grade["MATH"] / 100
data = pd.concat([kor, math], axis=1)
```

**# 학습 및 테스트 데이터로 분리**

```
data_train, data_test, label_train, label_test = W
```

```
train_test_split(data, label)
```

### # 학습하기

```
model = svm.SVC()
model.fit(data_train, label_train)
```

### # 테스트 데이터로 예측하기

```
predict = model.predict(data_test)
```

### # 결과 확인하고 평가하기

```
ac_score = metrics.accuracy_score(label_test, predict)
cl_report = metrics.classification_report(label_test, predict)
print("Accuracy =", ac_score)
print("Report =\n", cl_report)
```

<https://playground.tensorflow.org/>

### #딥러닝

```
from keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation
from keras.callbacks import EarlyStopping
import pandas as pd, numpy as np
```

### # grade가 저장된 CSV 파일 읽기

```
grade = pd.read_csv("grade.csv")
```

### #데이터 전처리 (MLP 학습을 위해 데이터를 가공)

```
grade["KOR"] /= 100 # 점수는 최대 100점
grade["MATH"] /= 100
```

```
X = grade[["KOR", "MATH"]].to_numpy() # 입력데이터 (독립변수)
lable_class = {"A": [1,0,0], "B": [0,1,0], "C": [0,0,1]} # Label (종속변수)
Y = np.empty((20000,3))
for i, j in enumerate(grade["GRADE"]):
    Y[i] = lable_class[j]

# 학습 데이터와 테스트 데이터 분할
X_train, Y_train = X[1:15001], Y[1:15001]
X_test, Y_test = X[15001:20001], Y[15001:20001]

# 학습을 위한 모델 구성하기
model = Sequential()
model.add(Dense(512, input_shape=(2,)))
model.add(Activation('relu'))
model.add(Dropout(0.1))
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.1))
model.add(Dense(3))
model.add(Activation('softmax'))

model.compile(
    loss='categorical_crossentropy',
    optimizer="rmsprop",
    metrics=['accuracy'])

model.summary()

# 모델을 이용한 학습
model.fit(
    X_train, Y_train,
    batch_size=100,
    epochs=20,
    validation_split=0.1,
```

```
callbacks=[EarlyStopping(monitor='val_loss', patience=2)],
verbose=1)
```

```
model.save_weights('my_model_weights.h5')
model.load_weights('my_model_weights.h5')
```

#### # 테스트 데이터를 이용한 평가

```
score = model.evaluate(X_test, Y_test)
print('loss=', score[0])
print('accuracy=', score[1])
```

#### # 생성한 모델 이용

```
X_Data = X[0:10]
```

```
result = model.predict(X_Data)
print(result)
```

```
predicted = result.argmax(axis=-1)
print(predicted)
```

-----

[수요일]

```
import seaborn as sns
```

```
df = sns.load_dataset('titanic')
```

```
print(df)
```

```
sns.countplot(data = df, x = 'class', hue = 'alive')
```

```
# 문장 분석
```

```
!pip install konlpy
```

```
from konlpy.tag import Okt
```

```
text = "우리 가족은 엄마 아빠 나 그리고 동생이다."
```

```
okt = Okt()
```

```
result = okt.pos(text, norm=True, stem=True)
```

```
print(result)
```

```
#워드카운트
```

```
from sklearn.feature_extraction.text import CountVectorizer
```

```
corpus = ['송아지 송아지 얼룩 송아지 엄마 소도 얼룩 소 엄마 닭았네']
```

```
vector = CountVectorizer()
```

```
print(vector.fit_transform(corpus).toarray()) # 코퍼스 분석
print(vector.vocabulary_)
```

```
-----

# 워드벡터
```

```
import nltk
from nltk.corpus import movie_reviews
from gensim.models.word2vec import Word2Vec

nltk.download('movie_reviews')
nltk.download('punkt')

sentences = [list(s) for s in movie_reviews.sents()]
model = Word2Vec(sentences)

print(model.wv.similarity('actor', 'actress'))
print(model.wv.similarity('he', 'she'))
print(model.wv.most_similar(positive=['actress', 'he'], negative=['she'])[0])
```

```
-----

def ngram(s, num):
    res = []
    slen = len(s) - num + 1
    for i in range(slen):
        ss = s[i:i+num]
        res.append(ss)
    return res

def diff_ngram(sa, sb, num):
    a = ngram(sa, num)
    b = ngram(sb, num)
    r = []
    cnt = 0
    for i in a:
        for j in b:
            if i == j:
```

```
        cnt += 1
        r.append(i)
    return cnt / len(a), r
a = "오늘 강남에서 맛있는 스파게티를 먹었다."
b = "강남에서 먹었던 오늘의 스파게티는 맛있었다."
# 2-gram
r2, word2 = diff_ngram(a, b, 2)
print("2-gram:", r2, word2)
# 3-gram
r3, word3 = diff_ngram(a, b, 3)
print("3-gram:", r3, word3)
```

-----



## # 텍스트 분류

```
from sklearn.naive_bayes import BernoulliNB, GaussianNB, MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer
```

## # 훈련 데이터 (텍스트)

```
corpus = ["광고 쇼핑물 가격 선물 세일",
          "투자 수익 소식 등급",
          "쿠폰 선물 무료 익일 배송",
          "백화점 상품 파격 세일",
          "프로젝트 보고서 지급",
          "회의 일정 변경"]
```

## # 훈련라벨 (0: 일반, 1: 스팸)

```
label = [1,1,1,1,0,0]
```

```
vector = CountVectorizer()
train = vector.fit_transform(corpus).toarray()
```

## # 나이브 베이즈 분류 모델 생성

```
model = BernoulliNB()
model.fit(train, label) # 학습
```

```
data = ["백화점 세일 무료 배송 알찬"]
test = vector.transform(data).toarray()
```

```
result = model.predict(test)
print(result)
```

virtualenv \_envKosa

○ bs.py

```
#pip install beautifulsoup4
```

# <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

# <http://www.w3school.com/>

```
from bs4 import BeautifulSoup
```

```
html_doc = """
```

```
<html>
```

```
<head><title>The Dormouse's story</title></head>
```

```
<body>
```

```
<p class="title"><b>The Dormouse's story</b></p>
```

```
<p class="story">Once upon a time there were three little sisters
```

```
<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,
```

```
<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and
```

```
<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;
```

```
and they lived at the bottom of a well.</p>
```

```
<p class="story">...</p>
```

```
</body>
```

```
</html>
```

```
"""
```

```
soup = BeautifulSoup(html_doc, 'html.parser')
```

```
print(soup.prettify())
```

```
print(soup.title)
```

```
print(soup.title.parent.name)
```

```
print(soup.p)
```

```
print(soup.a)
```

```
for link in soup.find_all('a'):
```

```
    print(link.get('href'))
```

```
-----
```

```
○ request_bs.py
```

```
-----
```

```
# pip install requests
```

```
import requests
from bs4 import BeautifulSoup

res =
requests.get('https://search.naver.com/search.naver?where=nexearch&sm=top_hy&fbm=1&ie=utf8&quer
y=%EA%B0%9C%EB%B0%9C%EC%9E%90')
print(res.content)

soup = BeautifulSoup(res.content, 'html.parser')

# Title 확인
title = soup.find('title')
print(title.get_text())

# 나무위키 찾기
# 선택자를 사용하여 HTML 요소를 가져옵니다.

element = soup.select_one('#main_pack section.sc_new.sp_ntotal._sp_ntotal._prs_sit_4po._fe_root_sit_4po
> div > ul > li > div.total_wrap.api_ani_send > div.total_group > div > div > div > a')
# 요소의 텍스트를 출력합니다.
print(element.getText())
# element = soup.select_one('//*[@id="main_pack"]/section[1]/div/ul/li/div[1]/div[3]/div/div/div/a')
element = soup.select_one('#main_pack section:nth-of-type(1) div ul li div:nth-of-type(1) div:nth-of-
type(3) div div div a')
# 요소의 텍스트를 출력합니다.
print(element.getText())

total_tit = soup.find_all( class_ = 'total_tit')
for one in total_tit:
    print(one.getText())

-----

○ WebCapture.py
-----

# pip install selenium
# https://chromedriver.chromium.org/
```

```
import time
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
```

```
options = Options()
options.add_argument('--headless')
```

```
driver = webdriver.Chrome(options=options)
```

```
url = "http://www.naver.com/"
driver.get(url)
driver.implicitly_wait(2)
```

```
driver.get_screenshot_as_file("webCapture.png")
```

```
driver.quit()
```

-----

○ webElement.py

-----

```
# pip install selenium
```

```
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.options import Options
```

```
options = Options()
options.add_argument("user-agent=Mozilla/5.0")
options.add_argument("--start-maximized")
```

```
driver = webdriver.Chrome(options=options)
```

```
url = "https://www.naver.com/"
```

```
driver.get(url)
```

```
time.sleep(2)

inputElement = driver.find_element(By.NAME,"query")
inputElement.clear()
time.sleep(1)
inputElement.send_keys("개발자")
time.sleep(1)
inputElement.submit()
time.sleep(1)

results = driver.find_elements(By.CLASS_NAME,'total_tit')
time.sleep(1)
for one in results:
    print(one.text)
time.sleep(1)

results[0].click()
time.sleep(2)

# WebDriver를 종료합니다. (브라우저 닫기)
driver.quit()
```

-----

[영상 처리에 대한 기본]

\* RGB : R+ G+ B (3바이트, 24비트의 압축되지 않은 이미지) => 모니터(화면) 출력 !!!

\* 정지영상 압축

- YCbCr (YUV) : Y(휘도) + CbCr(색차)

8     8     8  
=> 8     4     4  
=> 8     2     2 ==> JPEG

\* 동영상 압축 (정지영상의 연속, JPEG)

- 기준 이미지에서 변경된 요소들을 중심으로 압축한다.

<http://naver.me/FXKss3eD>

○ exImage.py

-----

```
import cv2
import numpy as np

imageFile = './data/lena.jpg'
img = cv2.imread(imageFile)
cv2.imshow('Lena color',img)

cv2.imwrite('./data/Lena.bmp', img)
cv2.imwrite('./data/Lena.png', img)
cv2.imwrite('./data/Lena2.jpg', img, [cv2.IMWRITE_JPEG_QUALITY, 90])

imageFile = './data/Lena.png'
img = cv2.imread(imageFile, cv2.IMREAD_GRAYSCALE)
cv2.imshow('Lena gray',img)

cv2.waitKey()
cv2.destroyAllWindows()
-----

pip install opencv-python
```

○ exImageDraw.py

```
-----
import cv2
import numpy as np

imageFile = './data/lena.jpg'
img = cv2.imread(imageFile)
cv2.imshow('Lena Original',img)

# 선과 사각형 그리기
pt1 = 100, 100
pt2 = 300, 300
cv2.rectangle(img, pt1, pt2, (0, 255, 0), 2)
```

---

```
cv2.line(img, (0, 0), (400, 0), (255, 0, 0), 10)
cv2.line(img, (0, 0), (0, 400), (0,0,255), 10)

# 특정 영역의 색상 변경
img[10:50, 10:50] = 0

# 텍스트 출력
font = cv2.FONT_HERSHEY_SIMPLEX
text = "Hello"
cv2.putText(img, text, (300, 300), font, 1, (255, 255, 255), 4)

cv2.imshow('Lena Edited',img)
cv2.imwrite('./data/Lena2.jpg', img, [cv2.IMWRITE_JPEG_QUALITY, 2])

imageFile = './data/Lena2.jpg'
img = cv2.imread(imageFile)
cv2.imshow('Lena 2nd',img)

cv2.waitKey()
cv2.destroyAllWindows()
-----

○ harris.py
-----

# 필요한 Module(Library) 임포트
import numpy as np
import cv2 as cv

# 도로 이미지 읽어서 표시
cimg = cv.imread('./data/road.jpg')
cv.imshow('Roads - original', cimg)

# 도로이미지 이진화, 에지 추출 후 표시
img = cv.cvtColor(cimg, cv.COLOR_BGR2GRAY)
_, img = cv.threshold(img, 170, 255, cv.THRESH_BINARY)
img = cv.Canny(img, 50, 200, None, 3)
cv.imshow('Image - binary & edge', img)

# Harris를 이용한 모서리 검출
k = 0.1
```

```
T = 0.01
R = cv.cornerHarris(img, 2, 3, k)

# 원본 이미지 + 모서리 표시
cimg[R > T * R.max()] = [0, 0, 255]
cv.imshow('Harris Coners', cimg)

# 사용자 입력키 대기 및 종료
cv.waitKey()
cv.destroyAllWindows()

-----

○ hough.py
-----

# 필요한 Module(Library) импорт
import numpy as np
import cv2 as cv

# 도로 이미지 읽어서 표시
cimg = cv.imread('./data/Roads.png')
cv.imshow('Roads - original', cimg)

# 도로이미지 이진화, 에지 추출 후 표시
img = cv.cvtColor(cimg, cv.COLOR_BGR2GRAY)
_, img = cv.threshold(img, 100, 255, cv.THRESH_BINARY)
img = cv.Canny(img, 50, 200, None, 3)
cv.imshow('Image - binary & edge', img)

# Hough를 이용한 직선 검출
lines = cv.HoughLines(img, 1, np.pi / 180, 150)

# 추출한 선을 원본 이미지 위에 표시
if lines is not None:
    for i in range(0, len(lines)):
        rho = lines[i][0][0]
        theta = lines[i][0][1]
        a = np.cos(theta)
        b = np.sin(theta)
        x = a * rho
        y = b * rho
```



```
pt1 = (int(x + 1000 * (-b)), int(y + 1000 * (a)))
pt2 = (int(x - 1000 * (-b)), int(y - 1000 * (a)))
cv.line(cimg, pt1, pt2, (0,0,255), 2)

# 원본 이미지 + Line 표시
cv.imshow('Hough Lines', cimg)

# 사용자 입력키 대기 및 종료
cv.waitKey()
cv.destroyAllWindows()
```

-----

○ exYouTube.py

-----

```
import cv2, pafy

url = 'https://www.youtube.com/watch?v=u_Q7Dkl7Alk'
video = pafy.new(url)
print('title = ', video.title)
print('video.rating = ', video.rating)
print('video.duration = ', video.duration)
best = video.getbest()
print('best.resolution', best.resolution)

cap=cv2.VideoCapture(best.url)
while(True):
    retval, frame = cap.read()
    if not retval:
        break
    cv2.imshow('frame',frame)

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    edges = cv2.Canny(gray,100,200)
    cv2.imshow('edges',edges)

    key = cv2.waitKey(25)
    if key == 27: # Esc
        break
```

```
cv2.destroyAllWindows()
```

```
-----
```

```
pip install pafy pygame
```

```
pip install youtube-dl==2020.12.2
```

○ testMatching.py

```
-----
```

```
import cv2
```

```
# 사람 이미지
```

```
template = cv2.imread('./data/wili.png')
```

```
th, tw = template.shape[:2]
```

```
cv2.imshow('template', template)
```

```
# step1) webCAM 이미지 준비
```

```
image = cv2.imread('./data/original.jpg')
```

```
# step2) 이미지 특징 매칭
```

```
result = cv2.matchTemplate(image, template, cv2.TM_CCOEFF_NORMED)
```

```
# step3) 매칭 좌표를 이미지에 그리기
```

```
minVal, maxVal, minLoc, maxLoc = cv2.minMaxLoc(result)
```

```
top_left = maxLoc
```

```
match_val = maxVal
```

```
bottom_right = (top_left[0] + tw, top_left[1] + th)
```

```
cv2.rectangle(image, top_left, bottom_right, (0, 0, 255), 2)
```

```
cv2.imshow('Result', image)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

```
-----
```

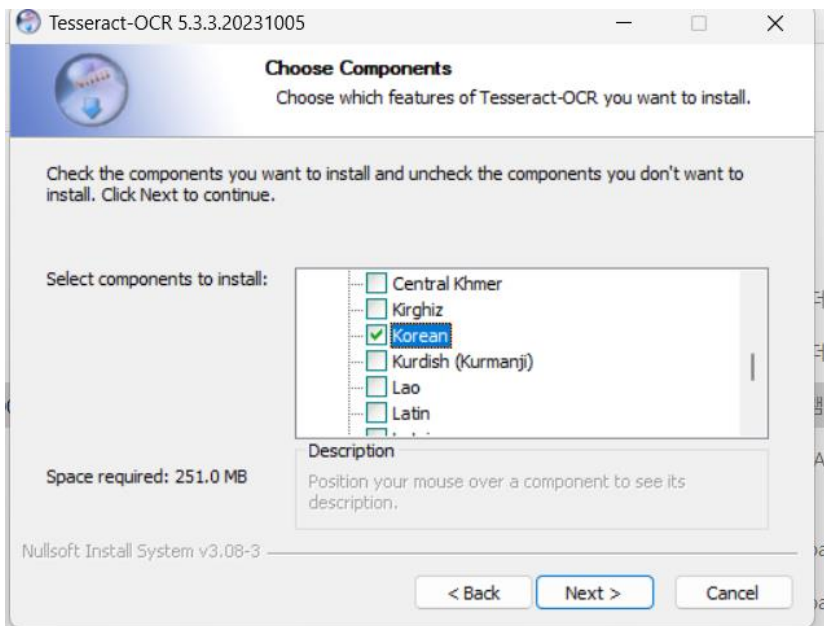
○ 설치가 필요한 내용

[OCR]

\* 윈도우 OCR 프로그램

- [Home · UB-Mannheim/tesseract Wiki \(github.com\)](https://github.com/UB-Mannheim/tesseract/wiki)

( <https://github.com/UB-Mannheim/tesseract/wiki> )



○ testOCR.py

```
-----
# pip install speechrecognition
# pip install gTTS
# pip install playsound==1.2.2
# pip install pillow
# pip install pytesseract
```

```
from PIL import Image
import pytesseract
import speech_recognition as sr
from gtts import gTTS
import playsound
```

```
filename = '.WdataWsound.mp3'
def speak(text):
    tts = gTTS(text=text, lang='ko')
    tts.save(filename)
    playsound.playsound(filename)
```

```
pytesseract.pytesseract.tesseract_cmd = 'C:WWProgram FilesWWTesseract-OCRWwtesseract'
```

```
myConfig = ('-l kor --oem 3 --psm 4')
str = pytesseract.image_to_string(Image.open('.WdataWsample.jpg'), config=myConfig)
```

```
print(str)
speak(str)
```

-----

#### [소프트웨어 공학]

\* 재사용 : 좋은 품질의 모듈(재사용의 단위)를 재사용하는 것이 중요하다.

내가 모듈을 만들었다. 그 모듈의 재사용하는데 좋을까? 품질일까?

=> 품질의 핵심 개념은 '기능(약속, 인터페이스)는 유지하면서 변화에는 유연해야 한다.'

1) 결합도 : 2개 이상의 모듈이 서로 느슨하게 연결되어야 한다.

[덧셈]            [계산기]

add(a, b)

add() 전역변수에 a,b 설정

add() 파일에서 a, b기록

add() add.py 안에 a, b 변수를 네가 직접 설정.

2) 응집도 : 1개의 모듈 안에서 필요한 기능만 잘 모여 있어야 한다.

\* 재사용 측면에서 개발 방법론의 발전 동향

1) 구조적 개발방법론 : 함수를 사용한다.

2) 객체지향 개발방법론 : 클래스 (함수, 데이터)

3) CBD (Component Based Development)

4) Web Service (표준 인터넷 기술을 이용한 서비스-기능 제공)

-----

o yolo\_object\_video.py

-----

# 출처 : <https://pysource.com/2019/06/27/yolo-object-detection-using-opencv-with-python/>

# 라이선스 : 학습으로만 사용.

```
import cv2
```

```
import numpy as np
```

```
# Load Yolo
```

```
net = cv2.dnn.readNet("./yolo_object_detection/yolov3.weights", "./yolo_object_detection/yolov3.cfg")
```

```
classes = []
```

```
with open("./yolo_object_detection/coco.names", "r") as f:
```

```
    classes = [line.strip() for line in f.readlines()]
```

```
layer_names = net.getLayerNames()
```

```
output_layers = [layer_names[i - 1] for i in net.getUnconnectedOutLayers()]
```

---

```
colors = np.random.uniform(0, 255, size=(len(classes), 3))

# Loading image
cap=cv2.VideoCapture(0)
while(True):
    retval, img = cap.read()
    if not retval:
        break
    #img = cv2.imread("./data/girl.png")
    img = cv2.resize(img, None, fx=0.4, fy=0.4)
    height, width, channels = img.shape

    # Detecting objects
    blob = cv2.dnn.blobFromImage(img, 0.00392, (416, 416), (0, 0, 0), True, crop=False)

    net.setInput(blob)
    outs = net.forward(output_layers)

    # Showing informations on the screen
    class_ids = []
    confidences = []
    boxes = []
    for out in outs:
        for detection in out:
            scores = detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > 0.5:
                # Object detected
                center_x = int(detection[0] * width)
                center_y = int(detection[1] * height)
                w = int(detection[2] * width)
                h = int(detection[3] * height)

                # Rectangle coordinates
                x = int(center_x - w / 2)
                y = int(center_y - h / 2)

                boxes.append([x, y, w, h])
```

```
confidences.append(float(confidence))
class_ids.append(class_id)
```

```
indexes = cv2.dnn.NMSBoxes(bboxes, confidences, 0.5, 0.4)
print(indexes)
font = cv2.FONT_HERSHEY_PLAIN
for i in range(len(bboxes)):
    if i in indexes:
        x, y, w, h = bboxes[i]
        label = str(classes[class_ids[i]])
        color = colors[class_ids[i]]
        cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
        cv2.putText(img, label, (x, y + 30), font, 2, color, 2)
```

```
cv2.imshow("Image", img)
key = cv2.waitKey(25)
if key == 27: # Esc
    break
```

```
cv2.destroyAllWindows()
```

\* <https://pabi.smartlearn.io/>

\* 소프트웨어를 테스트하는 다양한 관점  
(공통 용어)

- 정적 (~리뷰) vs 동적 (실행)
- 화이트 (내부구조를 참고) vs 블랙 (입/출력, 인터페이스)
- 명세기반 vs 경험

○ 성숙도 : 일을 잘하는 수준

- CMMI 5단계의 성숙도
  - 1) 초기 : 그때 그때 마다 다르다.
  - 2) 반복 : 과거의 경험이 쌓이면 그 분야에 대해서는 어느정도 잘 하는 단계
  - 3) 정의 : 일하는 방법이 (전사적으로) 통일되어 있어서 기대하는 수준에 따라 일할 수 있는 단계.
  - 4) 정량 : 일하는 결과, 성과를 계속 수치로 기록, 관리.
  - 5) 최적화 : 측정한 수치를 기반으로 계속 성장.

- 의사소통
  - 누구한테
  - 어떤 내용을
  - 어떤 수단으로
  - 얼마나 자주

○ 범위 관리 : WBS (Work Breakdown System/Structure)

- 프로젝트 (과업, 일, Task, 활동)을 수행하기 위해 필요한 일을 Top-Down (구조적, 트리형식으로) 세분화하여 정의하고 명세(문선화) 한 것.
- 특징
  - 1) 각각의 Task(Work Package)를 완성하면 전체가 완성
  - 2) (상황에 따라 다르나) 2주 정도의 일정 단위로 구성
  - 3) 해당 Task (Work Package)가 끝났다는 것을 확인할 수 있는 결과물.
  - 4) 담당자.

○ 위험관리

- 위험 : 우리 프로젝트에 영향을 미칠 수 있는 발생 가능한 사항(일)
- 관리:
  - 1) 위험 식별
  - 2) 위험 평가(비교) : 발생가능성 x 영향도
  - 3) 위험 대책 : 회피, 완화, 수용, 전가.
  - 4) 통제

<http://naver.me/FXKss3eD>

☐ 내려받기    네이버 MYBOX에 저장    4개 선택    Chris님이 공유한 폴더 ⓘ

종류	이름↓	크기	수정된 날짜	올린 날짜
<input checked="" type="checkbox"/>	WinSpy.zip	1.0MB	2020. 09. 03. 오전 09:14	2023. 05. 25. 오후 03:34
<input type="checkbox"/>	usbdevview-x64.zip	114.5KB	2020. 09. 01. 오후 12:58	2023. 05. 25. 오후 03:34
<input checked="" type="checkbox"/>	SysinternalsSuite.zip	25.2MB	2021. 08. 10. 오후 02:49	2023. 05. 25. 오후 03:34
<input type="checkbox"/>	SikuliX.zip	147.3MB	2020. 06. 21. 오후 12:27	2023. 02. 22. 오후 02:54
<input checked="" type="checkbox"/>	PView.exe	66.0KB	2021. 07. 14. 오전 09:15	2023. 05. 25. 오후 03:34
<input type="checkbox"/>	openjdk-11+28_windows-x64_bin.zip	178.7MB	2022. 06. 29. 오전 05:41	2023. 02. 22. 오후 02:54
<input checked="" type="checkbox"/>	HxDSetup.exe	3.1MB	2021. 07. 14. 오전 09:16	2023. 05. 25. 오후 03:34



## KOSA\_ITPro > VM\_Data

☒ 내려받기    네이버 MYBOX에 저장    3개 선택 | Chris님이 공유한 폴더 ⓘ

종류	이름↓	크기	수정일
<input checked="" type="checkbox"/>	 VC_redist.x64.exe	24.1MB	2022. 11. 01. 오전 0
<input checked="" type="checkbox"/>	 Kali-2019-2-amd64_toor.ova	4.0GB	2021. 09. 05. 오전 0
<input checked="" type="checkbox"/>	 CentOS7_team1234.ova	993.2MB	2020. 11. 22. 오전 0

## ○ 윈도우 프로그램의 첫번째 특징

==> 모든 GUI 응용 프로그램은 윈도우들의 모음으로 구성되고,  
각각의 윈도우는 메시지에 대해서 반응한다

○ NTFS : ADS (Alternative Data Stream)

○ hack.vbs

```
-----
Set WshShell = WScript.CreateObject("WScript.Shell")
WshShell.Run "notepad"
WScript.Sleep 100
WshShell.AppActivate "Notepad"
WScript.Sleep 200
WshShell.SendKeys "T"
WScript.Sleep 200
WshShell.SendKeys "h"
WScript.Sleep 200
WshShell.SendKeys "i"
WScript.Sleep 200
WshShell.SendKeys "s"
WScript.Sleep 200
WshShell.SendKeys " "
WScript.Sleep 200
```

```
WshShell.SendKeys "i"
WScript.Sleep 200
WshShell.SendKeys "s"
WScript.Sleep 200
WshShell.SendKeys " "
WScript.Sleep 200
WshShell.SendKeys "h"
WScript.Sleep 200
WshShell.SendKeys "a"
WScript.Sleep 200
WshShell.SendKeys "c"
WScript.Sleep 200
WshShell.SendKeys "k"
WScript.Sleep 200
WshShell.SendKeys "e"
WScript.Sleep 200
WshShell.SendKeys "d"
WScript.Sleep 200
WshShell.SendKeys "!"
WScript.Sleep 200
WshShell.SendKeys "..."
WScript.Sleep 200
```

-----

```
type c:\Windows\System32\calc.exe > test.txt:calc.exe
```

```
wmic process call create "c:\Temp\test.txt:calc.exe"
```

<https://learn.microsoft.com/ko-kr/sysinternals/>

<http://www.nirsoft.net/>

- 오픈소스 라이선스 : 출처 기술.
- AGPL

- GPL : GPL 오픈소스를 가져다가 수정 (GPL), 가져다고 내 소스코드에 함께 사용 (내 것도 GPL)
- LGPL : Lesser(Library), LGPL 오픈소스를 수정 (LGPL), 가져다가 단순히 링크 (OK)
- BSD, MIT 등 : 출처만 밝히면된다.

○ 소프트웨어

- 상용
- 쉼어
- 프리

○ 윈도우 서비스의 특징

WIN+R : services.msc

○ 윈도우 레지스트리

WIN+R : regedit

○ 리눅스 종료 명령어

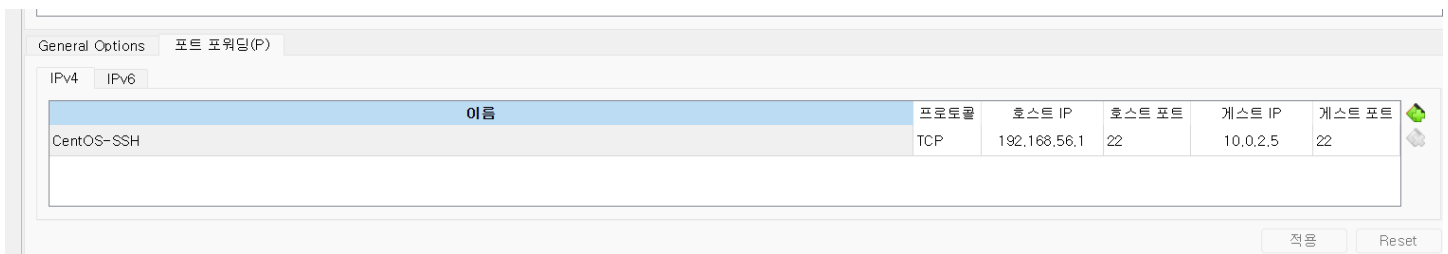
: shutdown -h now

[금요일]

- \* ~ 11:30 - 강의
- \* ~ 13:00 (오후 1시) : 점심
- \* ~ 14:00 (오후 2시) : 개인 스터디
- \* 평가: 2시~3시40 (100분) : 10문제 (서술식)
- \* 4시~ : 오후 강의
- \* 4시~5시50분 : 학습.

-----

\* VirtualBox => CentOS 부팅 (로그인)



\* ssh 접속 명령어 : ssh root@192.168.56.1

putty 프로그램 : <https://putty.softonic.kr/>

=> 다운로드 / (기본옵션) 설치

\* 10시 10분~ 시작하겠습니다.

○ 리눅스 셸 스크립트

- test.sh

-----

```
#!/bin/bash
# Shell script example
function helloWorld
{
    echo "Hello World"
}
```

```
numParam=$#

if [ $numParam -eq 1 ]
then
    echo "$1"
elif [ $numParam -eq 2 ]
then
    echo "$1"
    echo "$2"
else
    echo "No params"
fi

for i in {1..10}
do
    echo "$i processing"
done

MSG="Shell Script Example"
echo $MSG

helloWorld
```

-----

\* test.sh를 어떻게 실행하지.

```
# source test.sh
# . test.sh
# bash test.sh
# chmod +x test.sh
```

-----

웹서버를 설치하고 실행하고 싶다. (httpd )

설치: yum install httpd (우분투 : apt-get 혹은 apt install apache2)

실행: systemctl start httpd (실행)  
 systemctl enable httpd (자동실행)  
 systemctl status httpd (상태)

방화벽설정

```
firewall-cmd --permanent --add-port=80/tcp
firewall-cmd --reload
```

\* [모바일앱\(안드로이드\) 보안 취약점 점검](#)