



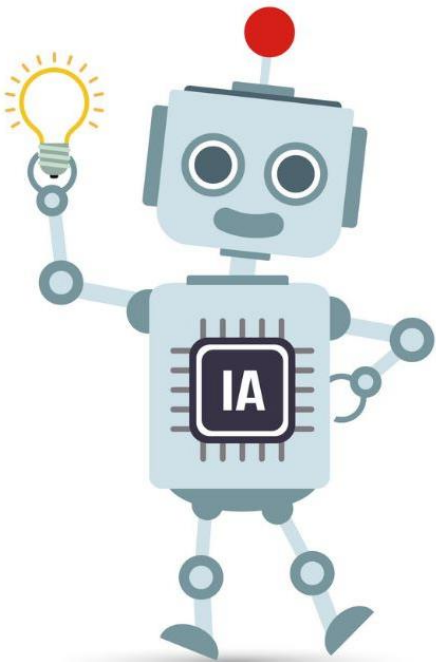
공개강의 제목

2019 / 06 / 25

정 윤 주

목 차

• SW중심대학

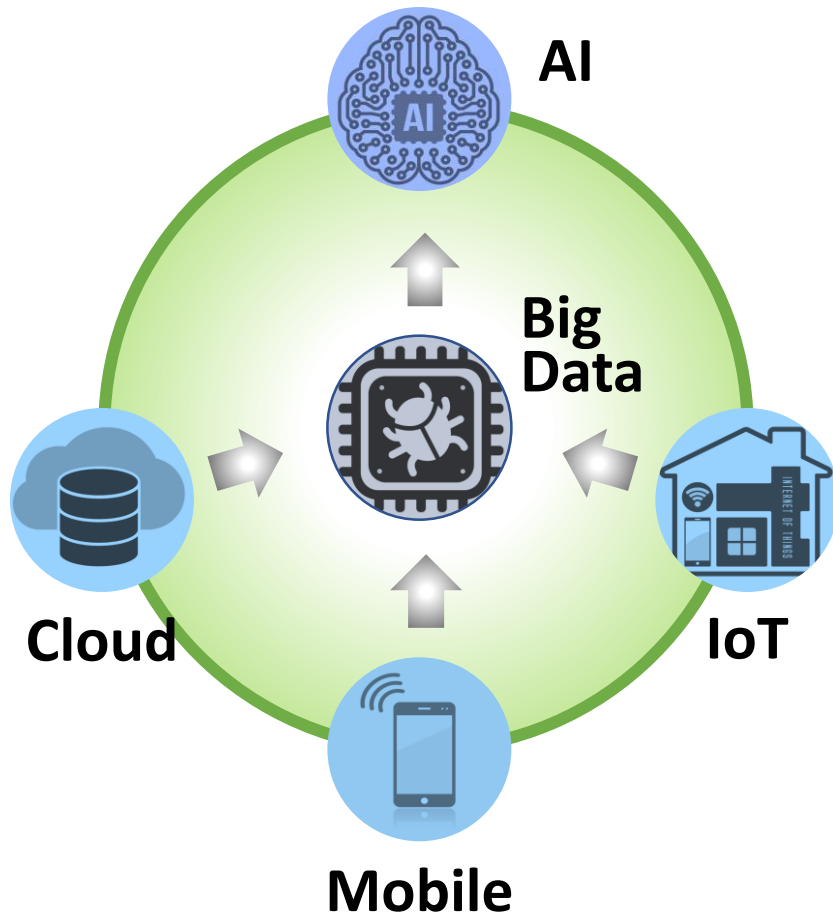


안동대학교 SW중심대학 교육의 설계

전공자 SW교육 설계

• SW중심대학

4차 산업혁명 이후 글로벌 소프트웨어 개발 핵심 인재의 양성



AI	IoT	Data Science	Mobile	Cloud
Python				
Java / Javascript				
C / C++				
R		R		클라우드 컴퓨팅언어 • SaaS PaaS IaaS
		SQL	앱인벤터	
		SAS	Objective-C	
			Swift	

Open Source System(OSS)를 통한 협업학습 및
분산 버전 관리

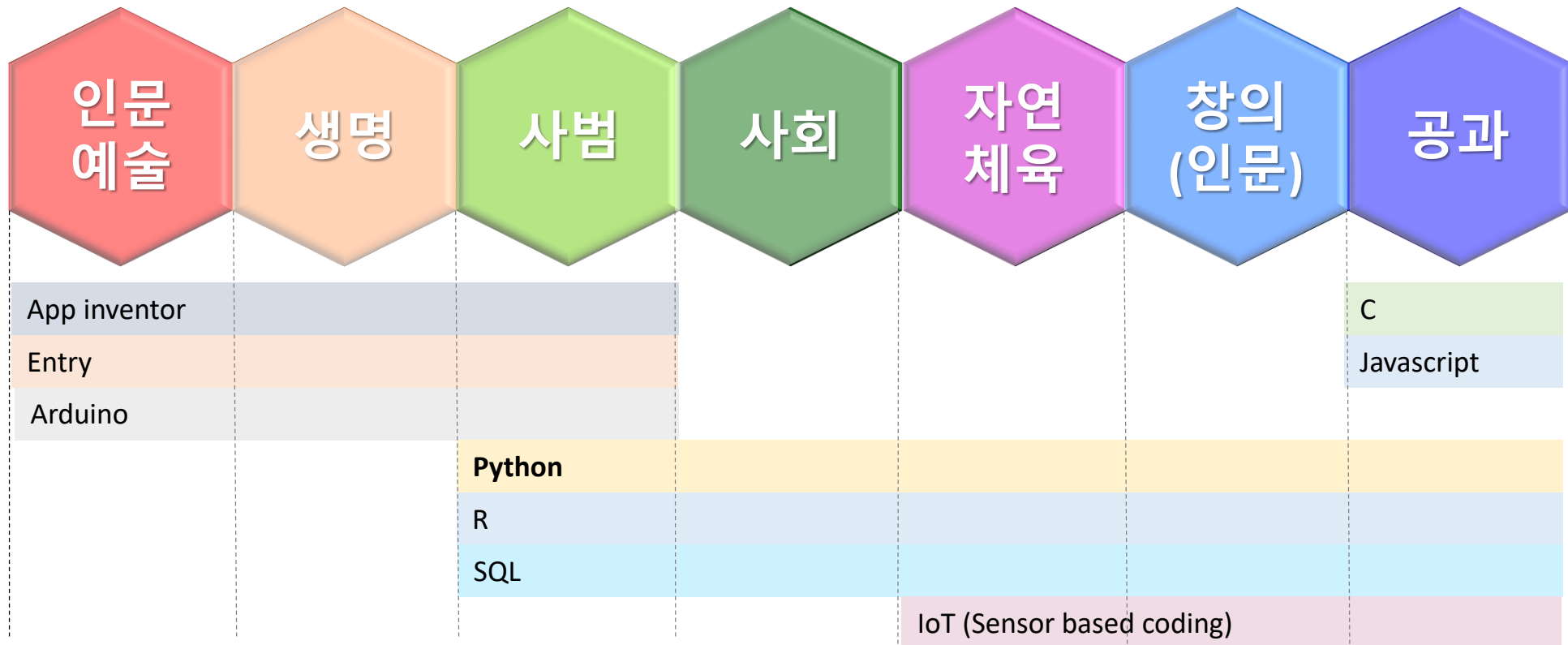


비전공자 SW교육 설계

• SW중심대학





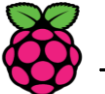


















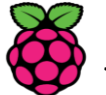




전공별 특성에 맞는 다양한 프로그래밍언어를 교육

: 창의적 사고와 코딩을 통하여 "창의적·실제적" 문제 해결 역량 강화를 목표



지역민들을 위한 SW 교육 설계

· SW중심대학

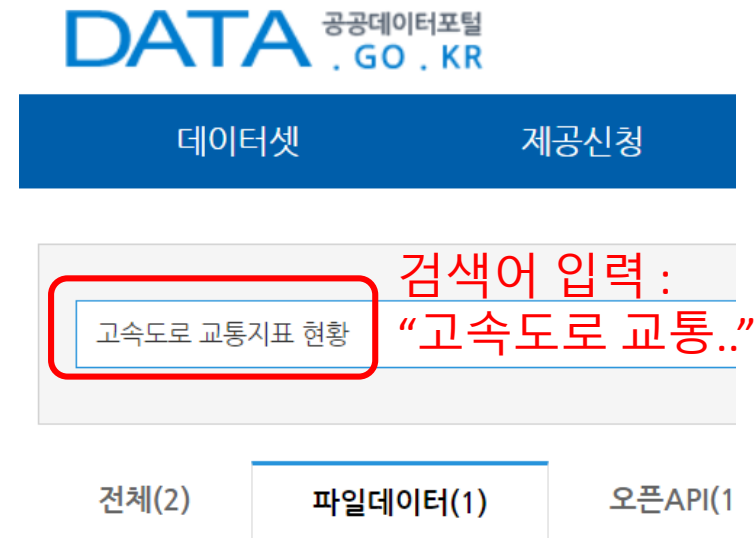
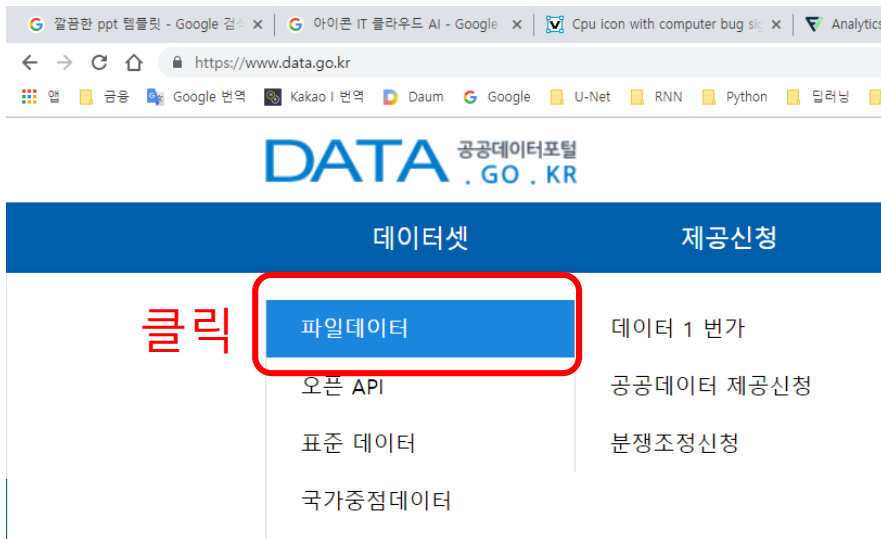
아동·청소년		경력단절 여성	교사 및 교육종사자	기업인·농업인
초 중 고	     	      	      	        JavaScript



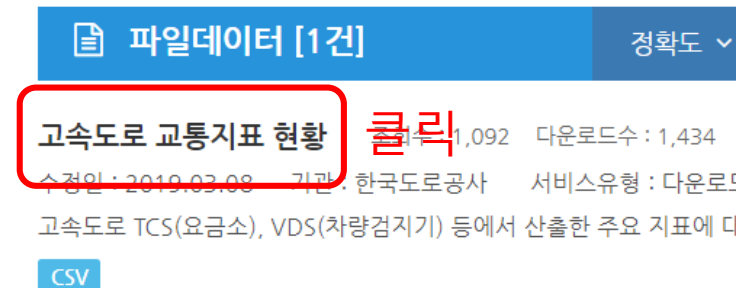
파이썬을 이용한 데이터 분석

SW중심대학

1. 공공 데이터 가져오기 <https://www.data.go.kr/>



파일데이터 1건을 찾았습니다.



파이썬을 이용한 데이터 분석

• SW중심대학

1. 공공 데이터 가져오기

☐ CSV 고속도로_상습정체구간(2015년11월...

↓ 멀티다운로드

🔍 상세정보

! 오류신고



☐ CSV 고속도로_노선별 월 변동계수(2015년...

↓ 멀티다운로드

🔍 상세정보

! 오류신고



☐ CSV 고속도로_노선별 요일 변동계수(2015...

↓ 다운로드

🔍 상세정보

! 오류신고



☐ CSV 고속도로_지정차로제지정현황(2015년)

↓ 다운로드

🔍 상세정보

! 오류신고



☐ CSV 고속도로_교통사고통계(2015년11월...

↓ 다운로드

🔍 상세정보

! 오류신고



☐ CSV 고속도로 시공간 분산지수(2016년)

↓ 다운로드

🔍 상세정보

! 오류신고



→ “고속도로교통사고현황.csv” 라는 이름으로 저장



파이썬을 이용한 데이터 분석

2. 데이터 파일 읽기

```
import pandas as pd
import matplotlib.pyplot as plt

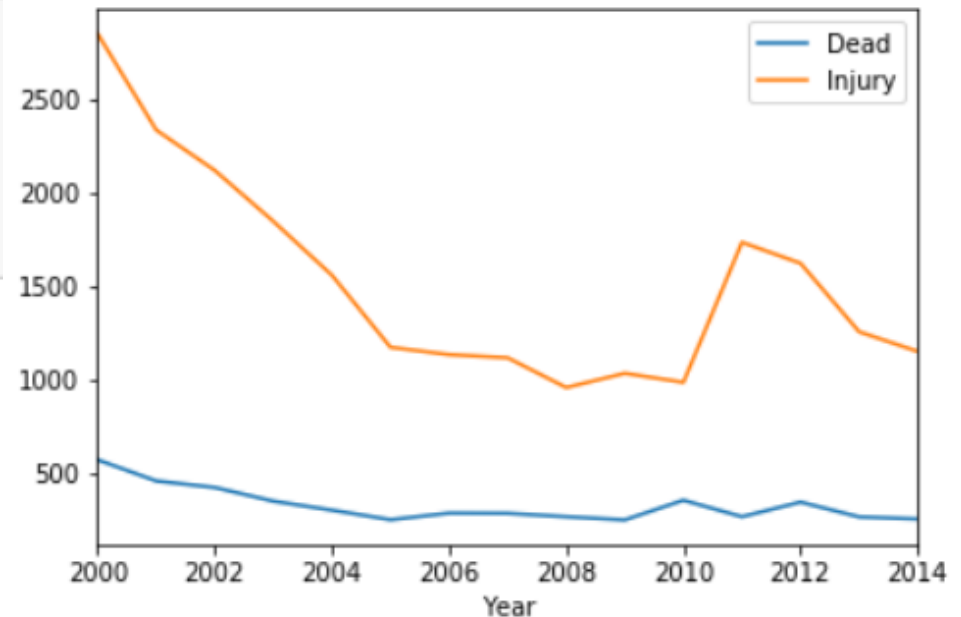
#df = pd.read_csv('고속도로교통사고현황.csv')
df = pd.read_csv('고속도로교통사고현황.csv', encoding='CP949')
df
```

	연도	사고	사망	부상
0	2000	3910	569	2845
1	2001	3638	456	2331
2	2002	3957	421	2115
3	2003	3585	348	1843
4	2004	3242	300	1555
5	2005	2880	249	1170
6	2006	2583	284	1131
7	2007	2550	283	1114
8	2008	2449	265	955
9	2009	2374	248	1031
10	2010	2368	353	983
11	2011	2640	265	1731
12	2012	2600	343	1619
13	2013	2496	264	1253
14	2014	2395	253	1148

파이썬을 이용한 데이터 분석

3. 데이터의 일부를 이용하여 시각화하기

```
df1=df.loc[:,['연도','사망','부상'] ]  
ax=df1.plot(kind='line', x='연도')  
ax.set_xlabel('Year')  
ax.legend(['Dead', 'Injury'])  
plt.show()
```



파이썬을 이용한 머신러닝

1. 머신러닝을 이용한 교통 사고수 예측

```
*test.py - C:/JYJ/ML_test01/test.py (3.6.5)*
File Edit Format Run Options Window Help
import tensorflow as tf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('고속도로교통사고현황.csv', encoding='CP949')
list1 = df.values.tolist()
ar = np.array(list1)
xData = ar[:, 0]-2000
yData = ar[:, 1]

W = tf.Variable(tf.random_uniform([1], -100, 100))
b = tf.Variable(tf.random_uniform([1], -100, 100))
X = tf.placeholder(tf.float32)
Y = tf.placeholder(tf.float32)
H = W * X + b
cost = tf.reduce_mean(tf.square(H-Y))
a = tf.Variable(0.01)
optimizer = tf.train.GradientDescentOptimizer(a)
train = optimizer.minimize(cost)
```

Ln: 25 Col: 0

파이썬을 이용한 머신러닝

1. 머신러닝을 이용한 교통 사고수 예측

```
*test.py - C:/JVJ/ML_test01/test.py (3.6.5)*
File Edit Format Run Options Window Help

init = tf.global_variables_initializer()
sess = tf.Session()
sess.run(init)

for i in range(5001):
    sess.run(train, feed_dict={X:xData, Y:yData})
    if i % 500 == 0:
        print(i, sess.run(cost, feed_dict={X:xData, Y:yData}), sess.run(W), sess.run(b))

print(sess.run(H, feed_dict={X:[16]}))

hY = sess.run([(W * x + b) for x in xData])

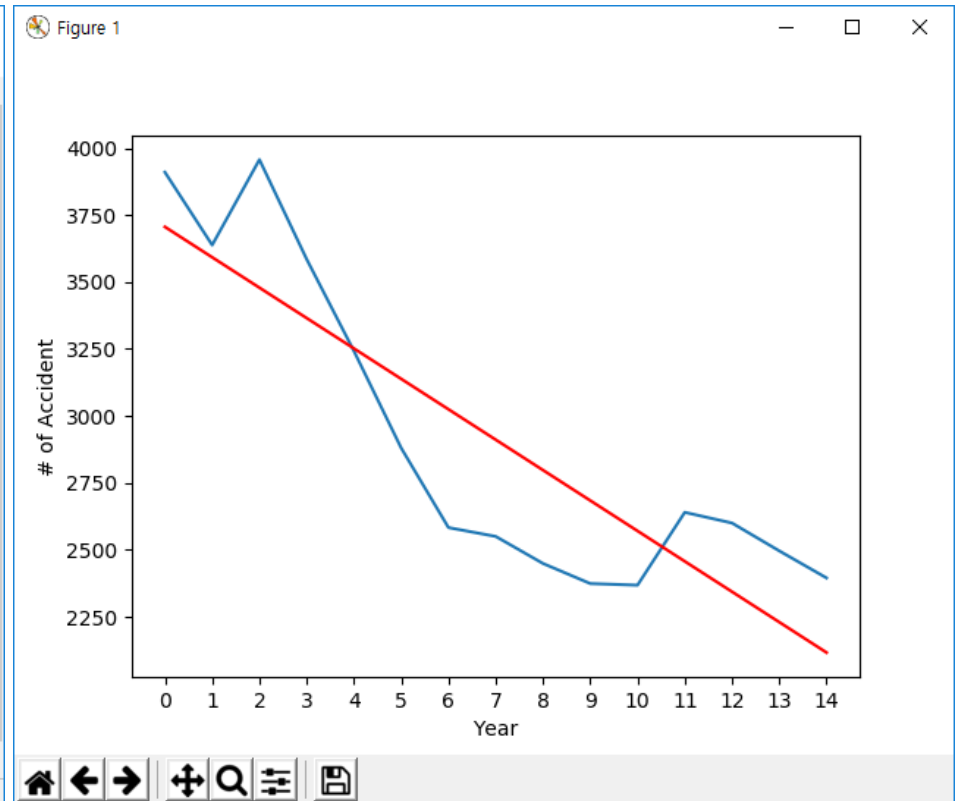
plt.plot(xData, yData)
plt.plot(xData, hY, 'red')
plt.show()
```

Ln: 41 Col: 0

파이썬을 이용한 머신러닝

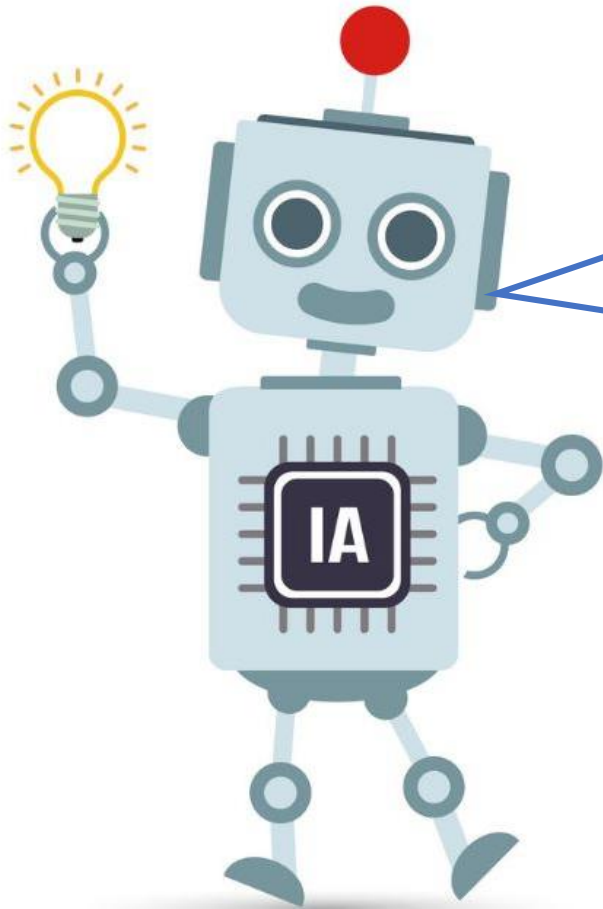
2. 실험 결과와 시각화

```
*Python 3.6.5 Shell*
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/JYJ/ML_test01
/test.py =====
0 4239504.5 [340.8426] [-26.399437]
500 97475.76 [-88.37711] [3463.6643]
1000 81358.12 [-111.90616] [3690.1946]
1500 81290.484 [-113.4302] [3704.8674]
2000 81290.234 [-113.52895] [3705.818]
2500 81290.22 [-113.53346] [3705.8616]
3000 81290.22 [-113.53346] [3705.8616]
3500 81290.22 [-113.53346] [3705.8616]
4000 81290.22 [-113.53346] [3705.8616]
4500 81290.22 [-113.53346] [3705.8616]
5000 81290.22 [-113.53346] [3705.8616]
[1889.3262]
```



다음 시간에 할 일

• SW중심대학



다음 시간 목표 :

머신러닝을 이용한 연도별 교통사고수
예측 프로그램을 분석하여 완벽히
이해하기

다음 시간에 만나요~~~~~

