

2022년도 1학기 개인별 교과목 포트폴리오

학부 :	컴퓨터공학과
학과 :	컴퓨터정보공학과
학번 :	20211012
성명 :	이용규



동양미래대학교

목차

- 목차
- 목차
- 목차
- 목차
- 목차
- 목차



- 파이썬 개요



1 파이썬이란?

파이썬은 배우기 쉽고 누구나 무료로 사용할 수 있는 오픈 소스 프래밍 언어입니다.

1991년 네덜란드의 귀도 반 로섬(Guido van ROssum)이 개발 했으며, 비영리 단체인 파이썬 소프트웨어 재단이 관리 하고 있습니다.

파이썬의 사전적 의미는 '비단뱀' 으로 그리스 신화에서 유래됐으며, 실제로는 개발자인 귀도가 애청하던 영국의 프로그래미 주인공인 6인조 코미디 그룹의 이름 'Monty Python'에서 따온 것이라고 합니다.

파이썬은 배우기 쉽고 간결하며, 개발속도가 빠르고 강력하기 때문에 현재 미국과 우리나라의 대학등 전 세계적으로 많이 가르치는 프로그래밍 언어중 하나이며 특히 비전공자의 컴퓨팅 사고력을 키우기 위한 프로그래밍 언어로도 많이 활용되고 있습니다.

실무에서도 사용이 급증하고 있으며 이를 증명하듯 스택오버플로에서 '가장 빠르게 성장하는 프로그래밍 언어'로 선택되기도 했습니다.

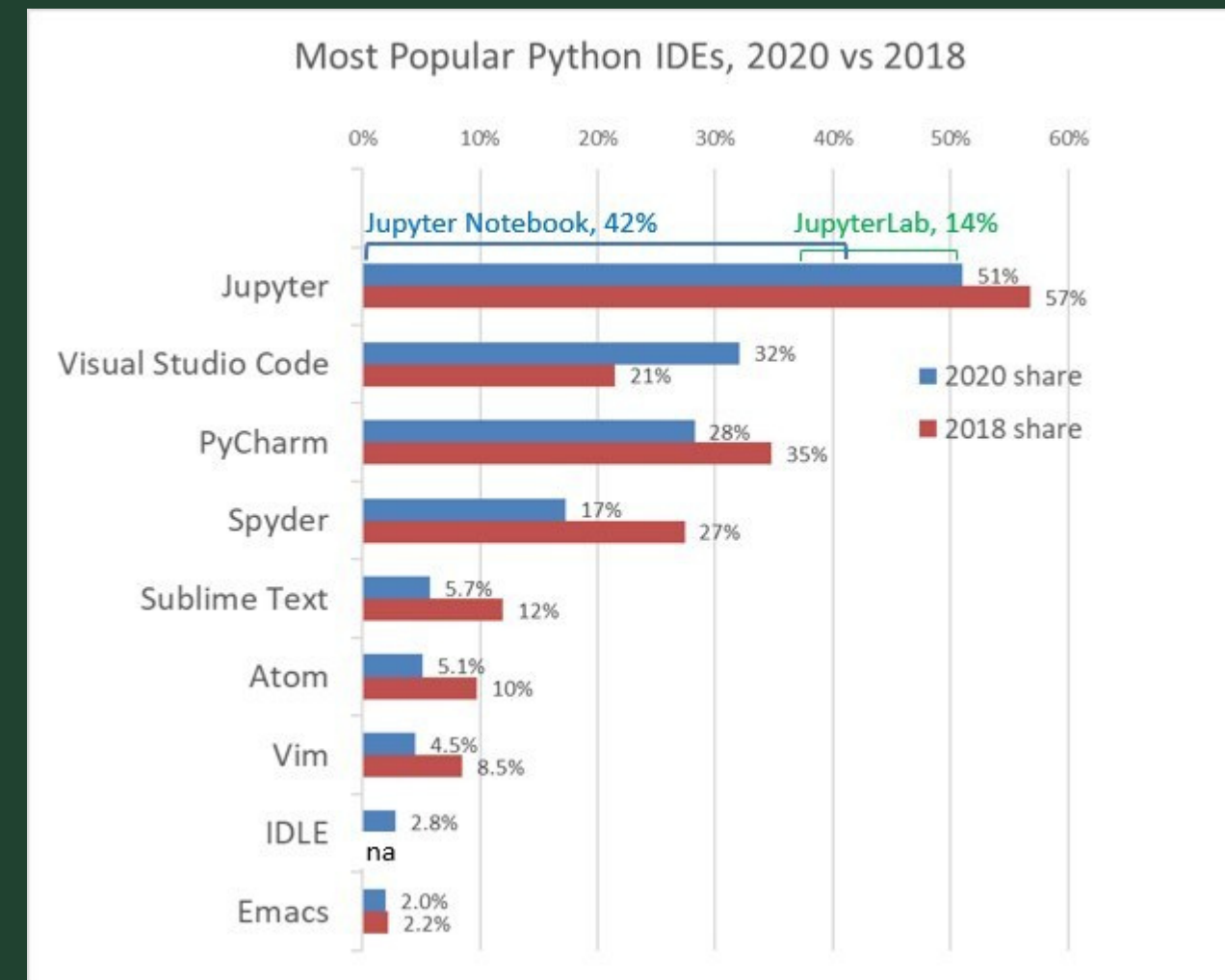
- 파이썬 개발도구 (IDE)



1 파이썬 개발도구의 종류

Python 코드를 작성하는 가장 좋은 방법은 통합 개발 환경 (IDE)을 사용하는 것입니다. 그들은 더 쉽게 작업 할뿐만 아니라 프로그래밍 경험과 효율성을 향상시킵니다.

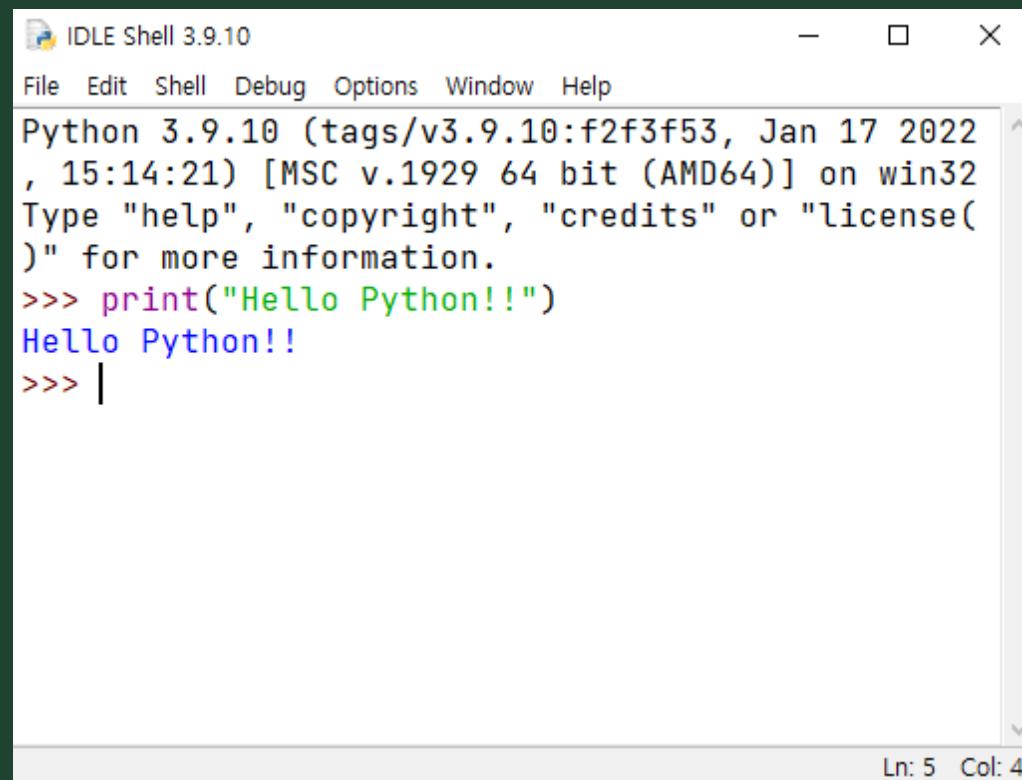
1. Pycharm
2. Jupyter (Anaconda)
3. Visual Studio Code (VS Code)
4. Atom
5. Spyder
6. ViM



<파이썬 개발도구 사용순위 비교 2020 vs 2018>

- 파이썬 개발 도구 (IDE)

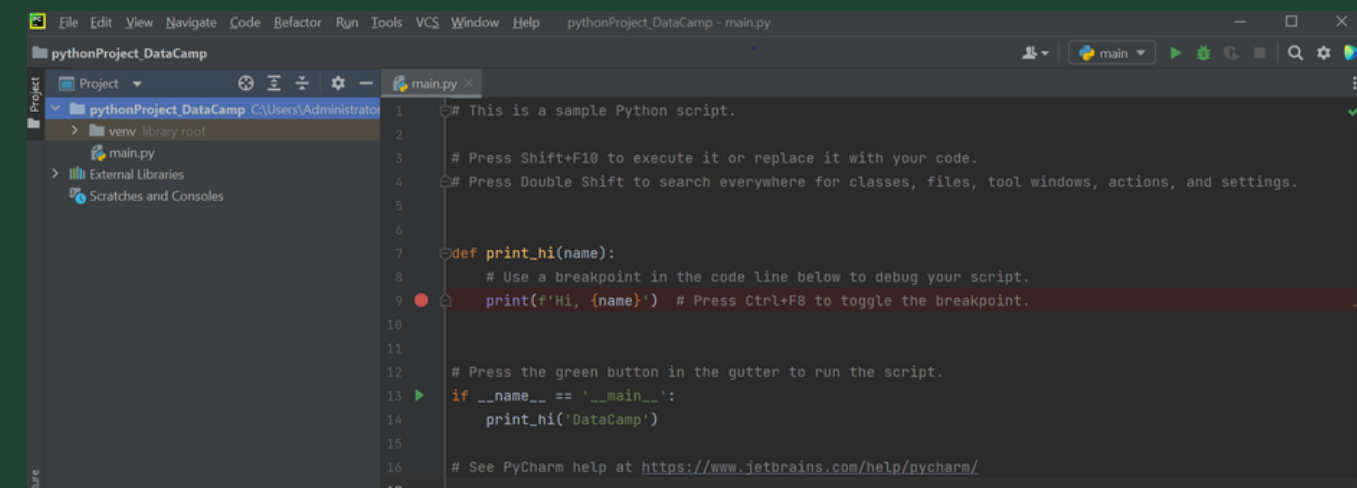
1 Python IDLE



```
Python 3.9.10 (tags/v3.9.10:f2f3f53, Jan 17 2022, 15:14:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello Python!!")
Hello Python!!
>>> |
```

- 파이썬을 설치할 때 기본적으로 설치되는 개발 도구
- 파이썬 셸은 파이썬의 주요 개발 도구인 인터프리터, 디버거 등과 사용자인 개발자 사이를 이어주는 대화형 프로그램으로 '대화형 창'의 모습을 띈다.

2 Pycharm



- 가장 유명한 python 개발 도구
- 자동 코드 완성, 오류 감지, 빠른 수정 등과 관련하여 도움을 주는 개발자에게 스마트 플랫폼을 제공합니다. 크로스 플랫폼 개발과 같은 풍부한 기능을 지원하므로 개발자가 다른 플랫폼에서도 스크립트를 작성할 수 있습니다.
- PyCharm는 유명한 소프트웨어 개발 회사 JetBrains에 의해 개발되었습니다. 그것은 인공지능 및 기계 학습 측면에서 가장 좋은 Python IDE로 간주됩니다

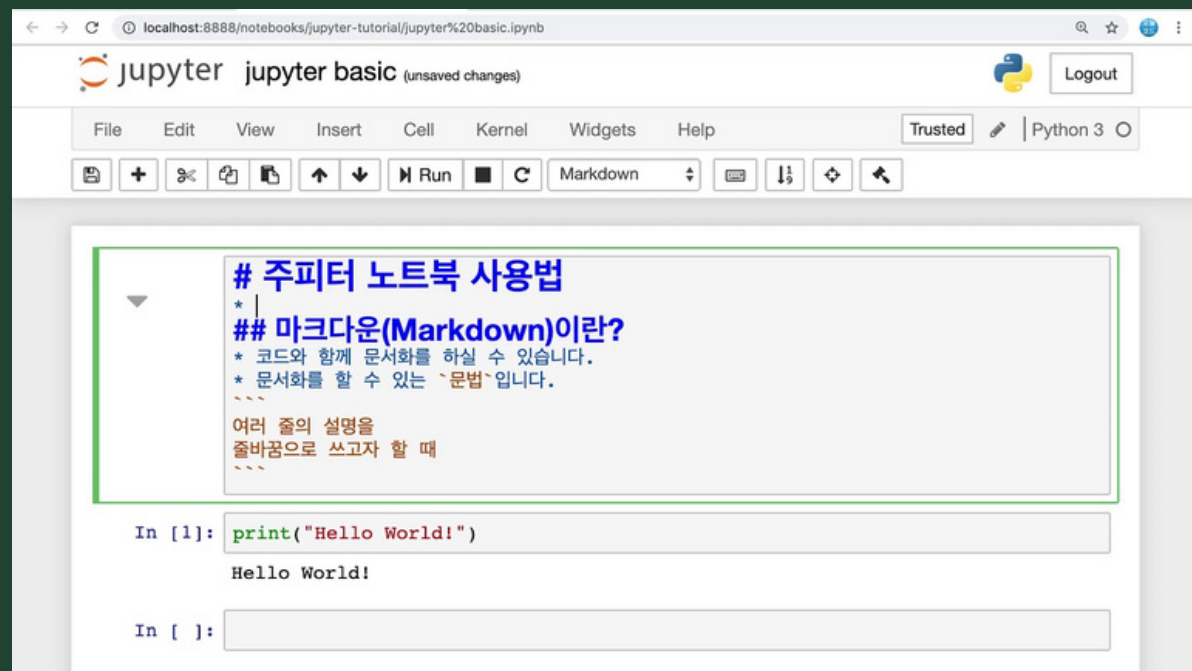


- 파이썬 개발도구 (IDE)



PYTHON

3 Jupyter Notebook

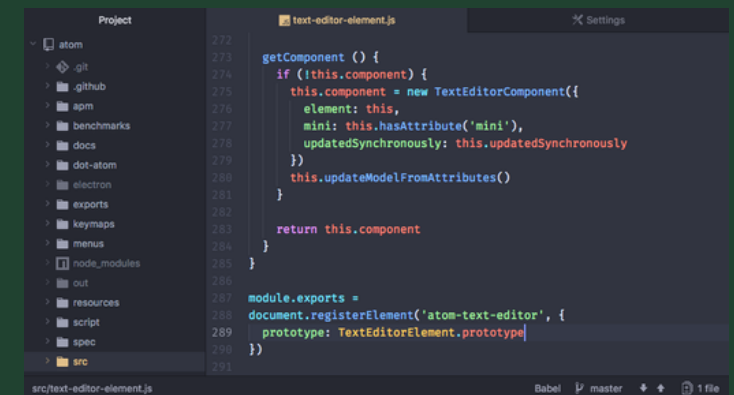


Jupyter Notebook의 가장 큰 장점은 단축키입니다. Jupyter Notebook에서 제공하는 단축키에 익숙해지면, 마우스가 필요 없어지고 개발 속도도 빨라집니다. 그리고 하나의 파이썬 파일에서 코드를 나눠서 실행해볼 수 있습니다. 디버깅 기능은 없지만, 이를 활용하여 디버깅을 할 수 있습니다

4 Atom

Atom은 무료 소스 코드 편집기이며 기본적으로 Node.js에서 개발된 플러그인 지원이있는 웹 기술을 통해 구축된 데스크톱 애플리케이션입니다.

거의 모든 프로그래밍 언어 (예 : PHP, Java)와 호환되는 오픈 소스 편집기입니다. 그것은 정기적으로, 신뢰할 수 있고 널리 사용되며 널리 사용됩니다. Github는 Atom에 대한 강력한 기능을 개발하여 SQL queries 패키지, Markdown Preview Plus 패키지 및 편집, 시각화 및 LaTeX 수식을 렌더링하기위한 패키지를 포함하여 훌륭한 프로그래밍 경험을 제공합니다.



- 기본 문법



PYTHON

1 문자열

- 파이썬에서 작은 따옴표나 큰 따옴표로 앞뒤를 둘러싸 '문자열' 또는 "문자열" 처럼 표현한다.

문자 하나도 문자열로 취급하고 따옴표로 둘러 쌓인 숫자도 문자열로 취급한다. 즉, 따옴표로 둘러 싸면 모두 문자열이다.

- 더하기 기호인 +는 문자열에서 문자열을 연결 하는 역할을 한다.

```
IDLE Shell 3.9.10
File Edit Shell Debug Options Window Help
Python 3.9.10 (tags/v3.9.10:f2f3f53, Jan 17 2022, 15:14:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Python " + "program")
Python program
>>> |
```

- asterisk (*) 는 문자열에서 문자열의 지정된 수만큼 반복하는 연산을 수행 합니다.

```
IDLE Shell 3.9.10
File Edit Shell Debug Options Window Help
Python 3.9.10 (tags/v3.9.10:f2f3f53, Jan 17 2022, 15:14:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('파이썬' * 4)
파이썬파이썬파이썬파이썬
>>> |
```

- 문자열이 길거나 필요에의해 문자열을 처리하기 위해 삼중따옴표를 이용하여 처리 합니다.

```
IDLE Shell 3.9.10
File Edit Shell Debug Options Window Help
Python 3.9.10 (tags/v3.9.10:f2f3f53, Jan 17 2022, 15:14:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('''String operator + and * are very easy!''')
String operator + and * are very easy!
>>> |
```

- # - 한 줄 주석
''' ''' - 여러줄 주석

```
IDLE Shell 3.9.10
File Edit Shell Debug Options Window Help
Python 3.9.10 (tags/v3.9.10:f2f3f53, Jan 17 2022, 15:14:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> '''주석으로 사용가능'''
'주석으로 사용가능'
>>> #주석으로 사용가능
>>> |
```

- 기본 문법



PYTHON

2 정수와 실수

- 숫자는 간단히 정수(integer)와 실수(real 또는 float)로 나눕니다.

```
>>> print(15)      Integer
15
>>> print(3.14)    Float
3.14
```

- 실수 표현

```
>>> 2.7834e-4
0.00027834
>>> 2.7834E5
278340.0
```

- 연산자

*	(multiply)	곱하기
/	(divide)	나누기
%	(modulus)	나머지
//	(floor division)	몫 나누기
**	(exponent)	거듭제곱

- eval - 문자열 실행 함수

```
>>> eval('3 + 15 / 2')
10.5
>>> eval('"java" * 3')    - 문자열 연산도 가능하다.
'java java java'
```


- 기본 문법



PYTHON

3 변수와 키워드, 대입 연산자

- 변수란, '변하는 자료를 저장하는 메모리 공간'이다. 변수는 자료를 담을 수 있는 그릇이며, 그릇에 이름인 태그가 부여 있다고 생각하면 이해하기가 쉽습니다.

- `type()` 함수를 통해 지정된 상수의 자료형을 알 수 있습니다.

```
>>> type(3)
<class 'int'>
>>> type(3.14)
<class 'float'>
>>> type('python')
<class 'str'>
```

- 변수

```
>>> unit = 3      - 상수인 3에 대입할수 없습니다.
```

```
>>> 3 = unit
```

- 대입 연산자

```
>>> a = b = c = 5      - 여러가지 변수에 같은 값을 대
>>> print(a)            입 할 수 있습니다.
```

```
5
>>> print(b)
5
```

- 기본 문법 ○



PYTHON

4 자료의 표준 입력과 자료 변환 함수

- input() 함수로 문자열 표준 입력

```
univ = input('대학은? ')
name = input('이름은? ')
print('대학: ', univ, '이름: ', name)
```

대학은? 동양미래대학교

이름은? 김동양

대학: 동양미래대학교 이름: 김동양

- input 함수는 사용자로부터 입력을 받는 함수로써 명령프롬프트로부터 사용자의 입력을 받을 수 있습니다.

input()함수의 반환으로 사용자가 입력한 것을 "문자열" 타입으로 반환 합니다.

- univ 변수와 같이 특정변수에 사용자의 입력을 처리 할 수 있습니다.

- 기본 문법



PYTHON

5 문자열 다루기

- len() 함수를 통해 문자열의 길이를 참조 할 수 있습니다.

```
>>> a = 'python'
>>> len(a)
6
```

- 문자열의 문자 참조

0	1	2	3	4	5
p	y	t	h	o	n

-6	-5	-4	-3	-2	-1
p	y	t	h	o	n

```
>>> 'python' [-1]
'n'
```

- 슬라이싱

0	1	2	3	4	5
p	y	t	h	o	n

-6	-5	-4	-3	-2	-1
p	y	t	h	o	n

```
>>> 'python' [2:4]      -양수 활용
'th'
>>> 'python' [-5 : -1]  -음수 활용
'ytho'
>>> 'python' [-6 : 4]   -혼합 활용
'pyth'
```

- start 와 end를 비우면 처음부터와 끝까지를 의미 하게 됩니다.

- 기본 문법 ○



PYTHON

6 조건과 반복

- 조건에 따른 선택을 결정하는 if문

```
weather = '화창'
if weather == '화창':
    print('어제 산 신발을 신고가야지!')
```

- 조건에 따라 하나를 선택하는 if ... else

```
n = int(input('정수 입력 >> '))
if n%2 == 0:
    print('%d은 짝수다.' % n)
else:
    print('%d는 홀수다.' % n)
```

- 여러 조건 중에서 하나를 선택하는 구문 if elif

```
point = 82
if 90 <= point:
    print('점수 {}, 성적{}'.format(point, 'A'))
elif 80 <= point:
    print('점수 {}, 성적{}'.format(point, 'B'))
elif 70 <= point:
    print('점수 {}, 성적{}'.format(point, 'C'))
elif 60 <= point:
    print('점수 {}, 성적{}'.format(point, 'D'))
else:
    print('점수 {}, 성적{}'.format(point, 'F'))
```

- 기본 문법 ○



PYTHON

6 조건과 반복

- 조건식이 true일 경우에 반복하는 while문

```
n = 1
while n <= 5:
    print(n, end = ' ')
    n += 1
```

- 반복횟수를 알고 있을때 사용하는 for 문

```
for i in 1.1, 2.5, 3.6, 4.2, 5.4:
    sum += i
    print(i, sum)
```

- 내장함수 range()를 사용한 for문

```
for i in range(5):
    print(i, end = ' ')
```

0 1 2 3 4 5

```
for i in range(1, 10):
    print(i, end = ' ')
```

1 2 3 4 5 6 7 8 9

- OPENCV

1 Open CV 란?

- Open Source Computer Vision Library 의 약어로 오픈소스 컴퓨터 비전 라이브러리 입니다.
- 실시간 영상처리에 중점을둔 영상 처리 라이브러리로서, Apache 2.0 라이선스하에 배포되어 학술적 용도외에 상업적 용도로도 사용할 수 있습니다.
- 계산 효율성과 실시간 처리에 중점을 두고 설계 되었습니다.
- 500가지가 넘는 알고리즘이 최적화돼 있으며 이 알고리즘을 구성하거나 지원하는 함수는 알고리즘 수의 10배가 넘습니다.
- Open CV는 컴퓨터 비전 관련 프로그래밍을 쉽게 할 수 있도록 도와주는 라이브러리로 이미지 딥러닝(알파고기술) 에도 많이 활용된다.



PYTHON

- 물체 인식, 얼굴 인식, 제스처 인식을 비롯해 자율주행 자동차, OCR 판독기, 불량 검사기 등에 활용할 수 있습니다.
- 원래는 인텔이 개발하였다. 인텔 CPU에서 사용되는 경우 속도의 향상을 볼 수 있는 IPP(intel performance Primitives)를 지원한다. 이 라이브러리는 윈도, 리눅스 등에서 사용가능한 크로스 플랫폼이며 오픈 소스 BSD 허가서 하에서 무료로 사용할 수 있다.
- 컴퓨터 비전(computer vision)은 카메라, 에지 기반 또는 클라우드 기반 컴퓨팅, 소프트웨어 및 인공지능(AI)을 결합하여 시스템이 사물을 "확인"하고 식별할 수 있게 합니다. 인텔은 범용 프로세싱을 위한 cpu, 컴퓨터 비전, 비전 처리장치(VPU)를 비롯한 AI 가속화를 지원하는 다양한 기술 포트폴리오를 보유하고 있습니다. 다양한 환경에서 가치를 발휘하는 컴퓨터 비전 시스템은 사물과 사람을 빠르게 인식하고, 고객 통계자료를 분석하고, 공산품을 검사하는 등의 작업을 수행할 수 있습니다.

컴퓨터 비전은 딥러닝을 사용하여 이미지 처리 및 분석 시스템

- OPENCV

2 파이썬용 OpenCV 설치

- openCV 라이브러리를 사용하기 위해 pip 명령어를 사용하여 설치 합니다.

```
pip install opencv-python
```

```
[1237]  
reserved.  
ject1>pip install opencv-python
```

- ▣ 윈도우 터미널 창에서 설치



PYTHON

- python openCV 는 다음과 같은 네종류의 패키지를 제공

```
opencv-python  
opencv-contrib-python  
opencv-python-headless  
opencv-contrib-python-headless
```

- contrib 포함된 패키지는 확장모듈이 포함된 패키지이며, 추가 모듈이 포함된 opencv를 설치합니다.
- headless가 포함된 패키지는 gui 라이브러리 종속성이 없어 서버 환경(Docker, Cloud) 에서 사용할 수 있는 openCV를 설치합니다.
- 특별한 경우가 아니면 일반적으로 opencv-python 패키지를 사용합니다.

- OPENCV

2 파이썬용 OpenCV 설치

- opencv 설치된 버전 확인 스크립트

```
import cv2  
print(cv2.__version__)
```

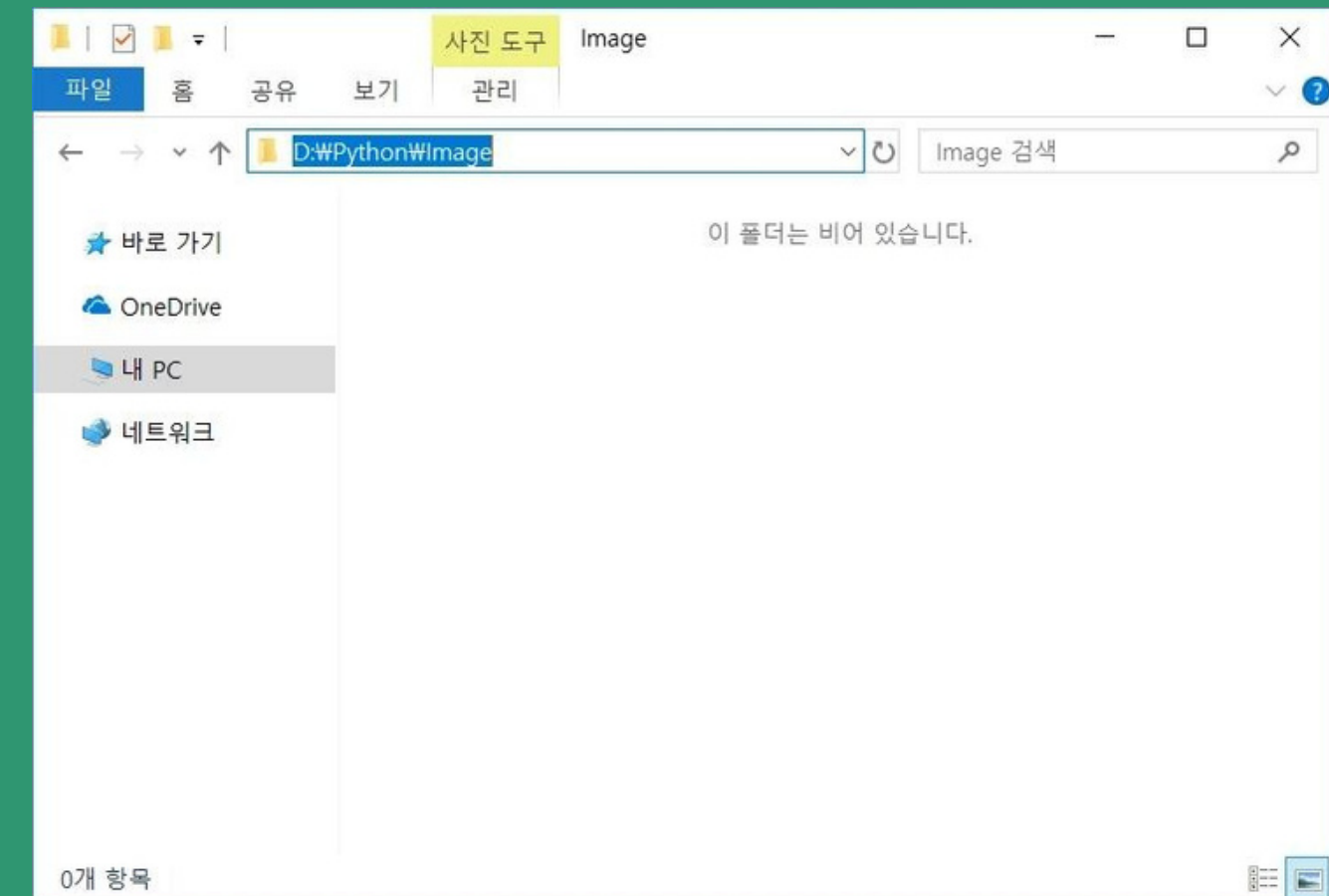
- ▬ 정상적으로 설치가 완료되었다면 OpenCV버전을 출력

```
결과 >>>  
(openCV 버전 출력)
```



PYTHON

- openCV통해 얻어올 이미지 및 동영상 처리



- ▬ D:\Python\Image 폴더 안에 이미지 및 동영상을 저장하여 사용합니다.

- OPENCV

3 OpenCV 이미지 처리

- imread(filename, flags) 함수

```
image = cv2.imread("D:/Python/image/img.jpg",  
cv.IMRED_UNCHANGED)
```

```
image = cv2.imread("D:/Python/image/img.jpg",  
cv.IMRED_UNCHANGED)  
cv.imshow("img", image)  
cv2.waitKey(0)  
cv2.destroyAllWindows()
```

- 2번째 인자(파라미터) 플래그 타입을 설정하지 않은경우

1로 한경우

cv2.IMREAD_COLOR한 경우



PYTHON

flags 타입	내용
IMREAD_COLOR	별도로 지정하지 않은 경우 기본값, 3채널 BGR 컬러 이미지로 반환
IMREAD_GRAYSCALE	단일 채널 회색조 이미지로 반환(내부 코덱 변환)
IMREAD_UNCHANGED	이미지 원본 그대로 반환한다. 투명도(Alpha 채널)를 포함, EXIF값은 무시됨
IMREAD_ANYDEPTH	설정된 경우 입력에 해당 깊이가있을 때 16 비트 / 32 비트 이미지를 반환하고 그렇지 않으면 8 비트로 변환합니다.
IMREAD_ANYCOLOR	설정된 경우 가능한 모든 색상 형식으로 이미지를 읽습니다.
IMREAD_LOAD_GDAL	설정된 경우 이미지로드에 gdal 드라이버를 사용합니다.
IMREAD_REduced_GRAYSCALE_2	설정된 경우 항상 이미지를 단일 채널 회색조 이미지로 변환하고 이 이미지 크기를 1/2로 줄입니다.
IMREAD_REduced_COLOR_2	설정된 경우 항상 이미지를 3 채널 BGR 컬러 이미지로 변환하고 이 이미지 크기를 1/2로 줄입니다.
IMREAD_REduced_GRAYSCALE_4	설정된 경우 항상 이미지를 단일 채널 회색조 이미지로 변환하고 이 이미지 크기를 1/4로 줄입니다.
IMREAD_REduced_COLOR_4	설정된 경우 항상 이미지를 3 채널 BGR 컬러 이미지로 변환하고 이 이미지 크기를 1/4로 줄입니다.
IMREAD_REduced_GRAYSCALE_8	설정된 경우 항상 이미지를 단일 채널 회색조 이미지로 변환하고 이 이미지 크기를 1/8로 줄입니다.
IMREAD_REduced_COLOR_8	설정된 경우 항상 이미지를 3 채널 BGR 컬러 이미지로 변환하고 이 이미지 크기를 1/8로 줄입니다.
IMREAD_IGNORE_ORIENTATION	설정된 경우 EXIF의 방향 플래그에 따라 이미지를 회전하지 마십시오.

- OPENCV

3 OpenCV 이미지 처리

- `imwrite(filename, image)` 함수

```
cv2.imwrite('save_image.jpg', src)
```

- 이미지를 다른 이름으로 저장하기 위해 `cv.imwrite()` 함수를 사용 합니다.



PYTHON

flags 타입	내용
IMWRITE_JPEG_QUALITY	JPEG의 경우 0에서 100까지의 품질이 될 수 있습니다 (높을수록 좋습니다). 기본값은 95입니다.
IMWRITE_JPEG_PROGRESSIVE	JPEG 기능 활성화 (0 또는 1), 기본값은 False입니다.
IMWRITE_JPEG_OPTIMIZE	JPEG 기능 활성화 (0 또는 1), 기본값은 False입니다.
IMWRITE_JPEG_RST_INTERVAL	JPEG 재시작 간격, 0-65535, 기본값은 0-재시작 안 함.
IMWRITE_JPEG_LUMA_QUALITY	별도의 채도 품질 수준, 0-100, 기본값은 0-사용하지 않습니다.
IMWRITE_PNG_COMPRESSION	PNG의 경우 0에서 9까지의 압축 수준이 될 수 있습니다. 값이 높을수록 크기가 작고 압축 시간이 길어집니다. 지정된 경우 전략은 IMWRITE_PNG_STRATEGY_DEFAULT (Z_DEFAULT_STRATEGY)로 변경됩니다. 기본값은 1 (최고 속도 설정)입니다.
IMWRITE_PNG_STRATEGY	기본값
IMWRITE_PNG_BILEVEL	바이너리 레벨 PNG, 0 또는 1, 기본값은 0입니다.
IMWRITE_PXM_BINARY	PPM, PGM 또는 PBM의 경우 이진 형식 플래그 0 또는 1 일 수 있습니다. 기본값은 1입니다.
IMWRITE_EXR_TYPE	

- OPENCV

4 OpenCV 카메라 출력

- 비디오 출력 클래스(cv.VideoCapture)

```
import cv2

capture = cv2.VideoCapture(0)
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

while cv2.waitKey(33) < 0:
    ret, frame = capture.read()
    cv2.imshow("VideoFrame", frame)

capture.release()
cv2.destroyAllWindows()
```



PYTHON

```
capture = cv2.VideoCapture(0)
```

- cv2.VideoCapture)를 통해 내장카메라 또는 외장카메라에서 정보를 받아 올 수 있습니다.
- index 값은 카메라의 장치 번호를 의미합니다. 내장카메라가 존재할 경우 장치번호는 0번이 됩니다.

카메라를 추가적으로 연결하여 외장카메라를 사용하는 경우 장치번호가 1~n 까지 순차적으로 할당됩니다.

```
capture.set(cv2.CAP_PROP_FRAME_WIDTH,640)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
```

- capture.set(propid, value)로 카메라의 속성과 값을 설정할 수 있습니다. CAP_PROP_FRAME_WIDTH,640 는 너비를 640으로 변경하는 것을 의미 합니다.

- OPENCV



4 OpenCV 카메라 출력

- 비디오 출력 클래스(cv.VideoCapture)

```
import cv2

capture = cv2.VideoCapture(0)
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

while cv2.waitKey(33) < 0:
    ret, frame = capture.read()
    cv2.imshow("VideoFrame", frame)

capture.release()
cv2.destroyAllWindows()
```

```
while cv2.waitKey(33) < 0:
    ret, frame = capture.read()
    cv2.imshow("VideoFrame", frame)
```

- 반복문을 이용해서 카메라에 프레임을 지속적으로 받아옵니다.
- cv.waitKey 는 지정된 33만큼 키입력이 있을때까지 프로그램을 지연시킵니다.
- cv.imshow(winname, mat)으로 윈도우 창의 제목과 이미지를 할당합니다.

```
capture.release()
cv2.destroyAllWindows()
```

- capture.release로 동영상 파일을 닫고 카메라 장치에 받아온 메모리를 해제합니다.
- cv2.destroyAllWindows()는 특정 윈도우창을 닫을때 사용합니다.

- OPENCV

4 OpenCV 비디오 출력

- 비디오 출력 클래스(cv.VideoCapture)

```
import cv2
capture = cv2.VideoCapture("Image/Star.mp4")

while cv2.waitKey(33) < 0:
    if capture.get(cv2.CAP_PROP_POS_FRAMES) ==
capture.get(cv2.CAP_PROP_FRAME_COUNT):
        capture.set(cv2.CAP_PROP_POS_FRAMES, 0)
    ret, frame = capture.read()cv2.imshow("VideoFrame",
frame)

capture.release()
cv2.destroyAllWindows()
```



PYTHON

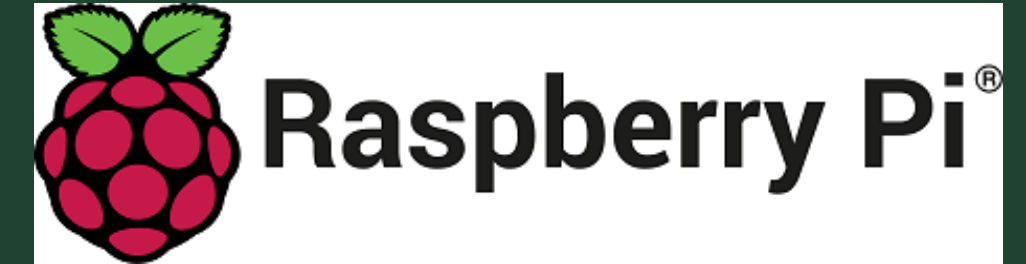
```
if capture.get(cv2.CAP_PROP_POS_FRAMES) ==
capture.get(cv2.CAP_PROP_FRAME_COUNT):
    capture.set(cv2.CAP_PROP_POS_FRAMES, 0)
```

- 비디오 속성 반환 메소드로 비디오의 속성을 반환합니다.

- videoCapture 메소드

메서드	의미
capture.isOpened()	동영상 파일 열기 성공 여부 확인
capture.open(filename)	동영상 파일 열기
capture.set(propid, value)	동영상 속성 설정
capture.get(propid)	동영상 속성 반환
capture.release()	동영상 파일을 닫고 메모리 해제

- 라즈베리파이



1 라즈베리파이

- Raspberry pi(라즈베리파이) 는 영국의 Raspberry Pi 재단이 학교와 개발도상국에서 기초 컴퓨터 과학 교육에 대한 보급을 증진 및 확대를 목적으로 개발한 신용카드 크기의 싱글 보드 컴퓨터입니다



- Raspberry Pi (라즈베리파이) 를 컴퓨터처럼 사용하기 위해서는 몇 가지 주변 장치들이 필요합니다. 모니터, 키보드, 마우스는 물론이고, HDMI (혹은 Micro-HDMI) 케이블, 전원 연결선, 그리고 SD (혹은 Micro-SD) 카드가 필요합니다. 앞서 말씀드린 주변 장치들을 Raspberry Pi (라즈베리파이) 에 연결하면 완전한 컴퓨터로 사용할 수 있게 됩니다.
- Raspberry Pi (라즈베리파이) 에서 접미 상표인 “Pi” 는 본래 Python (파이썬) 의 코드를 편집하거나 실행할 수 있는 컴퓨터를 제조할 목적에서 착안해 결합되었는데요. Raspberry Pi (라즈베리파이) 의 Pi 는 Python (파이썬) 을 의미합니다. 일반적으로, C 나 C++ 의 프로그래밍 언어 보다는 Python (파이썬) 이 친근하고 배우기 쉬운 언어로 알려져 있습니다.



감사합니다.!

