

# 파이썬 특강



2022.05.16

장윤재 janggoons@syu.ac.kr

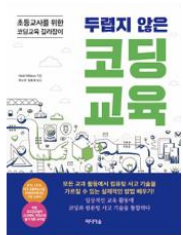
SW융합교육원

# Index

- 특강 소개
- 활동 Activity
- 실습 Practice
- 논의 Discussion
- 이론 Lecture
- SW 비교과 프로그램 안내

# 교수자 소개

- 장윤재, 컴퓨터 교육학자 Ph.D. Computer Science Education
- 대학에서, 컴퓨터 교육 Computer Science Education 전공
- 주로, 초보자를 위한 컴퓨팅 교육 Computing Education 연구 및 실천



# 특강 목표

- 파이썬을 통해 해결할 수 있는 다양한 문제를 살펴보고,
- 자신의 전공 또는 관심 분야의 문제를 해결하기 위한 아이디어를 찾는다.



# 활동 ACTIVITY


“컴퓨터 프로그램, 코드란 무엇일까?”

“코드는 지금껏 내가 알고 있는 것과 전혀 다른 별개의 것일까?”

# Write & Draw

### Write & Draw #1 worksheet

**A.**



A. 작품 제목 :

A. 작가 이름 :

A. 작가 이름 :

[illegible]

**B.**

B. 작품 제목 :

B. 작가 이름 :

**Write & Draw #2 worksheet**

**A.**

A blank sheet of graph paper with a grid pattern. The grid consists of small squares formed by thin gray lines. A thicker black border surrounds the entire grid area. There are no markings or text on the grid itself.

A. 작품 제목 :

A. 작가 이름 :

A. 작가 이름 :

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

**B.**

[illegible]

B. 작품 제목 :

B. 작가 이름 :

# Write & Draw

- 어떤 느낌이었나요? 무엇을 알게 되었나요?
- 어떤 점이 어려웠나요? 왜 그렇게 생각하나요?





벽 드로잉  
보스턴 미술관

어떤 쪽 펼쳐진 벽 표면 위에,  
연필을 사용해서,  
50개의 임의의 점을 찍습니다.

점들은 벽 표면에 골고루 분포되어야 합니다.

모든 점들은 직선으로 서로 연결되어야 합니다.

솔 르윗  
벽 드로잉, 보스턴 미술관  
연필

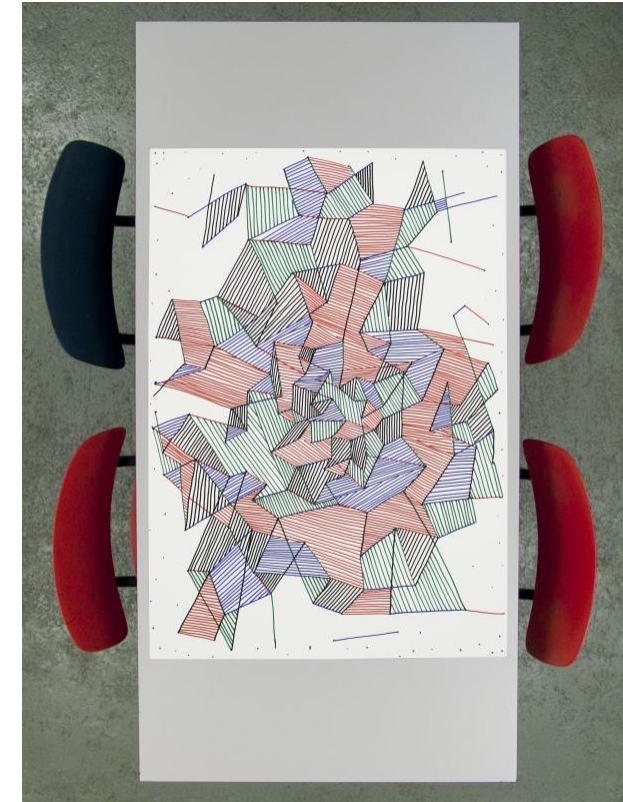


# Write & Draw

- 무엇이 작품일까?
- 저자의 아이디어(생각)가 담긴 것은 무엇일까?

# Conditional Design

- Conditional Design 은,
  - 디자인 전략 중 하나로,
  - 참가자 간의 협력을 자극하고,
  - 예측할 수 없는 결과를 가져오는,
  - 장난스럽게 디자인 된 일련의 규칙과 조건



## Initial setup

1. Play with four players.
2. Each player as a colored pen: red, green, blue and black.
3. Each player places a dot on the sheet at an arbitrary position but not further than 10 cm from each other.
4. The players take turns:

## Task to be performed every turn by each player

### Draw a straight line

1. The line must connect two dots.
2. The angle of the line must be within the following range: 0 to 45 degrees for the black lines, 45 to 90 degrees for the blue lines, 90 to 135 degrees for the red lines, 135 to 180 degrees for the green lines.
3. Connect the line to an existing line if possible.
4. Draw the shortest possible line from your starting point. If you enclose an area (a plane surrounded by lines) then hatch it with lines parallel to the line you enclosed the area with. The enclosed area may not contain unconnected dots or open ended lines.


### Place a dot





The position of the dot is not further away than 10 cm from other dots and not inside the **convex hull** of all dots.

# Formal Language / System

- 형식 언어 / 시스템
  - 구조, 범위 따위가 명확하게 규정되고 정의된 인공 언어.
  - 주로 어학, 논리, 프로그래밍 따위를 위하여 어법을 형식화하여 기술 한 것.





**HOW TO MARK SCORE**

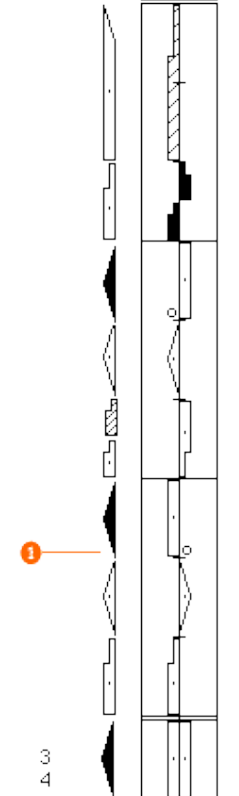
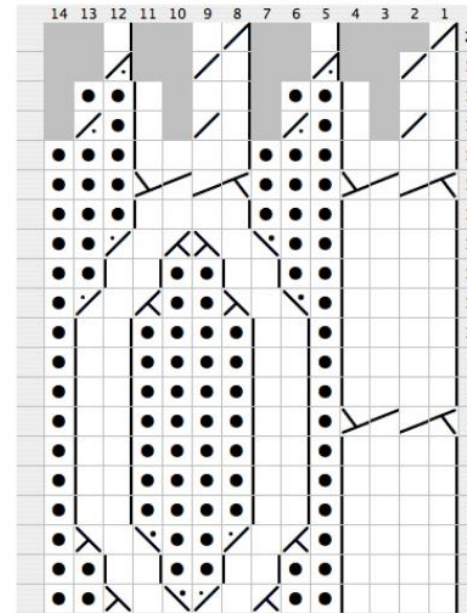
Strike (X) Count 10 plus what you knock down on next two balls.

SPARE (S) Count 10 plus what you knock down on next one ball.

**GOOD BOWLING ETIQUETTE**

- 1- Don't walk in front of someone preparing to bowl.
- 2- Wait until bowlers on the adjacent lanes have completed throwing their ball.
- 3- Avoid delay-be ready to bowl when it's your turn.
- 4- Confine "body English" to width of lane
- 5- Don't mar the approaches with chalk, gum, street shoes, Etc.
- 6- Do not roll ball before automatic pinsetter is ready.
- 7- Do not loft the ball
- 8- Bowling is fun-keep a check on your language and temper.
- 9- Please do not set drinks on the score table. Keep food and drinks back on the bowler's settee area.

	name	1	2	3	4	5	6	7	8	9	10	TOTAL
1	OWALD	10	8	7	7	4	3	3	2	7	1	50
2	HISE	8	4	3	3	6	8	9	7	1	7	75
3	HECK	9	12	25	34	42	51	61	64	71	80	80
4	DINN	1	5	13	17	35	73	52	60	60	87	87
5	NELSON	8	12	29	33	51	61	81	101	112	148	148
6	KARPINSKI	10	4	3	4	5	6	6	7	8	9	91
7												
8												
9												
10												
11												
12												



**Prelude**

Bach  
12 Short Preludes No. 7

©MichaelRosenblatt.com

Ever since my grandmother came to town.

We love to go to Seattle and rode the monorail.

I love to eat SugarBombs. You can feel the sweetness!

My brother loved to play on the saw see.

...And then he went home. The next time I saw him...

Yesterday I saw a pljuics dribbyng down the street

I think the space bar on my keyboard is broken!

This is a run on sentence it is hard to read.

He paired the pear of pares.

# 코드 Code

- 컴퓨터에게 내 생각이나 의도를 전달하기 위한 소통방식
- 코드(프로그램)는 대표적인 ‘형식 언어’
- 사람이 사용하는 자연어와 비슷하면서 다른 점이 있음
- 따라서, 편하게 느껴지는 부분과 불편하게 느껴지는 부분이 있음
- 하지만, 결코 한 가지 방법만 있다고 생각하지 말자.
- 코드(프로그램) 만드는 방법을 배우는 것은,
- 결국, 새로운 관점으로 생각하고 표현하는 방법을 익히는 것이다.



# 실습 PRACTICE

“컴퓨터에게 파이썬 언어로 내 생각을 어떻게 전달할까?”

“컴퓨터의 대답을 어떻게 이해할까?”



# 설치없이 1분만에 파이썬 스크립트 실행하기

- [repl.it](https://repl.it) 에서 스크립트 실행하기
- example 실행하기 > 결과 확인하기 > 서로 설명하기
  - built in functions – print()  
<https://docs.python.org/3.8/library/functions.html#print>
- 실행하기 > ctrl + enter

# 설치없이 1분만에 파이썬 스크립트 실행하기

- print 는 무엇일까?
  - 출력하기
  - 모니터 화면(콘솔창)에 출력하기
- () 는 무엇일까?
  - print() 라고 표시하면 print 라는 이름의 함수 function
  - 함수는 이미 정의된 수행 절차이며, 이를 대표하는 이름이 함수 이름
  - 수학의 함수에서 유래했으나, 정확하게는 다르게 동작함
    - $y = 2x + 3$  에서  $x$  에 어떤 값을 넣는냐에 따라  $y$  값이 달라진다.  $f(x) = 2x + 3$  이라고도 표현한다.

# 설치없이 1분만에 파이썬 스크립트 실행하기

- print() 는 파이썬에서 이미 정의된 함수
  - 내장 함수 built-in function
  - <https://docs.python.org/3.8/library/functions.html#print>
- print() 는 어떻게 사용할 수 있을까?
  - python print function 으로 검색하기
  - [https://www.w3schools.com/python/ref\\_func\\_print.asp](https://www.w3schools.com/python/ref_func_print.asp)

Python ▾

```
print("Hello", "how are you?")  
print("Hello", "how are you?", sep="---")
```

# 설치없이 1분만에 파이썬 스크립트 실행하기

- ‘ ’ 는 무엇일까?
  - print() 는 () 안에 ‘ ’ 로 입력된 내용을 문자열 string 으로 변환하여 출력
  - ‘ ’ 과 “ ” 은 차이가 있을까?
  - python coding style
  - <https://peps.python.org/pep-0008/#string-quotes>

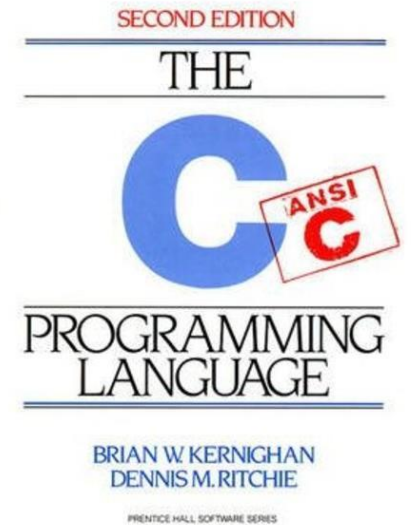
## String Quotes

In Python, single-quoted strings and double-quoted strings are the same. This PEP does not make a recommendation for this. Pick a rule and stick to it. When a string contains single or double quote characters, however, use the other one to avoid backslashes in the string. It improves readability.

Python에서 작은따옴표 문자열과 큰따옴표 문자열은 동일합니다. 이 PEP는 이에 대해 권장하지 않습니다. 규칙을 선택하고 준수하십시오. 그러나 문자열에 작은따옴표 또는 큰따옴표 문자가 포함된 경우 다른 하나를 사용하여 문자열에서 백슬래시를 방지하십시오. 가독성을 향상시킵니다.

# 설치없이 1분만에 파이썬 스크립트 실행하기

- Hello World! 는 무엇일까?
  - 1974년, 벨 연구소에서 브라이언 커니핸이 작성한 매뉴얼 “Programming in C: A Tutorial” 의 코드에서 시작
  - 이후, 1978년, 브라이언 커니핸과 데니스 리치가 저술한 책 “The C Programming Language” 에서도 소개됨
  - 컴퓨터에서 작성한 프로그램이 잘 동작하는지(컴파일러, 런타임 환경 등) 확인하는 가장 간단한 코드
  - 즉, 처음으로 내가 컴퓨터와 대화하는 방법을 터득하는 코드



# 파이썬 실습환경 준비

- 파이썬 설치하기
  - <https://www.python.org/>
  - Downloads > Download the latest version for Windows (3.10.4)
  - Install Now > 환경변수 설정(설치 시 add Python 3.10 to PATH 체크)
- 통합개발환경(IDE) 설치하기(파이참 PyCharm)
  - <https://www.jetbrains.com/ko-kr/pycharm/download/#section=windows>
  - Community 버전으로 다운로드

# 파이썬 실습환경 준비

- 바탕화면에 새로운 폴더 만들기
  - 폴더명 : python
- 새로운 폴더로 이동한 뒤, 해당 폴더에서 파이참 실행하기
  - 마우스 오른쪽 클릭 > open folder as PyCharm Community Edition Project
- 또는, 파이참 실행 후, 폴더 열기
  - File > Open... > ...Desktop\python 선택하기
- 실습 코드 다운로드 받기
  - 다운로드 링크 : <https://github.com/janggoons/python-sl>
  - 압축 해제 후 폴더를 python 폴더로 옮기기

# 똑딱똑딱 파이썬 웹 크롤링 - 워드클라우드

- 실습 개요
  - 내가 좋아하는 가수의 노래 가사는 어떤 특징을 가지고 있을까?
  - 워드클라우드를 만들어 보자.
- 워드 클라우드란?
  - 텍스트 문서의 키워드, 개념 등을 직관적으로 파악하기 위하여 핵심 단어를 시각화하는 방법
  - 많이 언급되는 단어를 크게 표시하여 한 눈에 들어올 수 있도록 표현하고, 주로 대용량의 문서 데이터의 특징을 도출할 때 사용함
- 파이썬 라이브러리 : wordcloud



```
# sample01-01.py
from wordcloud import WordCloud          # pip install wordcloud
import matplotlib.pyplot as plt          # pip install matplotlib

# .txt 파일에 있는 텍스트를 가져오기
lyrics = open('source.txt', 'r', encoding='utf-8-sig')
lyrics = lyrics.read()
#print(lyrics)                            # 읽어온 텍스트 출력하기

# 워드클라우드 생성
wc = WordCloud(background_color='white', width=800, height=800)
wc.generate(lyrics)

# 워드클라우드 저장
f = plt.figure(figsize=(8,8))
plt.axis('off')
plt.imshow(wc)
f.savefig('./sample01-01_result.png')
```

```
# sample01-02.py
import numpy as np                # pip install numpy
from PIL import Image
from wordcloud import WordCloud   # pip install wordcloud
import matplotlib.pyplot as plt   # pip install matplotlib

# .txt 파일에 있는 텍스트를 가져오기
lyrics = open('./source.txt', 'r', encoding='utf-8-sig')
lyrics = lyrics.read()

# 워드클라우드의 모양과 폰트 설정
mask = np.array(Image.open('./sample01_mask.png'))

# 제외할 단어 추가하기
stopwords = {'나'}

# 워드클라우드 생성
wc = WordCloud(background_color='white', mask=mask, width=800, height=800, stopwords=stopwords)
wc.generate(lyrics)

# 워드클라우드 저장
f = plt.figure(figsize=(8,8))
plt.axis('off')
plt.imshow(wc)
f.savefig('./sample01-02_result.png')
```

# 똑딱똑딱 파이썬 웹 크롤링 - 텍스트 정보 가져오기

- 실습 개요
  - 학사 정보 게시판에는 어떤 게시글이 올라올까?
  - 게시글의 정보를 가져와서 저장해보자.
  - 웹 데이터를 크롤링 해보자.
- 크롤링과 크롤러
  - 크롤링 Crawling : 웹에서 자동으로(주로 봇을 이용) 정보를 수집하는 행위
  - 크롤러 Crawler : 크롤링을 위해 제작된 프로그램
  - 스크래핑 Scraping : 모든 소스에서 원하는 정보를 검색, 추출하는 행위
  - 크롤링은 스크래핑 기술 중 하나
- 파이썬 라이브러리 : BeautifulSoup4

```
# sample02-01.py
from urllib.request import urlopen
from bs4 import BeautifulSoup          # pip install beautifulsoup4
import pandas as pd                    # pip install pandas

html = urlopen("https://www.syu.ac.kr/academic/academic-notice/")
soup = BeautifulSoup(html, 'html.parser')

tag_tbody = soup.find('tbody')
#print(tag_tbody)
#print(tag_tbody.find_all('tr'))
#post = tag_tbody.find('tr')
#print(post.find('span', attrs={'class':'tit'}).string)
result = []
for post in tag_tbody.find_all('tr'):
    post_title = post.find('span', {'class':'tit'})
    #print(post_title.text)
    post_writer = post.find('td', {'class':'step3'})
    #print(post_writer.text)
    post_date = post.find('td', {'class':'step4'})
    #print(post_date)
    result.append([post_title.text] + [post_writer.text] + [post_date.text])

#print(result)

post_tbl = pd.DataFrame(result, columns=('title', 'writer', 'date'))
#print(post_tbl)

post_tbl.to_csv('./post.csv', encoding='cp949', mode='w', index=True)
```

# 똑딱똑딱 파이썬 웹 크롤링 - 이미지 파일 가져오기

- 실습 개요
  - '000' 관련 이미지를 모두 수집하고 싶다.
  - 마우스를 클릭해서 직접 다운로드 하는 것 보다 자동으로 다운로드 받고 싶다.
  - 사용자의 동작에 따라 변화는 웹 페이지 정보도 다운로드 받고 싶다.
- 정적 크롤링과 동적 크롤링
  - 정적 static 크롤링 : 웹 서버에 미리 저장되어 전송되는 정적 웹 페이지를 크롤링 하는 것
  - 동적 dynamic 크롤링 : 사용자의 조작에 따라 페이지 내용이 변하는 동적 웹 페이지를 크롤링 하는 것
- 파이썬 라이브러리 : selenium

# 똑딱똑딱 파이썬 웹 크롤링 - 이미지 파일 가져오기

- 실습 준비하기 (1)
  - 크롬 설치하기
    - 다운로드 링크 : [https://www.google.com/intl/ko\\_kr/chrome/](https://www.google.com/intl/ko_kr/chrome/)
    - 크롬 버전 확인하기 : 설정 > 도움말 > 크롬 정보 > 버전 확인 (ex: 101.xxx 이라면 101)
  - 크롬 드라이버 다운로드
    - 다운로드 링크 : <https://chromedriver.chromium.org/downloads>
    - 이전 단계에서 확인한 크롬 버전에 맞는 파일 다운로드
    - 윈도우는 chromedriver\_win32
  - python 폴더에 selenium 폴더 만들고, 크롬 드라이버 파일 이동하기
  - selenium 폴더에서 파이썬 실행하기

# 똑딱똑딱 파이썬 웹 크롤링 - 이미지 파일 가져오기

- 실습 준비하기 (2)
  - 셀레니움 실행 테스트

```
from selenium import webdriver
```

```
driver = webdriver.Chrome(executable_path="chromedriver.exe")  
driver.get("https://google.com")
```

```
# sample03-01.py
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

# 크롬 브라우저 실행하고, python.org 사이트로 이동하기
driver = webdriver.Chrome()
driver.get("http://www.python.org")

# 검색창에 pycon 을 입력하여 검색하기
elem = driver.find_element_by_name("q")
elem.clear()
elem.send_keys("pycon")
elem.send_keys(Keys.RETURN)

# 브라우저 종료하기
# driver.close()
```



```
# sample03-02.py
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
import urllib.request

# 크롬 브라우저 실행하기
driver = webdriver.Chrome()

# 구글 이미지 검색 페이지로 이동하기
driver.get("https://www.google.co.kr/imghp?hl=ko&ogbl")

# 이미지 검색 창에서 키워드 이미지 검색하기
elem = driver.find_element_by_name('q')
elem.send_keys("삼육대학교")
elem.send_keys(Keys.RETURN)

# 검색한 이미지의 첫 번째 이미지를 불러오기
driver.find_elements_by_css_selector(".rg_i.Q4LuWd")[0].click()
time.sleep(3)

# 첫 번째 이미지의 URL 주소를 이용하여 파일로 다운로드하기
img = driver.find_element_by_xpath('//*[@id="Sva75c"]/div/div/div[3]/div[2]/c-wiz/div/div[1]/div[1]/div[3]/div/a/img')
imgUrl = img.get_attribute('src')
urllib.request.urlretrieve(imgUrl, "sample02.jpg")

# 브라우저 종료하기
# driver.close()
```

```
# sample03-03.py
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
import urllib.request

# 크롬 브라우저 실행하기
driver = webdriver.Chrome()

# 구글 이미지 검색 페이지로 이동하기
driver.get("https://www.google.co.kr/imghp?hl=ko&ogbl")

# 이미지 검색 창에서 키워드 이미지 검색하기
elem = driver.find_element_by_name('q')
elem.send_keys("삼육대학교")
elem.send_keys(Keys.RETURN)

# 검색한 이미지 파일에 해당하는 class selector 를 찾아 저장하기
images = driver.find_elements_by_css_selector(".rg_i.Q4LuWd")

# 해당 이미지를 클릭하여 나온 큰 이미지의 URL 주소를 이용하여 다운로드 받아 .jpg 파일로 저장하기
for i, image in enumerate(images):
    image.click()
    time.sleep(3)
    imgUrl = driver.find_element_by_css_selector('.n3VNCb').get_attribute('src')
    urllib.request.urlretrieve(imgUrl, str(i) + ".jpg")

# 브라우저 종료하기
driver.close()
```

```

# sample03-04.py
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.chrome.options import Options
import time
import urllib.request

# 크롬 브라우저 창 크기 설정하고 실행하기
options = Options()
options.add_argument('--window-size=1024,768')
driver = webdriver.Chrome(options=options)

# 구글 이미지 검색 페이지로 이동하기
driver.get('https://www.google.co.kr/imghp?hl=ko&ogbl')

# 이미지 검색 창에서 키워드 이미지 검색하기
elem = driver.find_element_by_name('q')
elem.send_keys('삼육대학교')
elem.send_keys(Keys.RETURN)

# 검색된 페이지를 아래로 스크롤 다운하여 더 많은 이미지 불러오기
body = driver.find_element_by_tag_name('body')
for i in range(50):
    body.send_keys(Keys.PAGE_DOWN)
    time.sleep(0.5)

# 검색한 이미지 파일에 해당하는 class selector 를 찾아 저장하기
images = driver.find_elements_by_css_selector('.rg_i.Q4LuWd')

# 해당 이미지를 클릭하여 나온 큰 이미지의 URL 주소를 이용하여 다운로드 받아 .jpg 파일로 저장하기
for i, image in enumerate(images):
    image.click()
    time.sleep(1)
    imgUrl = driver.find_element_by_css_selector('.n3VNCb').get_attribute('src')
    urllib.request.urlretrieve(imgUrl, str(i) + '.jpg')

# 브라우저 종료하기
# driver.close()

```

## [개선하기]

1. 다운로드 받은 이미지 파일을 다른 디렉터리에 저장하고 싶다. 디렉터리는 어떻게 만들지?
2. 동작이 실행되는 동안 다른 작업을 하고 싶다. 즉, 크롤링 작업을 직접 눈으로 확인하지 않고 동작시키고 싶다.

```

# sample03-04-01.py
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.chrome.options import Options
import time
import urllib.request

# 저장할 디렉터리 만들기
import os
save_dir = 'result'

try:
    if not os.path.exists(save_dir):
        os.makedirs(save_dir)
except OSError:
    print('Error: Creating directory. ' + directory)

# 크롬 브라우저 창 크기 설정하고 실행하기
options = Options()
options.add_argument('--window-size=1024,768')
driver = webdriver.Chrome(options=options)

# 구글 이미지 검색 페이지로 이동하기
driver.get('https://www.google.co.kr/imghp?hl=ko&ogbl')

# 이미지 검색 창에서 키워드 이미지 검색하기
elem = driver.find_element_by_name('q')
elem.send_keys('삼육대학교')
elem.send_keys(Keys.RETURN)

# 검색된 페이지를 아래로 스크롤 다운하여 더 많은 이미지 불러오기
body = driver.find_element_by_tag_name('body')
for i in range(50):
    body.send_keys(Keys.PAGE_DOWN)
    time.sleep(0.5)

```

## [개선하기]

1. 다운로드 받은 이미지 파일을 다른 디렉터리에 저장하고 싶다. 디렉터리는 어떻게 만들지?

```

# 검색한 이미지 파일에 해당하는 class selector 를 찾아 저장하기
images = driver.find_elements_by_css_selector('.rg_i.Q4LuWd')

# 해당 이미지를 클릭하여 나온 큰 이미지의 URL 주소를 이용하여 .jpg 파일로 저장하기
for i, image in enumerate(images):
    image.click()
    time.sleep(1)
    imgUrl = driver.find_element_by_css_selector('.n3VNCb').get_attribute('src')
    urllib.request.urlretrieve(imgUrl, save_dir + '/' + str(i) + '.jpg')

# 브라우저 종료하기
# driver.close()

```

```

# sample03-04-02.py
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.chrome.options import Options
import time
import urllib.request

# 크롬 브라우저 창 크기 설정하고 실행하기
options = Options()
options.add_argument('--window-size=1024,768')

# 헤드리스 headless 옵션 적용하기
options.add_argument('--window-size=1024,768')
options.add_argument('headless')

driver = webdriver.Chrome(options=options)

# 크롬 브라우저 실행하기
driver = webdriver.Chrome(options=options)

# 구글 이미지 검색 페이지로 이동하기
driver.get('https://www.google.co.kr/imghp?hl=ko&ogbl')

# 이미지 검색 창에서 키워드 이미지 검색하기
elem = driver.find_element_by_name('q')
elem.send_keys('삼육대학교')
elem.send_keys(Keys.RETURN)

# 검색된 페이지를 아래로 스크롤 다운하여 더 많은 이미지 불러오기
body = driver.find_element_by_tag_name('body')
for i in range(50):
    body.send_keys(Keys.PAGE_DOWN)
    time.sleep(0.5)

```

## [개선하기]

2. 동작이 실행되는 동안 다른 작업을 하고 싶다.  
즉, 크롤링 작업을 직접 눈으로 확인하지 않고  
동작시키고 싶다.

```

# 검색한 이미지 파일에 해당하는 class selector 를 찾아 저장하기
images = driver.find_elements_by_css_selector('.rg_i.Q4LuWd')

# 해당 이미지를 클릭하여 나온 큰 이미지의 URL 주소를 이용하여 .jpg 파일로 저장하기
for i, image in enumerate(images):
    image.click()
    time.sleep(1)
    imgUrl = driver.find_element_by_css_selector('.n3VNCb').get_attribute('src')
    urllib.request.urlretrieve(imgUrl, str(i) + '.jpg')

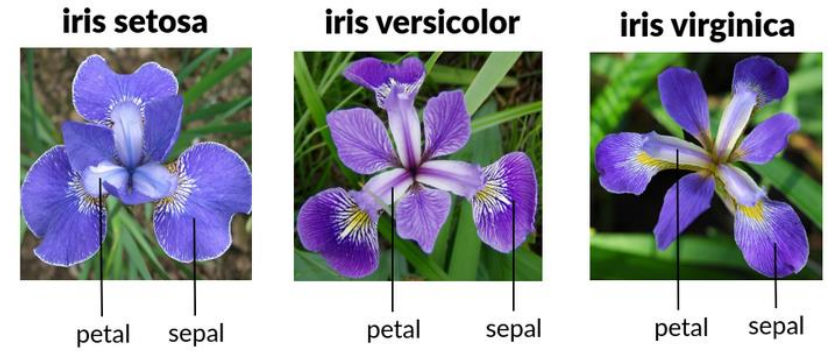
# 브라우저 종료하기
# driver.close()

```

# 웹 크롤링은 신중하게!

- 영리적인 목적으로 운영되는 웹 페이지의 정보를 가져올 때는 신중
- 해외 사례, 링크드인과 웹 크롤링
  - 링크드인(2018). 금지된 소프트웨어 및 확장 기능. [\[링크\]](#)
  - 지디넷코리아(2019). 美 법원 "웹사이트 공개 데이터 스크랩, 불법 아니다" [\[링크\]](#)
- 국내 사례, 야놀자와 여기어때
  - 오피니언타임스(2021). 명품 플랫폼 '무단 크롤링' 논란... 야놀자·여기어때 판결 재조명 [\[링크\]](#)
  - 지디넷코리아(2022). 대법원, 야놀자 정보 크롤링 한 여기어때 창업주 '무죄'. [\[링크\]](#)

# 나도 파이썬 인공지능 개발자



- 실습 개요
  - 붓꽃 품종 분류하기
  - 꽃잎<sub>petal</sub>의 길이와 너비, 꽃받침<sub>sepal</sub>의 길이와 너비를 가지고 3종류의 붓꽃 중 하나로 분류하는 모델 만들기
- 머신러닝
  - 사람은 눈으로 보고 구분한다면, 컴퓨터는 데이터를 보고 특징을 찾아 구분함
  - 주어진 데이터로 분류할 수 있는 특징을 찾는 것이 머신러닝
- 파이썬 라이브러리 : sklearn

# 나도 파이썬 인공지능 개발자

- 구글 코랩으로 실습하기
  - <https://colab.research.google.com/drive/1p5raTiNEnUj7oFIA1K-p7uU6bqXmug8P?usp=sharing>
- 주피터 노트북으로 실습하기
  - > pip install jupyter
  - > jupyter notebook
  - 웹 브라우저에서 실행하기





# 논의 DISCUSSION

“1% 만큼 배웠지만, 나머지 99% 는 상상해보자”

“1% 씩 생각해서 나누면 금방 99%를 모을 수 있다”

# 내가 할 수 있는 것과 하고 싶은 것

- 파이썬을 이용하여 해결해보고 싶은 문제를 고민해서,
- 어떻게 하면 해결할 수 있을지 아이디어를, 아래 링크에 입력해 봅시다.
- 링크 : <https://bit.ly/3wb3eqJ>
- 그리고, 다른 사람의 아이디어를 보고 이야기를 나눠봅시다.
  - 나랑 비슷한 아이디어가 있나요?
  - 내가 생각해보지 못한 흥미로운 아이디어가 있나요?
  - 함께 해결해보고 싶은 아이디어가 있나요?













# 이론 LECTURE

“파이썬이 더 궁금하다면...”

# 가장 인기있는 프로그래밍 언어, 파이썬

## • TIOBE Index for May 2022

May 2022	May 2021	Change	Programming Language	Ratings	Change
1	2	▲	 Python	12.74%	+0.86%
2	1	▼	 C	11.59%	-1.80%
3	3		 Java	10.99%	-0.74%
4	4		 C++	8.83%	+1.01%
5	5		 C#	6.39%	+1.98%
6	6		 Visual Basic	5.86%	+1.85%
7	7		 JavaScript	2.12%	-0.33%
8	8		 Assembly language	1.92%	-0.51%
9	10	▲	 SQL	1.87%	+0.16%
10	9	▼	 PHP	1.52%	-0.34%

source: <https://www.tiobe.com/tiobe-index/>

## • PYPL (May 2022)

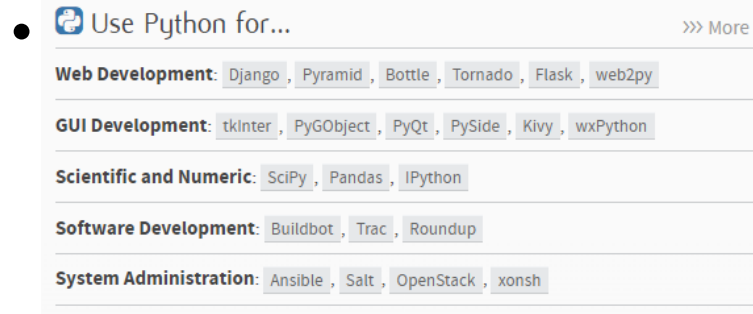
Worldwide, May 2022 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	27.85 %	-2.5 %
2		Java	17.86 %	-0.1 %
3		JavaScript	9.17 %	+0.4 %
4		C#	7.62 %	+0.7 %
5		C/C++	7.0 %	+0.4 %
6		PHP	5.36 %	-1.0 %
7		R	4.34 %	+0.5 %
8	▲▲▲▲	TypeScript	2.39 %	+0.7 %
9	▼	Objective-C	2.25 %	+0.0 %
10		Swift	2.05 %	+0.3 %

source: <https://pypl.github.io/PYPL.html>

# 파이썬으로 할 수 있는 다양한 프로젝트 사례

- 파이썬으로 할 수 있는 것

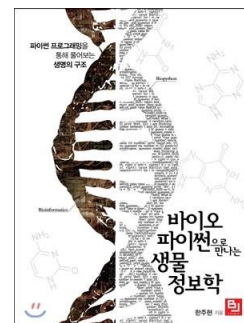
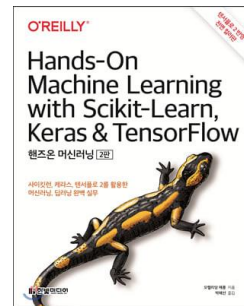
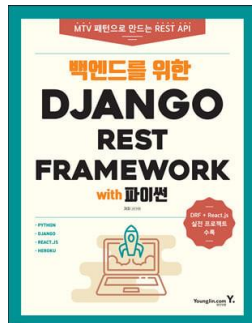


- awesome-python

- <https://github.com/vinta/awesome-python>

- 파이썬을 활용하는 기업

- 구글 google
- 유튜브 youtube
- 드롭박스 dropbox
- 나사 NASA
- 페이스북 Facebook
- 인스타그램 Instagram
- 넷플릭스 Netflix



# 파이썬을 배우면 좋은 이유

- 자동화를 쉽게 할 수 있다.
  - (웹에서) 데이터를 얻고 데이터를 정리하거나, 문서를 압축하고 정리하거나,
  - 유튜브 비디오를 다운로드 받거나, 오디오 파일을 압축하거나,
- 코드 읽기가 쉽다.

Python ▾

```
print("Hello World!")
```

C ▾

```
#include <stdio.h>
int main() {
    printf("Hello, world!\n");
    return 0;
}
```

Java ▾

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

# 파이썬을 배우면 좋은 이유

- 커뮤니티가 크다.
  - Python Korea : <https://www.facebook.com/groups/pythonkorea/>
  - Pycon Korea : <https://2022.pycon.kr/about/previous-pyconkr>
- (hit!) 데이터 과학 또는 인공지능 기술을 배울 수 있다.
- (new!) HTML 에서 파이썬 코드 작성 가능
  - PyScript : <https://pyscript.net/>

# 파이썬을 더 배우고 싶다면

- 책을 사서 혼자서 공부하기
  - 기타를 배우듯이, 운동을 배우듯이, 영어를 배우듯이
  - 작심삼일을 10번 하면 1달, 120번 하면 1년을 할 수 있다!
  - 나의 아이디어 구현하기(클론, 토이 프로젝트)
- 온라인 영상 또는 커뮤니티 활용하기
  - 생활코딩, 파이썬 입문 : <https://opentutorials.org/course/4769>
  - 모두를 위한 파이썬 : <https://www.edwith.org/pythonforeverybody>
  - 파이콘 참여하기, 파이썬 코리아 커뮤니티 참여하기
- 교내 교과 또는 비교과 프로그램 활용하기
  - 전공, 융합전공, 교양 교과





**QNA**

01



# SW중심대학 사업소개



# 01. SW중심대학 사업소개

## SW중심대학이 무엇인가요?

SW중심대학은  
대학교육을 SW중심으로 혁신함으로써,  
SW전문인력을 양성하고  
학생·기업·사회의 SW경쟁력을 강화해  
진정한 SW가치 확산을 실현하는 대학을 말합니다.



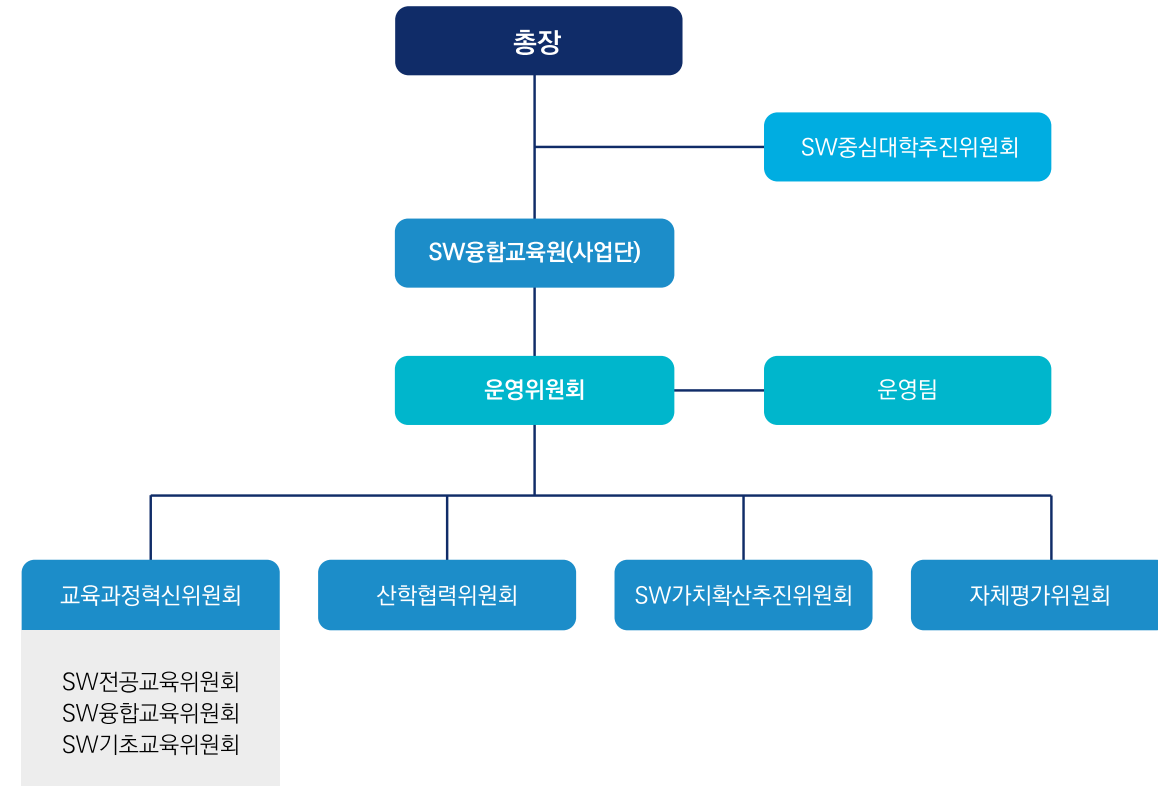
## SW중심대학에서는 무엇을 배우나요?

수요자에 맞는 SW교육을 통해  
전공자는 글로벌 경쟁력을 갖춘 실무형인재로,  
비전공자는 타 전공지식과 SW소양을 겸비한  
융합 인재로 양성시킵니다.

전공자는 이론과 실전 모두 잡는  
SW전문 교육을 통해 현장에서 능력을 발휘할 수 있는  
실무형 인재로 성장하도록!



비전공자는 전공별 특성에 맞는  
SW기초교육을 통해 SW소양을 쌓을 수 있도록!



# 01. SW중심대학 사업소개

사업기간 2021.04.01 ~ 2026.12.31

지원규모 특화트랙 2단계 선정  
총 6년간 60억 국고 지원

## 인력양성

### SW전공

- 미래융합대학소속 학과 재학생

### SW연계전공

- SW중독심리, SW중독재활, SW보건빅데이터

### SW비전공

- SW기초교육 필수 이수

SW기초교육

SW전공교육

SW융합교육

SW산학협력

SW가치확산

02



# 2021 학년도 SW 비교과 프로그램



## 02. 2021학년도 SW비교과프로그램

### 특강

- 「SW Step-Up 캠프」 4회
- 「오픈소스 SW라이선스」 1회
- 「메타버스」 1회
- 「AI」 1회

### 예비 신입생 입학 전 사전교육 「SW와 미래사회」

- 교육대상 : 수시합격자(406명)
- 교육방법 : e-Class System
- 교육 이수 시, 정규 교과목  
「SW중심의 미래사회」 3학점 인정

### 기타

- 「SW경진대회 지원사업」 상시 운영

### 대회

- 「SW프로젝트 경진대회」 1회
- 「SW알고리즘 경시대회」 1회

### 공모전

- 「SW프로그램 참여후기 공모전」 1회
- 「SW비교과프로그램 아이디어 공모전」 1회

### SW중심대학동아리 운영

- 동아리 13팀 128명
- 동아리 지원사항  
지도교수 1명, 전문가 멘토 1명, 회의비,  
재료비, 도서구입, 동아리방 지원 등



03



# 2022학년도 SW비교과프로그램



### 03. 2022학년도 SW비교과프로그램

#### 특강 및 설명회

- 「SW Step-Up 캠프」
- 「Python 특강」
- 「Big Data 특강」
- 「Metaverse 특강」
- 「AI 특강」
- 「XR 특강」
- 「SW창업 특강」
- 「SW중심대학 설명회」

#### 대회 및 공모전

- 「SW알고리즘 경진대회」
- 「SW프로젝트 경진대회」
- 「SW해커톤 경진대회」
- 「SW창업 아이디어 경진대회」
- 「SW프로그램 참여후기 공모전」
- 「SW비교과프로그램 아이디어 공모전」
- 「외부 SW경진대회 지원사업」
- 「TOPCIT 정기평가 지원」



04



# SW마일리지 운영 안내



## 04. SW마일리지

!! SW중심대학사업단 주관 프로그램만 인정 !!

### 비교과 프로그램 단계별분류

#### [1단계] 인지

핵심역량 개념 소개 및 체험을 통해 관심과 흥미를 유발하고  
기초 지식 및 기능을 습득하는 SW비교과프로그램

[1단계] 인지	2시간 미만	30	-
	2시간이상 ~ 4시간미만	50	-
	4시간이상 ~ 8시간미만	70	-
	8시간이상 ~ 16시간미만	90	-
	16시간이상 ~ 20시간미만	110	-
	20시간이상	130	-

#### [3단계] 심화

핵심역량의 핵심 지식 및 기능을 영역별로 학습하고  
집중 실습으로 심화하는 SW비교과프로그램

[3단계] 심화	2시간 미만	50	-
	2시간이상 ~ 4시간미만	90	-
	4시간이상 ~ 8시간미만	130	-
	8시간이상 ~ 16시간미만	170	-
	16시간이상 ~ 20시간미만	210	-
	20시간이상	250	-

#### [2단계] 발달

핵심역량의 기초 지식 및 기능을 활용하여 간단한 실습을  
진행하여 역량을 본격적으로 형성할 수 있는 SW비교과프로그램

[2단계] 발달	2시간 미만	40	-
	2시간이상 ~ 4시간미만	70	-
	4시간이상 ~ 8시간미만	100	-
	8시간이상 ~ 16시간미만	130	-
	16시간이상 ~ 20시간미만	160	-
	20시간이상	190	-

#### [4단계] 발현

핵심역량 관련 지식 및 기능을 종합적으로 활용하여 성과를  
도출하는 SW비교과프로그램

[4단계] 발현	2시간 미만	60	-
	2시간이상 ~ 4시간미만	110	-
	4시간이상 ~ 8시간미만	160	-
	8시간이상 ~ 16시간미만	210	-
	16시간이상 ~ 20시간미만	260	-
	20시간이상	310	-

## 04. SW마일리지

❗ SW중심대학사업단 주관 프로그램만 인정 ❗

### 자격증 / 시험 / 경력

분류	항목		마일리지	상한 점수	인증 조건
자격증 및 시험	TOPCIT 시험성적	~ 299	200	200	성적표 사본 제출
		300 ~ 399	300	300	
		400 ~ 499	400	400	
		500 ~ 599	500	500	
		600 ~	600	600	
	SW자격증 취득	기사급	500	-	자격증 사본 제출 * 인정 자격증: 국제자격증(CCNF, OCJD, MCSE, MCAD, OCP, SCNA급), 기사급의 국가공인자격증
		산업기사급	300	-	자격증 사본 제출 * 인정 자격증: 국제자격증(CCNA, CCDA, OCJP, MCP, OCA, SCSA, SCDA급), 산업기사급의 국가공인자격증
경력	SW 대회 · 공모전	참가	교내	20	참가증명서, 입상증명서(상장 등)
			교외	200	
		입상(교외)		1,000	
	SW특허	등록	1,000	-	특허등록증 사본
		출원	200	-	출원사실증명원 사본
	SW논문	게재	1,000		논문 및 게재(예정) 확인서
		발표	800		확인 가능한 증빙자료

## SW중심대학사업단 카카오톡 채널



[http://pf.kakao.com/\\_BxkkWb](http://pf.kakao.com/_BxkkWb)



감사합니다.