|  |  |
| --- | --- |
| 교육제목 | 데이터 기반 인공지능 시스템 엔지니어 양성 과정 |
| 교육일시 | 210927 |
| 교육장소 | YGL 학과장 |
| **교육내용** | |
| 오전 | 1. **모듈**   표준 모듈 등의 정보가 궁금할때는 Help > Python Reference > Library에서 찾아보면 됨  import math 입력 후 math 모듈 사용 가능  ex)  import math  print("- math.sin(5) =",math.sin(5))  print("- math.cos(5) =",math.cos(5))  print("- math.tan(5) =",math.tan(5))  print("- math.floor(5.5)(내림) =",math.floor(5.5))  print("- math.ceil(5.4)(올림) =", math.ceil(5.4))   1. **from 구문**   # from 모듈명 import 가져오고자 하는 변수 또는 함수.  from math import sin, trunc, floor  from math import \* #모두 다 가져오기, 이렇게는 잘 안 쓸듯  # math. 을 앞에 안써도 됨  print(sin(10))   1. **as 구문**   모듈의 이름이 너무 길어 짧게 줄여 사용하고 싶은 경우 : - import 모듈 as 사용하고 싶은 식별자  import math as m  print(m.sin(1))  print(m.cos(1))  print(m.tan(1))  print(m.floor(2.5))  print(m.ceil(2.5))  <정리>  모듈을 프로그램에 가져오는 방법 :  import 모듈명  import 모듈명 as 약어  from 모듈명 import 변수 또는 함수   1. **random 구문**   import random  print(" # random 모듈 ")  # random() : 0.0 <= x < 1.0 사이의 float를 리턴.  print("- random():", random.random())  # uniform(min, max) : 지정한 범위 사이의 float를 리턴.  print("- uniform(10, 20): ", random.uniform(10,20))  # randrange() : 지정한 범위의 int를 리턴  # - randrange(max) : 0 ~ max 값 리턴  # - randrange(min, max) : min부터 max사이의 값을 리턴.  print("- randrange(10): ", random.randrange(10))  # choice(list) : 리스트 내부에 있는 요소를 랜덤하게 선택  print("- choice([1,2,3,4,5]):", random.choice([1,2,3,4,5]))  # shuffle(list) : 리스트의 요소들을 랜덤하게 섞음  print("- shuffle([1,2,3,4,5])", random.shuffle([1,2,3,4,5]))  # sample(list, k=<숫자>) : 리스트의 요소 중에 k개를 뽑음  print("- sample([1,2,3,4,5],k=2)", random.sample([1,2,3,4,5], k=2))  a = ["월","화","수","목","금"]  print(random.shuffle(a))   1. **sys모듈**   시스템과 관련된 정보를 가지고 있는 모듈. 명령 매개변수를 받을 때 많이 사용됨.  사실 잘 쓰지는 않음  import sys  # 명령 매개변수를 출력합니다.  print(sys.argv)  print("---")  # 컴퓨터 환경과 관련된 정보를 출력합니다.  print("getwindowsversion:()", sys.getwindowsversion())  print("---")  print("copyright:", sys.copyright)  print("---")  print("version:", sys.version)  # 프로그램을 강제로 종료합니다.  sys.exit()   1. **os 모듈 : 운영체제와 관련된 모듈**   import os  print("현재 운영체제 :", os.name)  print("현재 폴더: ",os.getcwd())  print("현재 폴더 내부의 요소 : ", os.listdir())  # 폴더 만들고 제거  os.mkdir("hello")  os.rmdir("hello")  with open("original.txt", "w") as file:  file.write("hello")  os.rename("original.txt", "new.txt")  # 파일 제거  os.remove("new.txt")  # os.unlink("new.txt")   1. **# datetime 모듈 : date, time 관련된 코드들**   import datetime  print("현재 시간 :", datetime.datetime.now())  now = datetime.datetime.now() # now는 변수  print("{}년 {}월 {}일 {}시 {}분 {}초".format(now.year,now.month,now.day,now.hour,now.minute,now.second))  # 조금 더 쉽게 만드는 방법  out\_now = now.strftime("%Y{} %m{} %d{} %H{} %M{} %S{}".format(\*"년월일시분초"))  # 문자열 리스트 등 앞에 \* 붙이면 요소 하나하나가 매개변수로 지정됨.  print("{}년 {}월 {}일 {}시 {}분 {}초".format(now.year,now.month,now.day,now.hour,now.minute,now.second))  print("out\_now :", out\_now)  import time  print("5초동안 시간 정지")  time.sleep(5)  print("시간 정지 end")   1. **urllib 모듈 : 인터넷 주소에서 자료를 가져오는 라이브러리**   from urllib import request  # urlopen() 함수로 구글 메인 페이지 읽기  target = request.urlopen("https://google.com")  output = target.read()  #출력  print(output) # HTML 구문을 해석해야 함  # 330p, 문3)  output = os.listdir("./") # "."은 현재 디렉토리, .listdir(path) path: 경로 ./ : 현재 디렉토리(. = ./)  print("os.listdir() :", output)  print()  for path in output:  if os.path.isdir(path):  print("Folder :", path)  else:  print("File :", path)  print()    # 문4)  # 모듈 읽어 들이기  import os  # 폴더를 읽어 들이는 함수  def read\_folder(path):  # 폴더의 요소 읽어 들이기  output = os.listdir(path)  for item in output:  if os.path.isdir(item):  read\_folder(item)  else:  print("File :", item)    read\_folder(input("검색하고자 하는 디렉토리를 입력 > "))   1. **외부 모듈**   외부 모듈 설치하기  # !pip install 외부모듈 -> cmd 창에서 실행  from bs4 import BeautifulSoup  from urllib import request  target = request.urlopen("https://google.com")  output = target.read()  html = '''  <html>  <head>  <title>BeautifulSoup test</title>  </head>  <body>  <div id='upper' class\_='test' custom='good'>  <h3 title='Good Content Title'>Contents Title</h3>  <p>Test contents</p>  </div>  <div id='lower' class='test' custom='nice'>  <p>Test Test Test 1</p>  <p>Test Test Test 2</p>  <p>Test Test Test 3</p>  </div>  </body>  </html>'''  soup = BeautifulSoup(html) # </html> 태그 형식으로 변경  print(soup.find\_all('select')).  for item in soup.select("div"):  for value in item.select('p'):  print(value)  url1 = 'https://news.v.daum.net/v/20210927112602253'  # 제목을 출력  soup = BeautifulSoup(request.urlopen(url))  print(soup.select("h3").text)  for i1 in soup.select('p').text:  print(i1) |
| 오후 | 1. **함수 데코레이터**   : 함수 앞뒤에 꾸밀 부가적일 내용 혹은 반복할 내용을 데코레이터로 정의해서 손쉽게 사용할 수 있도록 한 것.  #e x)  def test(function):  def wrapper():  print("인사시작")  function()  print("인사끝")  return wrapper  # 데코레이터 붙여 함수 만들기  @test  def hello():  print("하이")  # 함수 호출  hello()   1. **패키지**   패키지 생성하는 방법  패키지 디렉토리 생성  패키지 디렉토리 안에 모듈 생성 ... 모듈 여러개 생성 가능  패키지명.모듈명 import  Ex)  import test\_pakage.module\_a as a  import test\_pakage.module\_b as b  print(variable\_a)  print(variable\_b)  from test\_pakage import \*  print(module\_a.variable\_a)  print(module\_b.variable\_b) |