|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9차시 | 1 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제1) 파일은 어디에 저장되는가?  (1) 중앙 처리 장치  (2) 주 기억 장치  **(3) 보조 기억 장치**  (4) 그래픽 처리 장치    문제2) 다음 확장자 중 바이너리 파일인 것을 모두 선택하세요.  (1) .txt **(2) .jpg**  (3) .docx **(4) .exe**  **(5).sqlite** (6) .py    문제3) 다음은 다른 운영체제에서 절대 경로를 표기한 것들이다. 각 항목마다 해당되는 운영체제를 답한다. 표기가 문법에 어긋난 것은 '틀림'으로 적으세요.  (1) C://Users//user1//ch09.py  **(1) 틀림. 윈도우에서는 경로를 분리하는 '/' 또는 ''를 한 번만 씀**  (2) C:/Users/user1/ch09.py  **(2) 윈도우**  (3) C:\Users\user1\ch09.py  **(3) 틀림. 윈도우용 경로를 파이썬 코드에서 문자열 형태로 표현한다면 이스케이프 시퀀스를 이용해서 ''를"\"로 표현해야 하기는 함. 문자열로 표현하는 것이라면 맞지만, 여기서는 문자열이라는 얘기는 없었으므로 틀렸다고 봐야 함**  (4) C:\Users\user1\ch09.py  **(4) 윈도우**  (5) /home/user1/ch09.py  **(5) 유닉스 계열 (리눅스, 맥 os 등)**  (6) //home//user1//ch09.py  **(6) 틀림**  문제4) 현재 작업 디렉토리를 출력하는 코드를 작성하시오.  **정답4)**  **import os**  **os.getcwd()** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9차시 | 5 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제5) 아래 제시된 내용을 적어서 “data.txt” 파일을 만드시오. 저장할 때 utf-8 부호화 방식으로 저장한다. 그 후 저장된 data.txt 파일에서 내용 한 줄을 읽고 화면에 출력하는 코드를 작성한다.  - "Hello\nThis is data.txt\n이 파일은 utf-8 형식으로 저장했습니다\n"  **정답5)**  **1. 메모장을 열어서 내용을 적고 파일을 저장한다.**  **2. 코드를 이용해서 파일을 생성하려면 다음 코드를 사용한다.**  **f = open("data.txt", "w", encoding="utf-8")**  **f.write("Hello\nThis is data.txt\n이 파일은 utf-8 형식으로 저장했습니다")**  **f.close()**  **3. 한 줄을 읽어서 화면에 출력한다.**  **f = open("data.txt", encoding = "utf-8")**  **line = f.readline()**  **print(line)**  **f.close()**    문제6) 파일 닫기에 대해 틀린 것은?  **(1) 파일을 닫는 것은 파일을 메모리에서 삭제하는 것과 동일합니다.**  (2) file.close() 메서드를 사용하여 파일을 닫을 수 있습니다.  (3) 파일 객체가 메모리에서 삭제되면 자동으로 파일이 닫힙니다.  (4) with 문을 사용하면 파일을 자동으로 닫을 수 있습니다.    문제7) 다음 코드에 오류가 있는지, 있으면 지적하세요(문제 6에서 저장한 “data.txt” 파일을 이용하세요).  with open("data.txt") as f:  for line in f:  print(line.strip())  f.close()  **정답7) 코드에는 문제가 없음. 하지만, 10번 문제에서 저장했던 utf-8 형식으로 저장된 파일을 읽는다면 운영체제의 시스템 설정에 따라 오류가 발생할 수 있음.**  **utf-8 형식으로 인코딩 방법을 강제화시키라면 다음 형태로 수정해야 함**  **with open("data.txt", encoding="utf-8") as f:**  **for line in f:**  **print(line.strip())**  문제8) 6번 문제에서 작성된 "data.txt" 파일의 맨 끝 줄에 다음 문자열을 추가하는 프로그램을 작성한다.  "Here comes to the end."  **정답8)**  **f = open("data.txt", "a", encoding="utf-8")**  **f.write("Here comes to the end.")**  **f.close()** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9차시 | 9 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제9) 아래의 파일의 인코딩 방식을 'utf-8'로 설정하도록 코드를 수정하시오.  with open("python.utf8.txt") as f:  s = f.read()  print(s)  **정답9)**  **with open("python.utf8.txt", encoding="utf-8") as f:**  **s = f.read()**  **print(s)**  문제10) 문제8에서 만들어진 파일의 내용을 한 줄씩 읽어, 새새로운 파일에 저장하는 프로그램을 작성하시오. 단 새로운 파일의 입력은 사용자로부터 입력 받으시오.  **정답10)**  **filename2 = input("새로운 파일 이름을 입력하시오: ")**  **f1 = open("data.txt", encoding="utf-8")**  **f2 = open(filename2, "w", encoding="utf-8")**  **for line in f1:**  **f2.write(line)**  **f1.close()**  **f2.close()**  문제11) 문제8번과 같은 방식으로 간단한 내용을 가지는 data2.txt 파일을 생성한다. 그 두 파일의 내용을 모두 리스트로 만들고, 리스트를 연결해서 새로운 리스트를 생성한다. 새로 만들어진 리스트의 내용을 세 번째 파일에 저장하는 코드를 작성하시오  **정답11)**  **filename1 = input("첫 번째 파일 이름을 입력하시오: ")**  **filename2 = input("두 번째 파일 이름을 입력하시오: ")**  **filename3 = input("세 번째 파일 이름을 입력하시오: ")**  **f1 = open(filename1, encoding="utf-8")**  **lst1 = f1.readlines()**  **f1.close()**  **f2 = open(filename2, encoding="utf-8")**  **lst2 = f2.readlines()**  **f2.close()**  **lst3 = lst1 + lst2**  **f3 = open(filename3, "w", encoding="utf-8")**  **for line in lst3:**  **f3.write(line)**  **f3.close()** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9차시 | 12 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제12) 검색할 문자열을 입력받아, 문제15에서 새로 만들어진 파일에서 검색하고 몇 번째 라인에 있는지 보여주는 코드를 작성해 본다.  **정답12)**  **lineNum = 1**  **found = False**  **filename = input("파일 이름을 입력하세요: ")**  **findStr = input("검색할 문자열을 입력하세요: ")**    **with open(filename, encoding="utf-8") as f:**  **lineNum = 1**  **line = f.readline()**  **while line:**  **if findStr in line:**  **found = True**  **print(f"{lineNum}:{line.strip()}")**  **lineNum += 1**  **line = f.readline()**  **if found == False:**  **print("검색 단어가 파일에 없습니다")**  문제13) 문제16을 readlines()을 이용해서 구현해본다  **정답13)**  **found = False**  **filename = input("파일 이름을 입력하세요: ")**  **findStr = input("검색할 문자열을 입력하세요: ")**  **with open(filename, encoding="utf-8") as f:**  **lines = f.readlines()**  **for lineNum in range(len(lines)):**  **if findStr in lines[lineNum]:**  **found = True**  **print(f"{lineNum + 1}:{lines[lineNum].strip()}")**    **if found == False:**  **print("검색 단어가 파일에 없습니다")** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9차시 | 14 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제14) 강의자료 실습문제 2 에서 파일에서 검색할 문자열을 찾으면, 검색 결과를 새로운 파일에 저장한다. 결과를 저장할 파일 이름은 사용자로부터 입력 받는다. 만약 파일에 검색할 문자열이 없으면, 화면에 "검색 단어가 파일에 없습니다"라는 메시지만 출력하고 파일은 생성하지 않는다.  - 검색할 문자열이 없는 경우, 파일을 생성하지 않는다. 그러나 문자열이 있는지 확인하려면 파일 전체를 검색해야 한다. 전체를 검색한 후 저장할 내용이 없으면 파일을 생성하지 않는다.  - 파일 내용을 전체 검색하면서 검색할 단어가 있으면, 이를 우선 리스트에 저장한다. 그런 다음 리스트의 크기가 1 이상이면 파일을 열고 검색된 내용을 저장한다.  **정답14)**  **filename = input("파일 이름을 입력하세요: ")**  **findStr = input("검색할 문자열을 입력하세요: ")**  **writeFileName = input("검색 내용을 저장할 파일 이름을 입력하세요: ")**  **lst = []**    **with open(filename, encoding="utf-8") as f:**  **lines = f.readlines()**  **for lineNum in range(len(lines)):**  **if findStr in lines[lineNum]:**  **lst.append(str(lineNum + 1) + ":" + lines[lineNum].strip())**    **if (len(lst) > 0):**  **f = open(writeFileName, "w")**  **for line in lst:**  **f.write(line)**  **f.close()**  **else:**  **print("검색 단어가 파일에 없습니다")** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9차시 | 15 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제15) data1.txt와 data2.txt에 각각 오름차순으로 정렬된 실수값들이 한 줄에 한 개씩 저장되어 있다. 두 개 파일의 데이터를 읽고 오름차순으로 정렬해서 "data3.txt" 파일에 저장하는 프로그램을 작성한다  (data1.txt와 data2.txt는 직접 만든다)  **정답15)**  **f1 = open("data11.txt")**  **lines = f1.readlines()**  **list1 = []**  **for line in lines:**  **k1 = float(line)**  **list1.append(k1)**  **f1.close()**  **f2 = open("data12.txt")**  **lines = f2.readlines()**  **list2 = []**  **for line in lines:**  **k2 = float(line)**  **list2.append(k2)**  **f2.close()**    **def sizecomparaison():**  **i = 0**  **n = 0**  **f3 = open("data13.txt", "w")**  **while 1:**  **if len(list1) == i or len(list2) == n:**  **break**  **if list1[i] <list2[n]:**  **f3.write(str(list1[i]) + '\n')**  **i += 1**  **elif list1[i] >list2[n]:**  **f3.write(str(list2[n]) + '\n')**  **n += 1**  **elif list1[i] == list2[n]:**  **f3.write(str(list1[i]) + '\n')**  **f3.write(str(list2[n]) + '\n')**  **i += 1**  **n += 1**  **# print("i = ", i, "n = ", n)**  **if len(list1) == i:**  **while 1:**  **if len(list2) - 1 == n:**  **f3.write(str(list2[n]) + '\n')**  **break**  **else:**  **f3.write(str(list2[n]) + '\n')**  **n += 1**  **if len(list2) == n:**  **while 1:**  **if len(list1) - 1 == i:**  **f3.write(str(list1[i]) + '\n')**  **break**  **else:**  **f3.write(str(list1[i]) + '\n')**  **i += 1**  **f3.close()** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9차시 | 16 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| ~~문제16) shapes.txt 파일에 도형의 모양과 좌표가 순서대로 저장되어 있다.~~    ~~첫 문장은 도형의 모양이다. 도형의 모양 다음에는 도형을 이루는 좌표값들이 나온다.~~  ~~사각형은 대각선을 이루는 두 개 꼭짓점의 좌표값들이 순서대로 배치된다. 사각형의 좌표는 어느 부분의 꼭짓점인지는 모르지만, 절대값(-)와 절대값( - )는 각각 너비와 높이에 해당된다.~~  ~~삼각형은 세 꼭짓점 좌표값들이 순서대로 나타난다. 세 개 꼭짓점의 좌표가 있을 때 삼각형의 면적을 구하는 공식은 8장 문제30에서 설명하는 헤론의 공식을 이용한다. 두 꼭짓점 사이의 거리를 구하는 공식은 아래를 참조한다.~~    ~~면적 =~~  ~~두 점 사이의 거리 =~~    ~~위 공식에서 삼각형 세변의 길이를 각각 a, b, c이며, 원은 반지름 값만 주어진다.~~  ~~shapes.txt파일을 읽고, 각 도형에 대해 면적을 계산해서 화면에 출력하는 프로그램을 작성한다.~~  **~~정답16)~~**  **~~import math~~**  **~~def getShapesInfos():~~**  **~~lst = []~~**  **~~f = open("shapes.txt", encoding="utf-8")~~**  **~~line = f.readline().strip()~~**  **~~while line:~~**  **~~lst.append(line) # 도형 모양 추가~~**  **~~t1 = ()~~**  **~~if line == "사각형":~~**  **~~for s in range(4):~~**  **~~n = int(f.readline())~~**  **~~t1 = t1 + (n,)~~**  **~~elif line == "삼각형":~~**  **~~for s in range(6):~~**  **~~n = int(f.readline())~~**  **~~t1 = t1 + (n,)~~**  **~~elif line == "원":~~**  **~~n = int(f.readline())~~**  **~~t1 = t1 + (n,)~~**  **~~lst.append(t1)~~**  **~~line = f.readline().strip()~~**  **~~f.close()~~**  **~~# print("\n", lst)~~**  **~~return lst~~**  **~~def getDistance(x1, y1, x2, y2):~~**  **~~return math.sqrt((x2 - x1) \*\* 2 + (y2 - y1) \*\* 2)~~**  **~~#print(getDistance(0, 0, 10, 0))~~**  **~~def calcRectangleArea(t):~~**  **~~width = getDistance(t[2], t[1], t[0], t[1])~~**  **~~height = getDistance(t[0], t[3], t[0], t[1])~~**  **~~return width \* height~~**  **~~def calcTriangleArea(t):~~**  **~~width = getDistance(t[4], t[5], t[2], t[3])~~**  **~~height = getDistance(t[4], t[5], t[0], t[1])~~**  **~~return width \* height / 2~~**  **~~def calcCircleArea(t):~~**  **~~return math.pi \* t[0] \* t[0]~~**    **~~l = getShapesInfos()~~**  **~~area = 0~~**  **~~for i in range(0, len(l), 2):~~**  **~~if l[i] == "사각형":~~**  **~~area = calcRectangleArea(l[i + 1])~~**  **~~elif l[i] == "삼각형":~~**  **~~area = calcTriangleArea(l[i + 1])~~**  **~~elif l[i] == "원":~~**  **~~area = calcCircleArea(l[i + 1])~~**  **~~print(f"{l[i]} 면적: {area}")~~** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9차시 | 17 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제17) 사용자로부터 텍스트 파일의 이름을 입력 받고, 주어진 파일을 복사하는 프로그 램을 작성한다. 파일을 복사할 때 학습한 내용만을 이용한다.  **정답17)**  **srcFileName = input("원본 파일 이름을 입력하세요: ")**  **destFileName = input("복사 대상 파일 이름을 입력하세요: ")**  **srcf = open(srcFileName, encoding="utf-8")**  **lines = srcf.readlines()**  **srcf.close()**  **destf = open(destFileName, "w", encoding="utf-8")**  **for line in lines:**  **destf.write(line)**  **destf.close()** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9차시 | 18 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제18) 문법상 오류가 없는 파이썬 코드 파일을 읽고, 주석문을 제거한 코드를 다른 파일에 저장하는 프로그램을 작성한다. 원본 코드 파일 이름은 사용자로부터 입력 받고, 새로 저장되는 코드의 파일 이름은 원본 파일이름 앞에 "new\_"를 붙인다.  **정답18)**  **fileName = input("주석문을 제거할 파이썬 코드 파일 이름을 입력하세요: ")**  **# 새로운 파일 이름 생성**  **newFileName = "new\_" + fileName**  **f = open(fileName, encoding="utf-8")**  **newf = open(newFileName, "w", encoding="utf-8")**  **# 원본 파일에 있는 각 줄을 처리**  **for line in f:**  **withinSmallQuoteString = False # ' ' 문자열에 있는 '#'을 무시하기 위한 변수**  **withinLargeQuoteString = False # " " 문자열에 있는 '#'을 무시하기 위한 변수**  **idx = 0**  **for ch in line:**  **# 문자열 시작 또는 끝 확인**  **if ch == "'":**  **withinSmallQuoteString = not withinSmallQuoteString**  **elif ch == '"':**  **withinLargeQuoteString = not withinLargeQuoteString**  **# '#' 문자가 문자열 내에 있는 것이 아니면, '#' 이후는 제거 후 줄 바꿈 문자 붙이기**  **if ch == '#' and (withinSmallQuoteString == False and withinLargeQuoteString == False): # 문자열 내에 있지 않은 ''#' 처리**  **line = line[:idx] + '\n'**  **break**  **idx += 1**  **if line.strip() != "":**  **newf.write(line)**  **f.close()**  **newf.close()**  문제19) 사용자로부터 텍스트 파일 이름을 입력 받고, 파일에 있는 단어 개수를 출력하는 프로그램을 작성한다. 단어는 줄바꿈 문자, 탭문자, 공백 문자로 분리된다고 가정한다.  **정답19)**  **fileName = input("파일 이름을 입력하세요: ")**  **numWords = 0 # 단어 개수 저장**  **with open(fileName, encoding="utf-8") as f:**  **for line in f: # 파일에 있는 각 줄에 대해서 처리**  **wordsLst = line.split() # 단어를 줄바꿈, 탭, 공백 문자들로 분리**  **numWords += len(wordsLst) # 단어 개수 더하기**  **print(numWords)** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9차시 | 20 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제20) 두 개의 CSV 파일(data1.csv, data2.csv)을 읽어 하나의 CSV 파일(data3.csv) 로 합치고, 두번째 필드인 나이가 20세 이상인 데이터만 가져오는 프로그램 작성.  **정답20)**  **import csv**    **input\_files = ['data1.csv', 'data2.csv']**  **output\_file = 'data3.csv'**  **filter\_column = 1 #두번째 필드**  **filter\_value = 20**  **with open(output\_file, 'w', newline='', encoding="utf-8") as outfile:**  **writer = csv.writer(outfile)**  **for filename in input\_files:**  **with open(filename, 'r', encoding="utf-8") as infile:**  **reader = csv.reader(infile)**  **for row in reader:**  **if int(row[filter\_column]) >filter\_value:**  **writer.writerow(row)** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9차시 | 21 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제21) 주어진 디렉토리에서 특정 확장자를 가진 파일들을 찾아 다른 디렉토리로 복사 하는 프로그램을 작성하세요.  **정답21)**  **import os**  **import shutil**  **input\_dir = 'D:\Python'**  **output\_dir = 'D:\Python3'**  **file\_extension = '.txt'**  **if not os.path.exists(output\_dir):**  **os.makedirs(output\_dir)**  **for root, dirs, files in os.walk(input\_dir):**  **for file in files:**  **if file.endswith(file\_extension):**  **full\_file\_path = os.path.join(root, file)**  **shutil.copy(full\_file\_path, output\_dir)**  문제22) 주어진 텍스트 파일에서 각 단어가 몇 번 등장하는지 세는 프로그램을 작성하세요.  **정답22)**  **from collections import Counter**  **import re**  **filename = 'data1.csv'**  **def word\_count(filename):**  **with open(filename, 'r', encoding="utf-8") as file:**  **text = file.read()**  **words = re.findall(r'\w+', text.lower())**  **word\_counts = Counter(words)**  **for word, count in word\_counts.items():**  **print(f"{word}: {count}")**  **word\_count(filename)** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9차시 | 23 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제23) 주어진 디렉토리 내의 중복 파일을 찾아 삭제하는 프로그램을 작성하세요.  이 문제는 파일 시스템 관리에서 매우 유용합니다.  **정답23)**  **import os**  **import hashlib**    **def get\_file\_hash(filename): #파일의 해시 코드(값)를 제공. 파일의 내용이 같으면 같은 해시값을 가짐.**  **hasher = hashlib.md5()**  **with open(filename, 'rb') as file:**  **buffer = file.read()**  **hasher.update(buffer)**  **return hasher.hexdigest()**    **def deduplicate\_files(directory):**  **seen\_files = {}**  **for root, \_, files in os.walk(directory): #os.walk() 함수는 directory를 시작으로 하위 디렉토리를 재귀적으로 탐색함**  **for filename in files:**  **filepath = os.path.join(root, filename)**  **filehash = get\_file\_hash(filepath)**  **if filehash in seen\_files:**  **os.remove(filepath)**  **print(f"Removed duplicate file: {filepath}")**  **else:**  **seen\_files[filehash] = filepath**    **# 디렉토리 이름을 적절히 바꾸어 사용하세요.**  **deduplicate\_files('target directory')** | | | | | | |