|  |
| --- |
| **SSAC**  **파이썬을 활용한 비즈니스 빅데이터 분석가 양성 과정**  **(TEMPLATE) 자가학습 워크시트** |

|  |  |
| --- | --- |
| **이름** | 백승재 |
| **날짜** | 2021-10-07 |
| **온라인 콘텐츠(인강)**  **학습 범위** | * 직장인을 위한 Python : 05-07~-6-13 * Python : CH07\_01 ~ Ch0704 |
| **복습 노트 작성 (자유 양식)** | - 05-07. List Comprehension  -for~in구조, 편리한 문법 한줄 코딩  -대괄호 안에 for 반복문 -> 반복문 앞에 i변수 -> for 문 뒤에 조건 필터  # 변수 가공 가능 i에 연산자를 가해 가공  - 05-08 문자열 가지고 놀기  - Len 문자열 길이  -split 문자열 쪼개기 : 리스트로 반환된다.  -upper/lower 대문자/소문자 만들기 : 문자열로 반환, 컬럼 값 카운트 전 데이터 대/소문자 통일  -startswith, endswith~로 시작하는, ~로 끝나는 -> 논리값으로 반환(True False)  -replace : 바꾸기, 원본 변경안됨, 변경 필요시 새로운 변수, 또는 자신의 변수에 담아 주기  -strip : 불필요한 좌우 공백 제거하기, 문자열로 반환  - 05-09 패키지와 Import  - 모듈 : 함수들이 뭉쳐진 하나의 .py파일 안에 이루어진 것  - 여러 개의 모듈을 그룹화하면 패키지가 됨  - 패키지는 종종 라이브러리라고도 불림  - 06-01 Numpy란 ~ 06-02 Import와 Alias설정 ndarray(n-dementional array)  -06-03 ndarray생성, array에서의 데이터 타입  -type, shape 확인, array의 data type 단일이어야 함  - int와 float이 혼재인 경우 -> float으로 통일 됨  - dtype지정 (int,float,string)  -06-04 index,slicing : 리스트와 동일함  -06-05 Fancy Indexing : slicing과 비슷하지만 요소가 떨어져 있는 경우에도 Slicing이 가능함  -06-06 Boolean Indexing:  - array[조건 필터], indexig과정 : 조건 필터는 boolean값으로 반환-> 인덱싱  -06-07 arrange(array), range(for문)  - keyword 옵션 : start, end,step  -06-08 정렬(sort) : 내림차순 np.sort()[::-1], method 방식 정렬 : array.sort()  -06-09 인덱스를 반환하는 argsort : 값이 아닌 현재 값에 해당 열 또는 행에서의 인덱스 순위를 반환  -06-10 matrix(행렬) 개요 :  -06-11 matrix(행렬)의 연산 : 덧셈, 뺄셈, 곱셈: shape이 같아야 연산이 가능함(행과 열)  -06-12 matrix(행렬)의 연산 : 벡터 내적, 닿는 shape이 같아야 연산이 가능함  -06-13 Broadcasting : array전체 요소에 같은 연산을 수행해 줌  07-01 데이터 소개와 분석 목표 설명  - 파일이 여러 개 일 때 데이터를 불러 오는 방법  - 빅데이터>Pandas로 어떻게 코드화 시킬까  07-02 문제풀이1 데이터 리스트 불러오기(glob) -> 모든 CSV 병합하기(pd.concat) -> index reset하기  07-03 문제풀이 2~ 07-04. 문제풀이 3  필요 컬럼만 추출하여 DF만들기 ([[]] -> 팬시 인덱싱으로 컬럽 값 으로 슬라이싱하기, 팬시 인덱싱 문자 포함 값으로 슬라이싱하기-> 인덱싱 초기화(index = range(length())-> 출력하기 |
| **퀴즈 제출** | Q1. 다음 출력값이 error가 나오는 경우는 ?  1) array = np.array([1, 3.14, '테디', '1234'])  2) arr = np.array([1, 2, 3, 3.14])  3) arr = np.array([1, 3.14, '1234'], dtype=int)  4) arr = np.array([1, 3.14, '테디', '1234'],dtype=int)  정답 : 4  ‘테디’는 int로 변경 할 수 없다.  array에서는 list와는 다르게 1개의 단일 데이터 타입만 허용된다.  1) string으로 변경되어 출력  2) float으로 변경되어 출력  3) int로 변경되어 출력  Q2. Dataframe df 컬럼중 하나이 상호명 값에 ‘스타벅스’를 포함하는 상호명의 개수를 확인하려고 한다. 상호명에 슬라이싱 방법으로 적절한 것은 ?  (단, 상호명에 해당하는 values들은 결측값을 포함한다.)  1) str.contains(“스타벅스”)를 이용하여 DF 슬라이싱 한다.  2) 결측값이 없는 데이터 프레임을 먼저 구성 후 str.contains(“스타벅스”)를 이용하여 DF 인덱싱한다.  3) df[df[“상호명”]==’스타벅스’]로 슬라이싱한다.  4) df[“상호명”]==’스타벅스’]로 슬라이싱한다.  정답 : 2번  또는 결측치제거(edit\_data = df.dropna())->edit\_data slicing edit\_data[edit\_data[“상호명].str.contains(“스타벅스”)]  해설 :  1) 결측값 에러 발생함    3) 스타벅스특수점(스타벅스 dt점)은 포함안된다.  4) 팬시 인덱싱만 한값으로 슬라이싱하려면 DF[]안에 넣어줘야 하며 해당 값은 3번과 같은 결과값이 나온다.    Q3. 리스트(list1)를 이용하여 3차원 정형 형태의 array(array1)를 생성하려고한다. 3차원 array를 생성할 수 있는 리스트를 고르시오.  # 참고 : array1 =np.array(list)  1) mylist1 = [[[1,2,3,4]],  [[5,6,7,8],[7,8,9,10]],  [[11,12,13,14]]]  2) mylist1 = [[[1,2,3,4]],  [[5,6,8]],  [[9,10,11,12]]]  3) mylist1 = [[[1,2,3,4]],  [[5,6,7,8]],  [[9,10,11,12,15]]]  4) mylist1 = [[[1,2,3,4]],  [[5,6,7,8]]]  정답 : 4 -> array shape (2,1,4)  3차원 배열에서 axis0의 방안에 있는 2 dimentional Shape (행X열)은 1개여야 3차원 정형 형태의 array를 만들 수 있다.    1~3 array 생성 시 warning 발생  1) Z 2번에 행이 1개가 더 있어 비정형  2) Z 2 번에 열이 1개가 부족하여 비정형  3) Z 3 번에 열이 1개 더 많아 비정형 |
| **교육생 사진 (교육생 모습 + 교육 화면)** |  |
| **화면 캡쳐**  **(교육 화면 + 교육일시)** | 텍스트, 스크린샷, 컴퓨터, 노트북이(가) 표시된 사진  자동 생성된 설명 |