|  |  |
| --- | --- |
| 교육 제목 | **가상공간 생성 및 패키지 설치 및 주피터에 등록** |
| 교육 일시 | 2021년 11월 15일 월요일 |
| 교육 장소 | 영우글로벌러닝 2층 |
| **교육 내용** | |
| 오전 | 1. 가상공간 생성    1. 설정 저장       1. conda env export > ml\_packages.yml    2. 생성된 가상공간을 주피터 노트북에서 인식시키기       1. python -m ipykernel install --user --name ml(가상공간 이름) 2. array    1. array 생성   >>> a = np.array([1,2,3 ])  >>> b = np.array( [ ( 1 . 5,2,3), (4, 5,6) ], dtype = float)  >>> c = np.array([[ ( 1 . 5,2,3), (4, 5,6) ] , [(3,2, 1), (4, 5,6)]], dtype = float   * 1. array 탐색   >>> np.info(np.ndarray.dtype)  >>> a.shape  >>> len(a)  >>> b.ndim  >>> e.size  >>> b.dtype  >>> b.dtype.name  >>> b.astype(int)   * 1. 데이터 타입   >>> np.int64  >>> np.float32  >>> np.complex  >>> np.bool  >>> np.object  >>> np.string\_  >>> np.unicode\_   * 1. Array 조작   >>> i = np.transpose(b)  >>> i. T  >>> b.ravel()  >>> g.reshape(3, 2)  >>> h.resize((2,6))  >>> np.append( h,g)  >>> np.insert(a, 1, 5)  >>> np.delete(a,[1])  >>> np.concatenate((a,d),axis= 0) array( [ 1, 2, 3, 10, 15, 20])  >>> np.vstac k((a,b)) array([[ 1. , 2. , 3. ], [ 1 . 5, 2. , 3. ], [ 4. , 5. , 6. ]]) >>> np.r\_ [e,f ] >>> np. hstack((e,f)) array([[ 7., 7., 1., 0 . ], [ 7., 7., 0., 1 .]])  >>> np.column\_stac k((a,d)) array([[ 1, 10], [ 2, 15], [ 3, 20]])  >>> np.c\_ [a,d ] |
| 오후 | 1. 데이터 프레임 생성    1. df = pd.DataFrame( {"a" : [4 ,5, 6], "b" : [7, 8, 9], "c" : [10, 11, 12]}, index = [1, 2, 3]) Specify values for each column.    2. df = pd.DataFrame( [[4, 7, 10], [5, 8, 11], [6, 9, 12]], index=[1, 2, 3], columns=['a', 'b', 'c']) Specify values for each row. a b    3. df = pd.DataFrame( {"a" : [4 ,5, 6], "b" : [7, 8, 9], "c" : [10, 11, 12]}, index = pd.MultiIndex.from\_tuples( [('d',1),('d',2),('e',2)], names=['n','v']))  Create DataFrame with a MultiIndex 2. 데이터 프레임 리쉐이프    1. df.sort\_values('mpg') Order rows by values of a column (low to high). df.sort\_values('mpg',ascending=False) Order rows by values of a column (high to low).    2. df.rename(columns = {'y':'year'}) Rename the columns of a DataFrame df.sort\_index() Sort the index of a DataFrame    3. df.reset\_index() Reset index of DataFrame to row numbers, moving index to columns.    4. df.drop(columns=['Length','Height']) Drop columns from DataFrame 3. 서브셋 관측치 처리    1. df[df.Length > 7] Extract rows that meet logical criteria.    2. df.drop\_duplicates() Remove duplicate rows (only considers columns). df.head(n) Select first n rows.    3. df.tail(n) Select last n rows    4. df.iloc[10:20] Select rows by position.    5. df.loc[:,'x2':'x4'] Select all columns between x2 and x4 (inclusive). df.iloc[:,[1,2,5]] Select columns in positions 1, 2 and 5 (first column is 0). df.loc[df['a'] > 10, ['a','c']] Select rows meeting logical condition, and only the specific columns |