	The brocket method  data = {     "beer_data": beers,     "brewery_data": breweries }  The dict() method, option 1  tuple_list = [     "brewery_data", breweries),     ("brewery_data", breweries),     ("beer_data", beers) }
In [1]:	Key Value Key Value data3 = dict(tuple_list)  Creating Dictionary  # Membuat dictionary kosong
Out[1]:	<pre>mydict = {} mydict  {}  # dictionary dengan kunci integer</pre>
Out[2]: In [3]:	<pre># dictionary dengan kunci campuran mydict = {'warna':'merah', 1:[2, 3, 5]}</pre>
Out[3]:	<pre>mydict {'warna': 'merah', 1: [2, 3, 5]}  # membuat dictionary menggunakan fungsi dict() my_dict = dict([(1, 'sepatu'), (2, 'bola')]) my_dict</pre>
	{1: 'sepatu', 2: 'bola'}  Accesing Dictionary  # get vs [] for retrieving elements
In [6]:	<pre>mydict = {'name': 'Jack', 'age':26}  # Output: Jack print(mydict['name'])  Jack  # Output: 26</pre>
In [8]:	<pre>print(mydict.get('age')) 26</pre>
In [9]:	<pre>print(my_dict['address', 'age']) </pre>
	<pre><ipython-input-9-6335214169b8> in <module></module></ipython-input-9-6335214169b8></pre>
In [11]:	<pre>my_dict['age'] = 27</pre>
In [12]: In [13]:	<pre>print(my_dict) {'name': 'Jack', 'age': 27}</pre>
In [14]: In [15]:	<pre>print(my_dict) {'name': 'Jack', 'age': 27, 'address': 'Downtown'}  a_dict = {'a':1, 'b':2, 'c':3}</pre>
	<pre>new_key = 'A' old_key = 'a'  a_dict[new_key] = a_dict.pop(old_key)  print(a_dict) {'b': 2, 'c': 3, 'A': 1}</pre>
In [19]:	Removing Dictionary  # Removing elements from a dictionary  # Create a dictionary
In [20]:	<pre>squares = {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}  # Remove a particular item, returns its value # Output: 16 print(squares.pop(4))  # Output: {1: 1, 2: 4, 3: 9, 5: 25} print(squares)</pre>
In [21]:	16 {1: 1, 2: 4, 3: 9, 5: 25}
In [22]:	(5, 25)
In [23]:	{1: 1, 2: 4, 3: 9}
	NameError  Traceback (most recent call last) <ipython-input-23-dfc3cc2138c1> in <module>  1  # delete the dictionary itself&gt; 2 print(squares)  3  # Throws Error  NameError: name 'squares' is not defined</module></ipython-input-23-dfc3cc2138c1>
	Method  Clear()  Removes all items from the dictionary.  Copy()  Returns a shallow copy of the dictionary.  Returns a new dictionary with keys from sea, and value equal to v
	fromkeys(seq[, v])  Returns a new dictionary with keys from seq and value equal to v (defaults to None).  Returns the value of the key. If the key does not exist, returns d (defaults to None).  items()  Return a new object of the dictionary's items in (key, value) format.
	Returns a new object of the dictionary's keys.  Removes the item with the key and returns its value or d if key is not found. If d is not provided and the key is not found, it raises  KeyError.
	popitem()  Removes and returns an arbitrary item (key, value). Raises KeyError if the dictionary is empty.  Returns the corresponding value if the key is in the dictionary. If not, inserts the key with a value of d and returns d (defaults to None).
	update([other])  Updates the dictionary with the key/value pairs from other, overwriting existing keys.  values()  Returns a new object of the dictionary's values  Understanding Pandas
In [24]:	Pandas adalah sebuah library di Python yang berlisensi BSD dan open source yang menyediakan struktur data dan analisis data yang mudah digunakan Struktur data dasar pada Pandas dinamakan DataFrame, yang memudahkan kita untuk membaca sebuah file dengan banyak jenis format seperti file .txt, . csv, dan .tsv.  # Import pandas
111 [24].	import pandas as pd  Dataframe
	Series Series DataFrame  apples oranges apples oranges
	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$
In [25]:	3 1 3 2
In [26]: Out[26]:	<pre>data = [1, 2, 3, 4, 5] df = pd.DataFrame(data) df</pre>
	<ul> <li>0 1</li> <li>1 2</li> <li>2 3</li> <li>3 4</li> </ul>
In [27]:	<pre>import pandas as pd data = [['Alex', 10], ['Bob', 12], ['Clarke', 13]] df = pd.DataFrame(data, columns = ['Name', 'Age']) df</pre>
Out[27]:	Name         Age           0         Alex         10           1         Bob         12           2         Clarke         13
In [28]: Out[28]:	<pre>data = [['Alex', 10], ['Bob', 12], ['Clarke', 13]] df = pd.DataFrame(data, columns = ['Name', 'Age'], dtype = float) df</pre>
	<ul> <li>O Alex 10.0</li> <li>1 Bob 12.0</li> <li>2 Clarke 13.0</li> </ul>
	<pre>Create a DataFrame from Dict of ndarrays / Lists  import pandas as pd data = {'Name': ['Tom', 'Jack', 'Steve', 'Ricky'], 'Age': [28, 24, 29, 27]} df = pd.DataFrame(data) print(df)</pre>
	Name Age  0 Tom 28  1 Jack 24  2 Steve 29  3 Ricky 27   Create a DataFrame from List of Dicts
In [30]:	<pre>#With two column indices, values same as dictionary keys data = [{'a' : 1, 'b' : 2}, {'a': 5, 'b': 10, 'c': 20}]</pre>
	<pre>#With two column indices with one index with other name df1 = pd.DataFrame(data, index = ['first', 'second'], columns = ['a', 'b', 'c']) print(df1)</pre>
In [31]:	<pre>df1 = pd.DataFrame(data, index = ['first', 'second'], columns = ['a', 'b', 'c']) print(df1)  a b c first 1 2 NaN second 5 10 20.0  # import pandas as pd import pandas as pd</pre> import pandas as pd
In [31]:	<pre>df1 = pd.DataFrame(data, index = ['first', 'second'], columns = ['a', 'b', 'c']) print(df1)</pre>
In [31]:	df1 = pd.DataFrame(data, index = ['first', 'second'], columns = ['a', 'b', 'c']) print(df1)  a b c first 1 2 NaN second 5 10 20.0  # import pandas as pd import pandas as pd import pandas as pd  # df = pd.read_csv ('file_name.csv') df = pd.read_csv('nba.csv')  # print(df)  Name Team Number Position Age Height Weight \ 0 Avery Bradley Boston Celtics 0.0 PG 25.0 6-2 180.0 1 Jae Crowder Boston Celtics 99.0 SF 25.0 6-6 235.0 2 John Holland Boston Celtics 30.0 SG 27.0 6-5 205.0 3 R.J. Hunter Boston Celtics 30.0 SG 27.0 6-5 185.0 4 Jonas Jerebko Boston Celtics 8.0 PF 29.0 6-10 231.0 453 Shelvin Mack Utah Jazz 8.0 PG 26.0 6-3 203.0 454 Raul Neto Utah Jazz 25.0 PG 24.0 6-1 179.0 455 Tibor Pleiss Utah Jazz 21.0 C 26.0 7-3 256.0
In [31]:	df1 = pd.DataFrame(data, index = ['first', 'second'], columns = ['a', 'b', 'c']) print(df1)  a b c first 1 2 NaN second 5 10 20.0  # import pandas as pd import pandas as pd import pandas as pd  # df = pd.read_csv ('file_name.csv') df = pd.read_csv('nba.csv')  # print(df)  Name
In [31]:	dfl = pd.DataFrame(data, index = ['first', 'second'], columns = ['a', 'b', 'c']) print(dfl)  a b c first 1 2 NaN second 5 10 20.0  # import pandas as pd import pandas as pd import pandas as pd  # df = pd.read_csv ('file_name.csv') df = pd.read_csv ('rfile_name.csv') print(df)  Name  Name  Name  Name  Name  Name  Number Position  Age Height  5.0 6-2 188.0  2 John Holland Boston Celtics 9.0 FG 25.0 6-6 225.0  3 R.J. Hunter Boston Celtics 30.0 SG 27.0 6-5 205.0  3 R.J. Hunter Boston Celtics 8.0 FG 22.0 6-5 185.0  4 Jonas Jerebko Boston Celtics 8.0 FG 29.0 6-10 231.0  1  453 Shelvin Mack Utah Jazz 8.0 FG 26.0 6-3 203.0 454 Agul Neto Utah Jazz 21.0 C 26.0 7-3 256.0 455 Jibor Pleiss Utah Jazz 24.0 C 26.0 7-3 256.0  College Salary NaN
In [31]:	dfl = pd.bataFrame(data, index = ['first', 'second'], columns = ['a', 'b', 'c'])  print(dfl)  a b c first 1 2 NaN second 5 10 20.0   # import pandas as pd import pandas as pd import pandas as pd import pandas as pd  # off = pd.read_csv ('file_name.csv') df = pd.read_csv('rba.csv')  # print(df)  Name  Team Number Position Age Height Weight \ 0 Avery Bradley Boston Celtics 90.0 PG 25.0 6-2 180.0 1 Jae Crowder Boston Celtics 99.0 SF 25.0 6-2 235.0 2 John Boiland Boston Celtics 30.0 SG 27.0 6-5 205.0 3 R.J. Hunter Boston Celtics 8.0 SF 25.0 6-6 235.0 4 Jonas Derebko Boston Celtics 8.0 FF 29.0 6-10 231.0
	dfl = pd.DataFrame(data, index = ['first', 'second'], columns = ['a', 'b', 'c'])  print(dfl)  # a b c first 1 2 NaN second 5 10 20.0  # import pandas as pd import pandas as pd import pandas as pd  # df = pd.read_csv ('file_name.csv')  # print(df)    Name
	### and DetaFrame(data, index = ('first', 'second'), columns = ('a', 'b', 'c'))  ### print(dIT)  ### a b c  ### print(dIT)  ### print(dIT)    Name
In [32]:	disposition
In [32]: In [34]:	disposition
In [32]: In [34]:	### ### #### #########################
In [32]:  In [34]:  In [35]:	dri
<pre>In [32]: In [34]: Out[34]: Out[35]:</pre>	Second   1   2   2   2   2   2   2   2   2   2
In [34]:  In [34]:  Out [34]:  Out [35]:	
In [32]:  In [34]:  Out [34]:  Out [36]:  In [37]:	### ### ##############################
In [34]:  In [34]:  Out [34]:  Out [35]:  Out [37]:	### ### ### ### #### #### ############
In [34]:  In [34]:  Out [34]:  Out [35]:  Out [37]:	### Company   Proceedings   Control   Control
In [34]:  In [35]:  Out [36]:  Out [37]:  Out [37]:	### Common Commo
In [34]:  In [35]:  Out [36]:  Out [37]:  Out [37]:	### 1995 Answeringstown, Today in "Internation, Assembly, Assembly, Principles (1995)    Total
In [34]:  Out [34]:  Out [36]:  Out [37]:  Out [37]:	### Company of the Co
In [34]:  In [37]:  Out [36]:  Out [37]:  Out [37]:  Out [37]:	### STATE OF THE PROPERTY OF T
In [34]:  Out [34]:  Out [36]:  Out [37]:  Out [37]:  Out [39]:	### Command and Co
In [34]:  In [37]:  Out [37]:  Out [37]:  Out [37]:  Out [37]:  Out [40]:  In [40]:  Out [40]:	### Property of the Company of the C
In [34]:  Out [34]:  Out [37]:  Out [37]:  Out [37]:  Out [37]:  Out [37]:  Out [40]:  In [41]:  Out [42]:  Out [42]:  Out [43]:	April   Apri
In [34]:  Out [34]:  Out [37]:  Out [37]:  Out [37]:  Out [37]:  Out [40]:  In [40]:  Out [40]:  Out [40]:  Out [40]:	
In [34]:  Out [34]:  Out [37]:  Out [37]:  Out [37]:  Out [37]:  Out [40]:  In [41]:  Out [41]:  Out [41]:  Out [41]:	
In [34]:  Cut [34]:  Cut [37]:  In [37]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [40]:  In [41]:  Cut [42]:  Cut [42]:  Cut [42]:  Cut [42]:  Cut [42]:  Cut [43]:  Cut [44]:	### Common Commo
In [34]:  Cut [34]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [41]:  Cut [42]:  In [42]:  Cut [42]:  Cut [42]:  Cut [43]:  Cut [44]:	### PROPRIES OF THE PROPERTY O
In [34]:  Cut [34]:  Cut [37]:  In [37]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [40]:  In [41]:  Cut [42]:  Cut [42]:  Cut [42]:  Cut [42]:  Cut [42]:  Cut [43]:  Cut [44]:	### Compare Andrew Compare Com
In [34]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [41]:  Cut [41]:  Cut [42]:  In [42]:  Cut [43]:  Cut [43]:  Cut [44]:	### Company of the Co
In [34]:  Cut [34]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [41]:  Cut [42]:  In [42]:  Cut [42]:  Cut [43]:  Cut [43]:  Cut [43]:  Cut [44]:	### A PACK STORY OF THE PACK S
In [33]:  In [34]:  Cut [34]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [41]:  Cut [42]:  Cut [42]:  Cut [42]:  Cut [43]:  Cut [44]:	### Comparison of the Comparis
In [33]:  In [34]:  Cut [34]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [41]:  Cut [42]:  In [42]:  Cut [43]:  Cut [43]:  Cut [43]:  Cut [43]:  Cut [44]:  Cut [47]:  Cut	Security of the control of the contr
In [33]:  In [34]:  Cut [34]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [41]:  Cut [42]:  In [42]:  Cut [43]:  Cut [43]:  Cut [43]:  Cut [43]:  Cut [44]:  Cut [47]:  Cut	### Part
In [34]:  Cut [34]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [40]:  Cut [41]:  Cut [42]:  Cut [42]:  Cut [42]:  Cut [42]:  Cut [42]:  Cut [43]:  Cut [43]:  Cut [43]:  Cut [43]:  Cut [43]:  Cut [44]:  Cut [47]:  Cut [50]:  Cut [50]:  Cut [50]:  Cut [51]:	### Part
In [34]:  Cut [37]:  Cut [40]:  In [41]:  Cut [41]:  Cut [42]:  In [42]:  Cut	### 1998
In [34]:  Cut [34]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [37]:  Cut [40]:  In [41]:  Cut [41]:  Cut [41]:  Cut [41]:  Cut [42]:  In [42]:  Cut [43]:  Cut [44]:  Cut	### Company of the Co