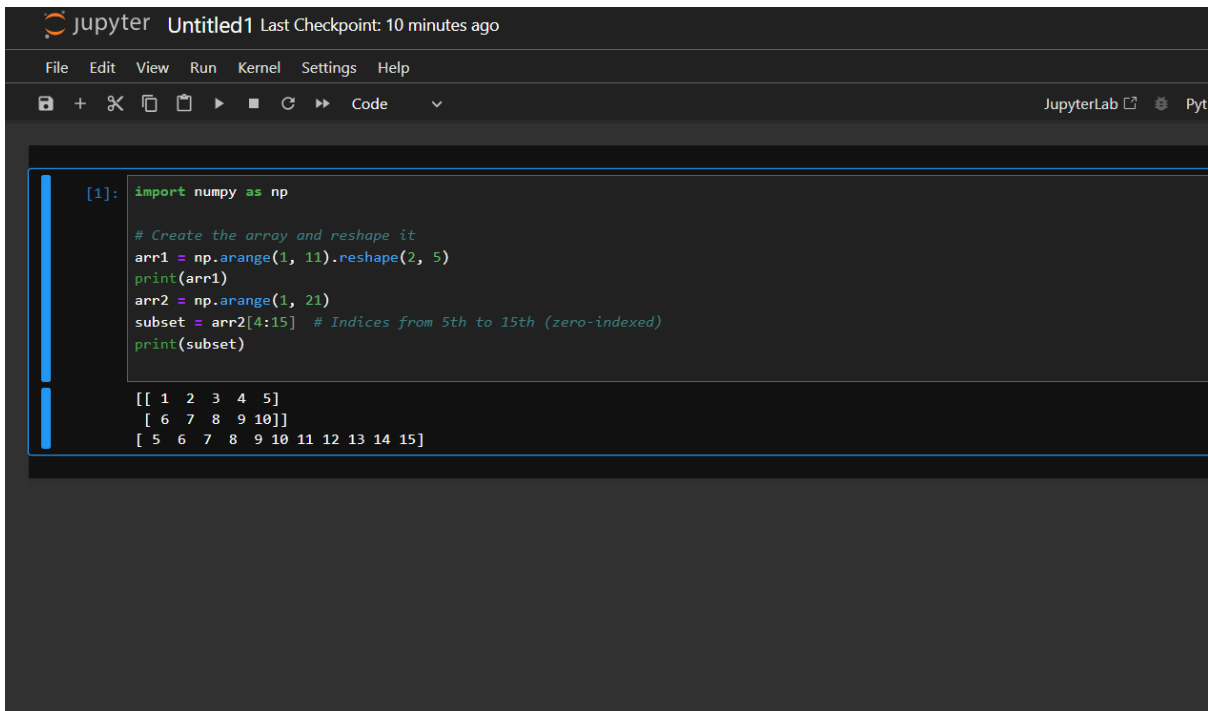


ASSIGNMENT_7 NUMPY



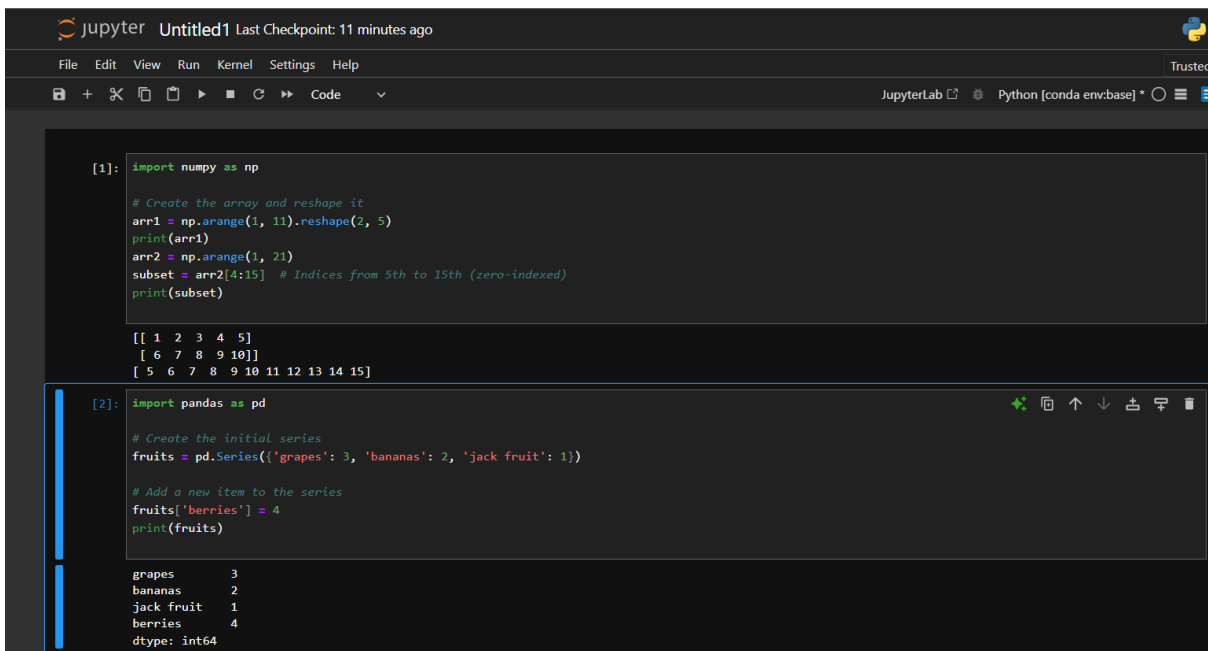
The image shows a JupyterLab interface with a code cell. The code imports NumPy as np, creates an array arr1 of size (2, 5) with values from 1 to 11, and prints it. Then it creates an array arr2 of size (1, 21) with values from 1 to 21, and prints a subset of arr2 from index 4 to 15 (zero-indexed). The output shows the printed arrays.

```
[1]: import numpy as np

# Create the array and reshape it
arr1 = np.arange(1, 11).reshape(2, 5)
print(arr1)
arr2 = np.arange(1, 21)
subset = arr2[4:15] # Indices from 5th to 15th (zero-indexed)
print(subset)
```

```
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]]
[ 5  6  7  8  9 10 11 12 13 14 15]
```

2.



The image shows a JupyterLab interface with two code cells. The first cell is identical to the one in the first image. The second cell imports Pandas as pd, creates a Series named fruits with values for 'grapes' (3), 'bananas' (2), and 'jack fruit' (1), and prints it. Then it adds a new item 'berries' with a value of 4 and prints the updated Series. The output shows the printed Series.

```
[1]: import numpy as np

# Create the array and reshape it
arr1 = np.arange(1, 11).reshape(2, 5)
print(arr1)
arr2 = np.arange(1, 21)
subset = arr2[4:15] # Indices from 5th to 15th (zero-indexed)
print(subset)
```

```
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]]
[ 5  6  7  8  9 10 11 12 13 14 15]
```

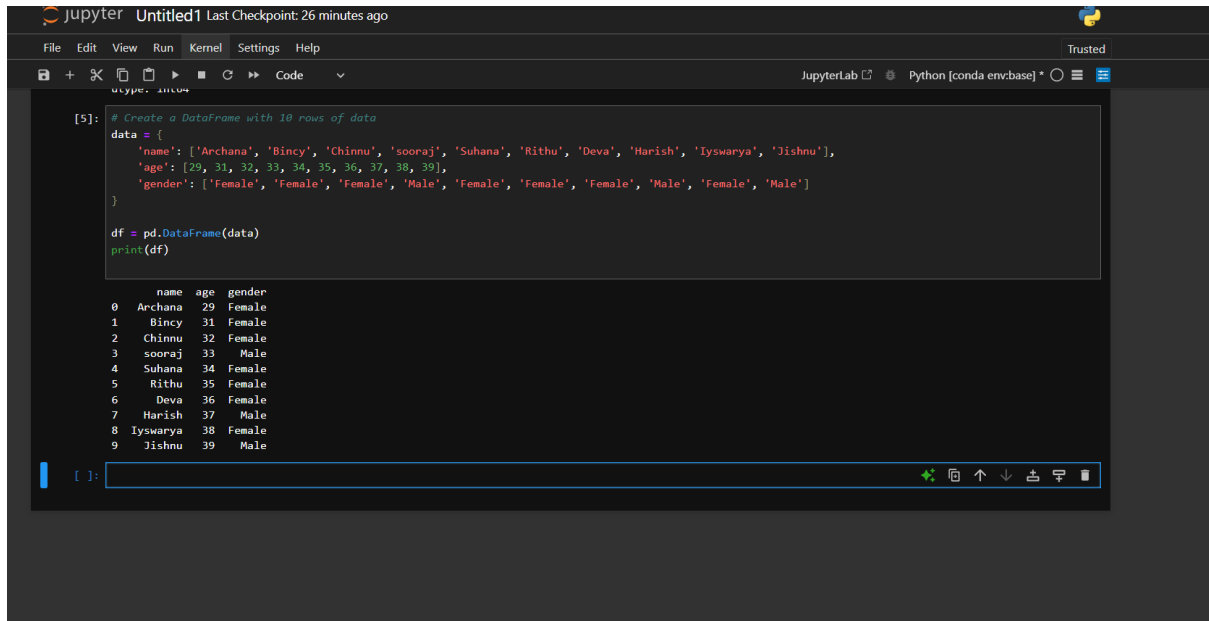
```
[2]: import pandas as pd

# Create the initial series
fruits = pd.Series({'grapes': 3, 'bananas': 2, 'jack fruit': 1})

# Add a new item to the series
fruits['berries'] = 4
print(fruits)
```

```
grapes    3
bananas   2
jack fruit 1
berries    4
dtype: int64
```

3.



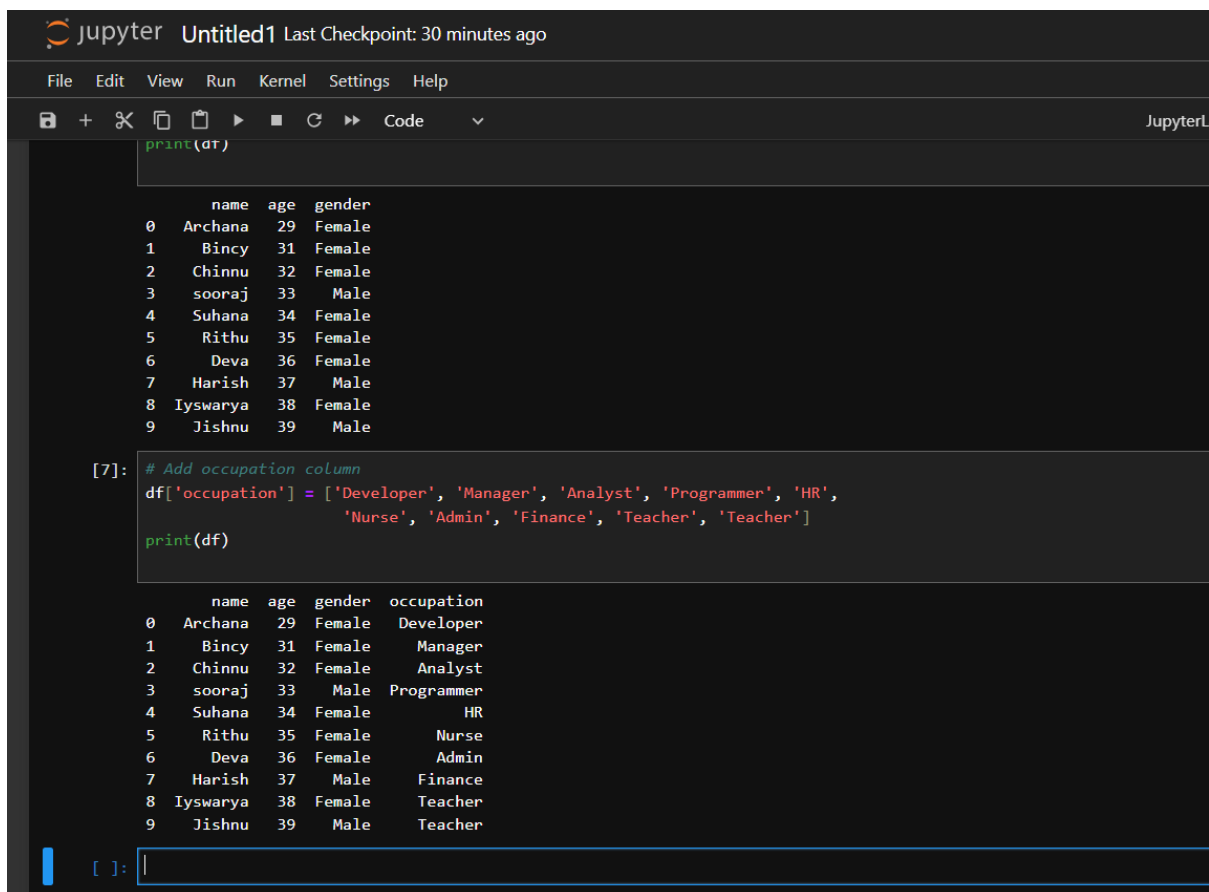
The screenshot shows a JupyterLab interface with a code editor and a console output. The code in the editor creates a DataFrame with 10 rows of data. The output in the console shows the DataFrame structure and its contents.

```
[5]: # Create a DataFrame with 10 rows of data
data = {
    'name': ['Archana', 'Bincy', 'Chinnu', 'sooraj', 'Suhana', 'Rithu', 'Deva', 'Harish', 'Iyswarya', 'Jishnu'],
    'age': [29, 31, 32, 33, 34, 35, 36, 37, 38, 39],
    'gender': ['Female', 'Female', 'Female', 'Male', 'Female', 'Female', 'Female', 'Male', 'Female', 'Male']
}

df = pd.DataFrame(data)
print(df)
```

	name	age	gender
0	Archana	29	Female
1	Bincy	31	Female
2	Chinnu	32	Female
3	sooraj	33	Male
4	Suhana	34	Female
5	Rithu	35	Female
6	Deva	36	Female
7	Harish	37	Male
8	Iyswarya	38	Female
9	Jishnu	39	Male

4.



The screenshot shows a JupyterLab interface with a code editor and a console output. The code in the editor adds an 'occupation' column to the DataFrame. The output in the console shows the DataFrame structure and its contents.

```
[7]: # Add occupation column
df['occupation'] = ['Developer', 'Manager', 'Analyst', 'Programmer', 'HR',
                  'Nurse', 'Admin', 'Finance', 'Teacher', 'Teacher']
print(df)
```

	name	age	gender	occupation
0	Archana	29	Female	Developer
1	Bincy	31	Female	Manager
2	Chinnu	32	Female	Analyst
3	sooraj	33	Male	Programmer
4	Suhana	34	Female	HR
5	Rithu	35	Female	Nurse
6	Deva	36	Female	Admin
7	Harish	37	Male	Finance
8	Iyswarya	38	Female	Teacher
9	Jishnu	39	Male	Teacher

5.

The image shows a JupyterLab interface with a code cell containing the following Python code:

```
print(filtered_df)

[11]: # Save the DataFrame to a CSV file
df.to_csv('data.csv', index=False)

# Read the CSV file
df_from_csv = pd.read_csv('data.csv')
print(df_from_csv)
```

The output of the code is a table of employee data:

	name	age	gender	occupation
1	Bincy	31	Female	Manager
2	Chinnu	32	Female	Analyst
3	sooraj	33	Male	Programmer
4	Suhana	34	Female	HR
5	Rithu	35	Female	Nurse
6	Deva	36	Female	Admin
7	Harish	37	Male	Finance
8	Iyswarya	38	Female	Teacher
9	Jishnu	39	Male	Teacher

The JupyterLab interface also shows the file menu, edit menu, view menu, run menu, kernel menu, settings menu, and help menu. The status bar at the bottom indicates the current environment is Python [conda env:base] and the file is named Untitled1.