V2_using_pandas

August 16, 2025

1 Activity: Using Pandas

1.1 Introduction

In this activity you will practice using some of the basic functionality associated with Pandas. This activity will cover the following topics: - Creating a `DataFrame` - Displaying `DataFrame` information - Accessing column data - Getting ranges of column data - Creating Series from dictionaries - Using the `iloc()` and `loc()` methods - Getting data from multiple columns - Getting rows

Question 1 Create a DataFrame called `df` from the given CSV file `student_data.csv` then using the `df` DataFrame, assign the `Name` column to a `Series` called `names`.

[18]:
```
import pandas as pd
df = pd.read_csv("student_data.csv")
names = df["Name"]
print(df.head())
print(names.head())
```

```
              Name  Age  Math Grade  English Grade
0  Jennifer Jackson   14          84             81
1   Michael Johnson   14          92             85
2        Robert Lee   18          87             80
3      Linda Harris   13          90             77
4     Michael Moore   18          88             99
0    Jennifer Jackson
1     Michael Johnson
2          Robert Lee
3        Linda Harris
4       Michael Moore
Name: Name, dtype: object
```

[9]:
```
# Question 1 Grading Checks

assert isinstance(df, pd.DataFrame), "Did you create a DataFrame called df?"
assert isinstance(names, pd.Series), "Did you assign the Name column in a␣
 ↪variable called names?"
```

Question 2 Using the `df` DataFrame, assign the `Age` and `Math Grade` columns to a DataFrame called `age_math`.

[13]:
```
age_math = df[["Age", "Math Grade"]]

print(age_math.head())
```

```
   Age  Math Grade
0   14          84
1   14          92
2   18          87
3   13          90
4   18          88
```

[ ]:
```
# Question 2 Grading Checks

assert isinstance(age_math, pd.DataFrame), "Did you assign the Age and Math␣
↪Grade columns to a variable called age_math?"
```

Question 3 Using the `.loc()` method, assign the `Age` and `Math Grade` columns for the first 30 rows of `df` to a variable called `first_thirty_loc`.

[16]:
```
first_thirty_loc = df.loc[0:29, ["Age", "Math Grade"]]

print(first_thirty_loc.head())
```

```
   Age  Math Grade
0   14          84
1   14          92
2   18          87
3   13          90
4   18          88
```

[ ]:
```
# Question 3 Grading Checks

assert isinstance(first_thirty_loc, pd.DataFrame), "Did you correctly assign␣
↪the first 30 rows to a variable called first_thirty_loc?"
```

Question 4 Get the even numbered rows from the `Name` and `English Grade` columns and assign the result to a variable called `even_rows_english`.

[17]:
```
even_rows_english = df.loc[::2, ["Name", "English Grade"]]
print(even_rows_english.head())
```

```
              Name  English Grade
0  Jennifer Jackson             81
2       Robert Lee             80
```

```
4     Michael Moore             99
6       Sarah Smith             90
8     Linda Jackson             79
```

[ ]:
```
# Question 4 Grading Checks

assert isinstance(even_rows_english, pd.DataFrame), "Did you correctly assign␣
↪even numbered rows to a variable called even_rows_english? Hint: the first␣
↪row is index 0."
```