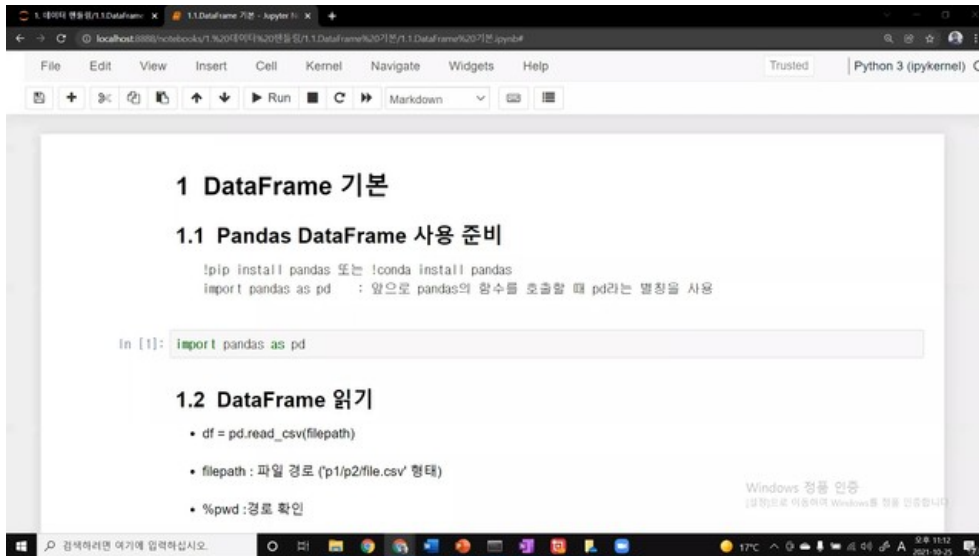


02

## DataFrame 기본

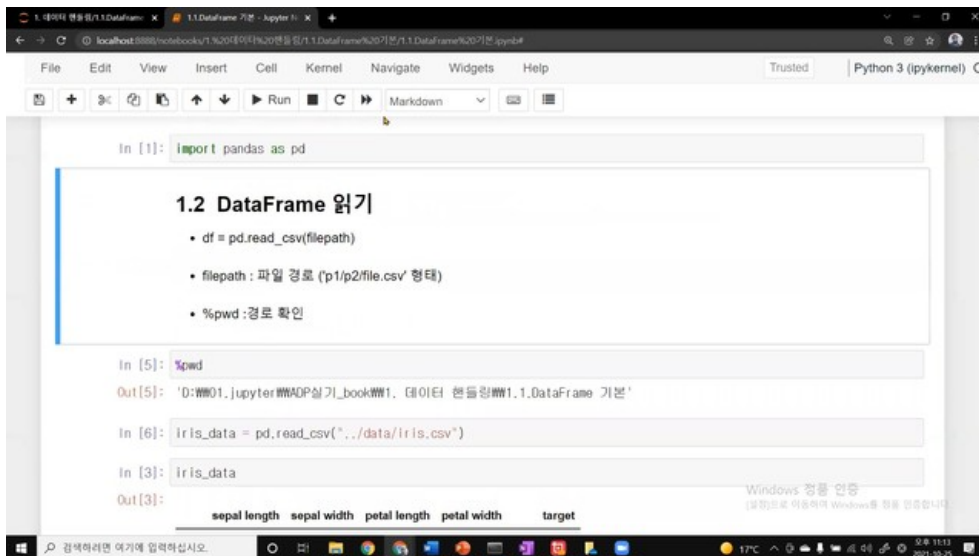
### < DataFrame 사용 준비 >

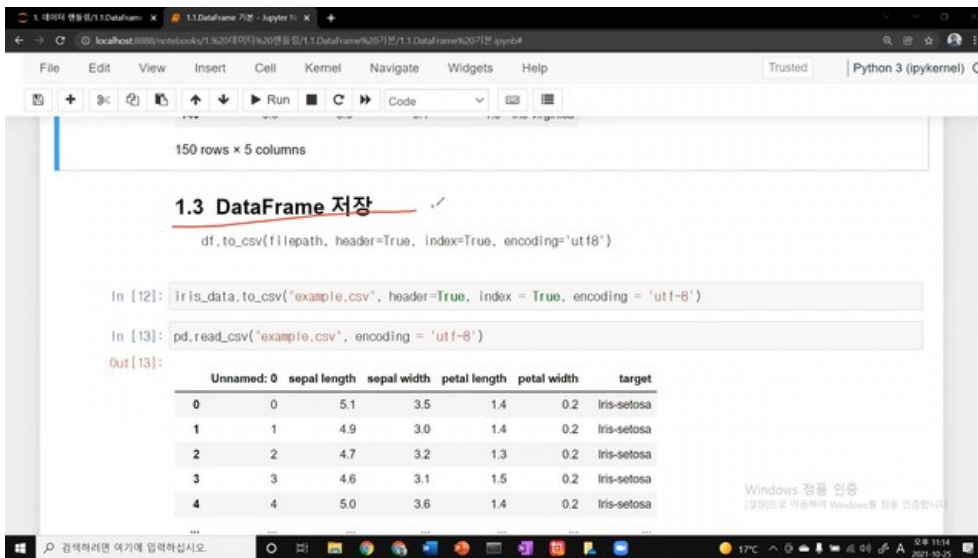
0:10



### < DataFrame 읽기 >

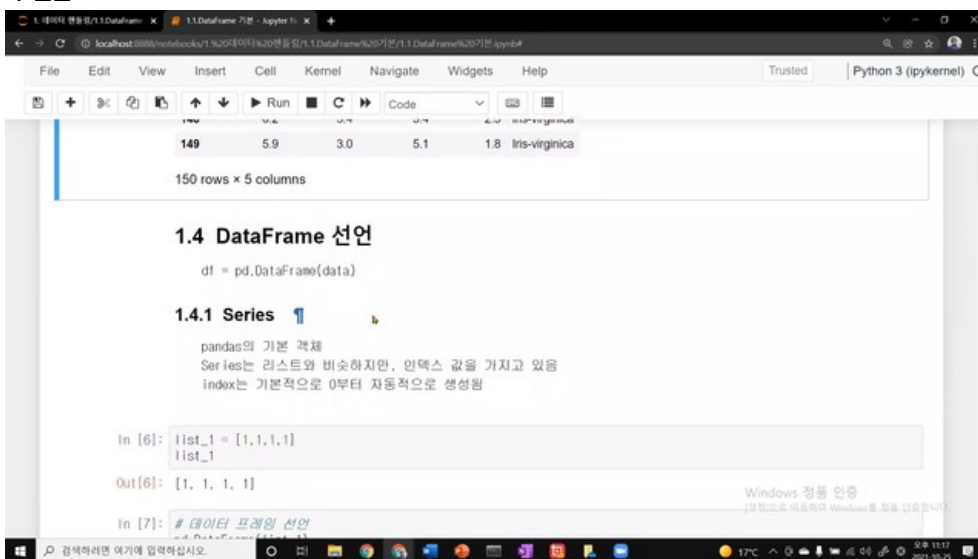
0:49





## < DataFrame 선언, Series 설명 >

4:22



## < DataFrame 출력>

5:48

```

In [9]: ## 인덱스와 값이 선언하기
Series_2 = pd.Series([1, 1, 1, 1], index=['2021-09-15', '2', 'dd', '43'])

In [10]: Series_2
Out[10]:
2021-09-15    1
2             1
dd            1
43            1
dtype: int64

1.5 DataFrame 출력

df.head(n)    # 처음 n개 출력, 미지정시 5개
df.tail(n)    # 마지막 n개 출력, 미지정시 5개

In [11]: iris_data.tail()
Out[11]:
   sepal length  sepal width  petal length  petal width  target
145           6.7         3.0          5.2          2.3  Iris-virginica
146           6.3         2.5          5.0          1.9  Iris-virginica
147           6.5         3.0          5.2          2.0  Iris-virginica
148           6.2         3.4          5.4          2.3  Iris-virginica
149           5.9         3.0          5.1          1.8  Iris-virginica

```

## < DataFrame 요약>

# 데이터 요약은 EDA와 연결되니 외워주세요

요약을 보고 해석할 수 있어야 합니다.

6:23

```

df.tail(n)    # 마지막 n개 출력, 미지정시 5개

In [14]: iris_data.tail()
Out[14]:
   sepal length  sepal width  petal length  petal width  target
145           6.7         3.0          5.2          2.3  Iris-virginica
146           6.3         2.5          5.0          1.9  Iris-virginica
147           6.5         3.0          5.2          2.0  Iris-virginica
148           6.2         3.4          5.4          2.3  Iris-virginica
149           5.9         3.0          5.1          1.8  Iris-virginica

1.6 DataFrame 요약 / 통계정보 확인

• df.info() # 총 데이터 건수와 데이터 타입, Null건수 확인
• df.describe() # 숫자형 컬럼들의 n-percentile분포도, 평균값, 최댓값, 최솟값 확인
• df['컬럼명'].min() 또는 df['컬럼명'].max() # 해당 컬럼의 min, max 값을 확인

```

## < DataFrame 인덱스 확인, 추가, 리셋>

# 데이터 프레임의 인덱스를 이해하는 것이 중요합니다!

12:15

```

In [23]: iris_data['sepal_length'].shape
Out[23]: (150,)

In [26]: iris_data['target'].unique()
Out[26]: array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)

In [18]: iris_data['target'].count()
Out[18]: 150

In [19]: iris_data['target'].value_counts()
Out[19]: Iris-versicolor    50
Iris-virginica          50
Iris-setosa             50
Name: target, dtype: int64

```

**1.7 DataFrame 인덱스 확인, 추가, 리셋**

df.index # 인덱스 확인

< DataFrame value 값 확인>

14:56

```

Out[34]:
   target  sepal length  sepal width  petal length  petal width
0  Iris-setosa         5.1         3.5         1.4         0.2
1  Iris-setosa         4.9         3.0         1.4         0.2
2  Iris-setosa         4.7         3.2         1.3         0.2
3  Iris-setosa         4.6         3.1         1.5         0.2
4  Iris-setosa         5.0         3.6         1.4         0.2

```

**1.8 DataFrame Value값 확인**

```

In [26]: iris_data.values[:5]
Out[26]: array([[0, 'Iris-setosa', 5.1, 3.5, 1.4, 0.2],
 [1, 'Iris-setosa', 4.9, 3.0, 1.4, 0.2],
 [2, 'Iris-setosa', 4.7, 3.2, 1.3, 0.2],
 [3, 'Iris-setosa', 4.6, 3.1, 1.5, 0.2],
 [4, 'Iris-setosa', 5.0, 3.6, 1.4, 0.2]], dtype=object)

```

**1.9 DataFrame 컬럼명 확인, 변경**

< DataFrame 컬럼명 확인, 변경>

# 앞으로 컬럼명을 변경할 일이 많을 거예요 꼭 외우고 숙달해주세요

15:38

**1.9 DataFrame 컬럼명 확인, 변경**

```

df.columns # 컬럼명 확인
df.columns = ['new1', 'new2']
df.rename({'old1':'new1', 'old2':'new2'}, axis='columns', inplace=True)
df.columns = df.columns.str.replace('기존문자', '대체할 문자') # 컬럼명 특정문자 대체
df.drop(['컬럼명'], axis=1, inplace=True)

```

```

In [37]: iris_data.columns
Out[37]: Index(['target', 'sepal length', 'sepal width', 'petal length', 'petal width'], dtype='object')

In [28]: iris_data.columns = ['a', 'b', 'c', 'd', 'e']
iris_data.head()
Out[28]:

```

	a	b	c	d	e
0	Iris-setosa	5.1	3.5	1.4	0.2
1	Iris-setosa	4.9	3.0	1.4	0.2

## < DataFrame 컬럼(변수) 타입 변경 >

# 컬럼 = feature = 변수입니다. 타입을 변경하실 수 있어야 합니다.

20:52

```

In [47]: iris_data.columns = iris_data.columns.str.replace('%', '_') # 컬럼명 특정문자 대체
iris_data.head(1)
Out[47]:

```

	target	sepal_length	sepal_width	petal_length	petal_width
0	Iris-setosa	5.1	3.5	1.4	0.2

**1.10 DataFrame 컬럼 타입 변경**

```

df['컬럼명'] = df['컬럼명'].astype('타입') # 컬럼 타입 변경
df = df.convert_dtypes() 모든 컬럼에 대해 가능한 가장 적절한 dtypes로 변경

```

```

In [33]: iris_data.info()
Out[33]:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   target      150 non-null    object
1   sepal_length 150 non-null    float64

```

다음 시간에는 pandas를 이용하여 데이터 프레임에서 행, 열을 추가하고 삭제하는 방법에 대해서 배워보겠습니다

다음 강의에서 보요~

아래와 같이 실행하시면 pdf 파일이 깨지지 않고 저장됩니다~

과제 제출하실 때 참고하세요^^

c 드라이브 >> user >> 자기계정명 >> .jupyter >> nbconfig 폴더 생성 >> 텍스트파일 생성 >> 아래 코드 붙여넣기 >> notebook.json으로 저장

```
{  
  
  "MarkdownCell": {  
  
    "cm_config": {  
  
      "lineWrapping": true  
    }  
  },  
  
  "CodeCell": {  
  
    "cm_config": {  
  
      "lineWrapping": true  
    }  
  }  
}
```