

seaborn: statistical data visualization

- matplotlib를 기본으로 다양한 시각화 기법을 제공하는 라이브러리
- 예쁘게 그래프를 그려준다 + 시각화 했을 때 데이터 확인이 쉽다
- pandas 데이터프레임이랑 호환이 잘됨

In [1]:

```
# 라이브러리와 데이터 불러오기
import seaborn as sns
sns.set_theme(style='whitegrid')

pen = sns.load_dataset('penguins') # 데이터 불러오기
pen
```

Out[1]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0
3	Adelie	Torgersen	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0
...
339	Gentoo	Biscoe	NaN	NaN	NaN	NaN
340	Gentoo	Biscoe	46.8	14.3	215.0	4850.0
341	Gentoo	Biscoe	50.4	15.7	222.0	5750.0
342	Gentoo	Biscoe	45.2	14.8	212.0	5200.0
343	Gentoo	Biscoe	49.9	16.1	213.0	5400.0

344 rows x 7 columns

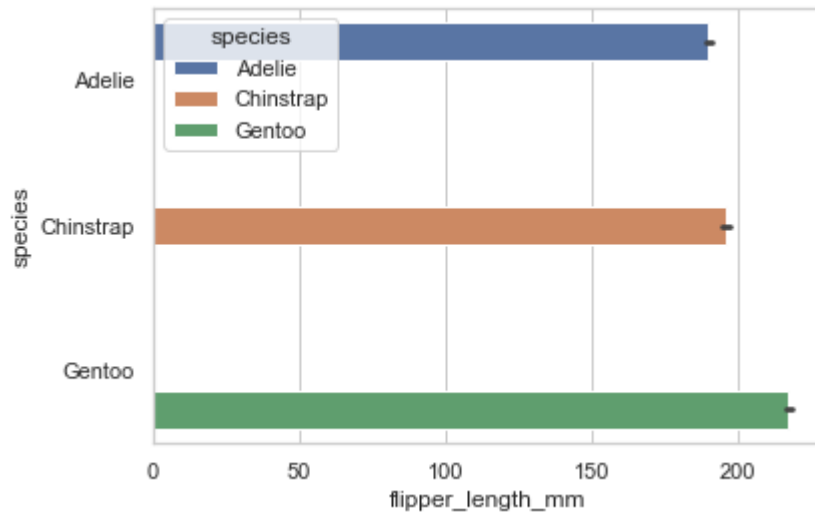
In [7]:

```
# 01. Barplot
# 어떤 데이터에 대한 값의 크기를 막대로 보여주는 plot

sns.barplot(data=pen, x='flipper_length_mm', y='species', hue='species') # hue
# sns.barplot(data=pen, y='flipper_length_mm', x='species', hue='species') #
```

Out[7]:

<AxesSubplot:xlabel='flipper_length_mm', ylabel='species'>

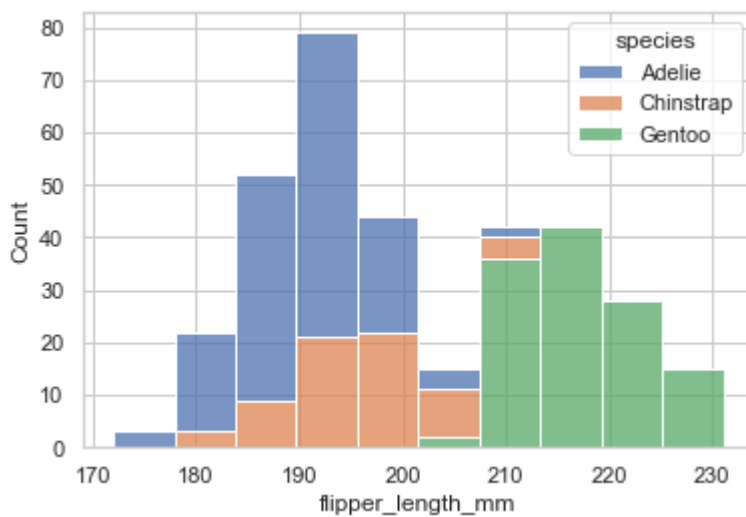


In [8]:

```
# 02. Histplot
# 가장 기본적으로 사용되는 히스토그램을 출력하는 plot
# 전체 데이터를 특정 구간별 정보를 확인할 때 사용함 (시계열 데이터를 확인할 때 좋음)

sns.histplot(data=pen, x='flipper_length_mm',
             hue='species',
             multiple='stack') # 그래프가 겹치지 않게 보여주는 옵션
```

Out[8]: <AxesSubplot:xlabel='flipper_length_mm', ylabel='Count'>

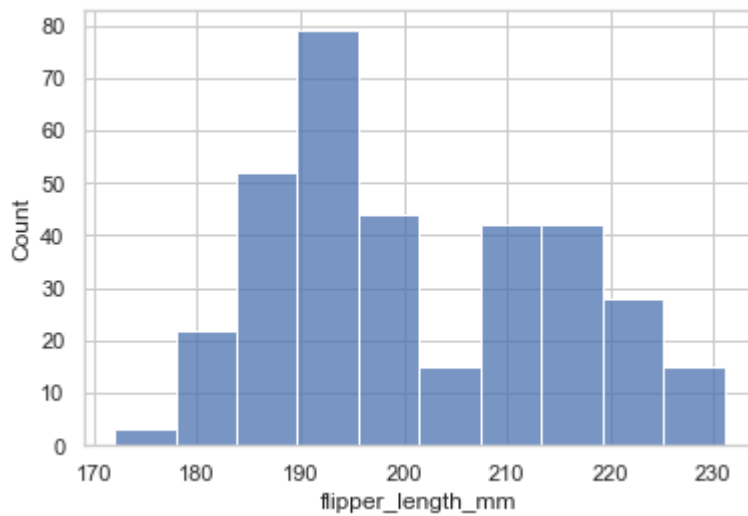


In [10]:

```
# 내부 도수를 분리시켜서 보여주는 거 같다

sns.histplot(data=pen, x='flipper_length_mm')
```

Out[10]: <AxesSubplot:xlabel='flipper_length_mm', ylabel='Count'>

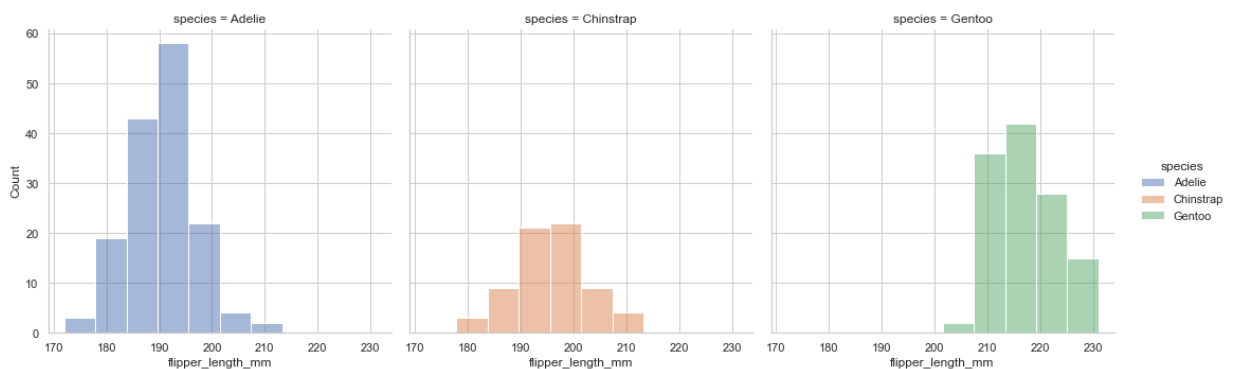


In [11]:

```
# 03. Displot
# distribution들을 여러 subplot들로 나눠서 출력해주는 plot.
# displot에 kind를 변경하는 것으로, histplot, kdeplot, ecdfplot 모두 출력 가능 (디폴트는 histplot)

sns.displot(data=pen, x='flipper_length_mm',
            hue='species',
            col='species') # 종류별로 그래프 만들어줌
```

Out[11]: <seaborn.axisgrid.FacetGrid at 0x7fbd896ff220>

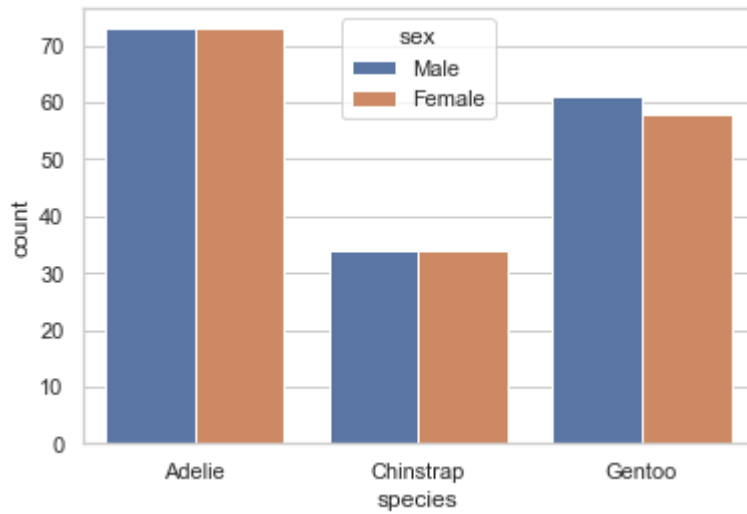


In [12]:

```
# 04. Countplot
# 범주형 속성을 가지는 데이터들의 histogram을 보여주는 plot.
# 종류별 count를 보여주는 방법

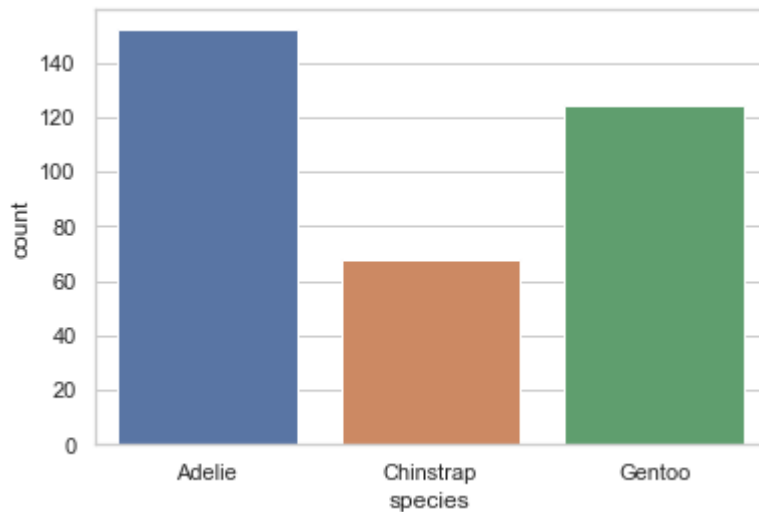
sns.countplot(data=pen, x='species', hue='sex')
```

Out[12]: <AxesSubplot:xlabel='species', ylabel='count'>



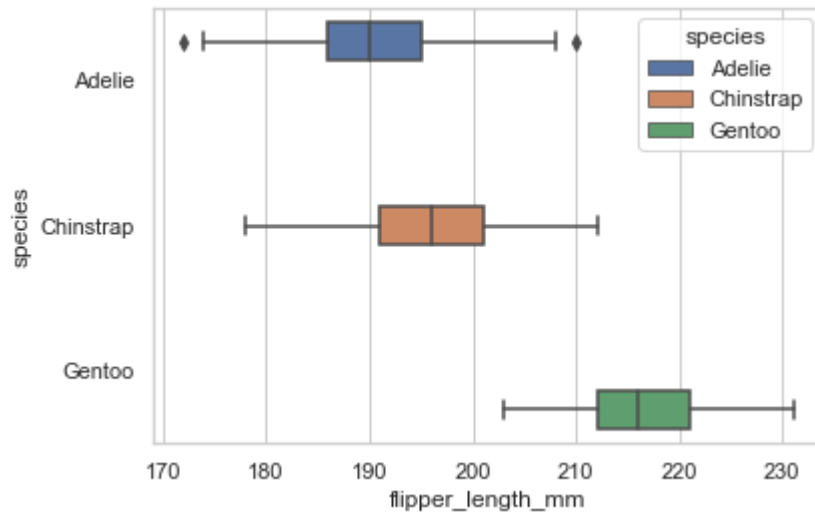
In [13]: `sns.countplot(data=pen, x='species')`

Out[13]: `<AxesSubplot:xlabel='species', ylabel='count'>`



In [14]: `# 05. Boxplot`
`# 데이터의 각 종류별로 사분위 수(quantile)를 표시하는 plot.`
`# 특정 데이터의 전체적인 분포를 확인하기 좋은 시각화 기법`
`# box와 전체 range의 그림을 통해 outlier를 찾기 쉬움 (IQR: Inter-Quantile Range)`
`sns.boxplot(data=pen, x='flipper_length_mm', y='species', hue='species')`
`# Adelie 선 좌우 점 처럼 선 밖의 점들을 통해 이상값 확인 가능`

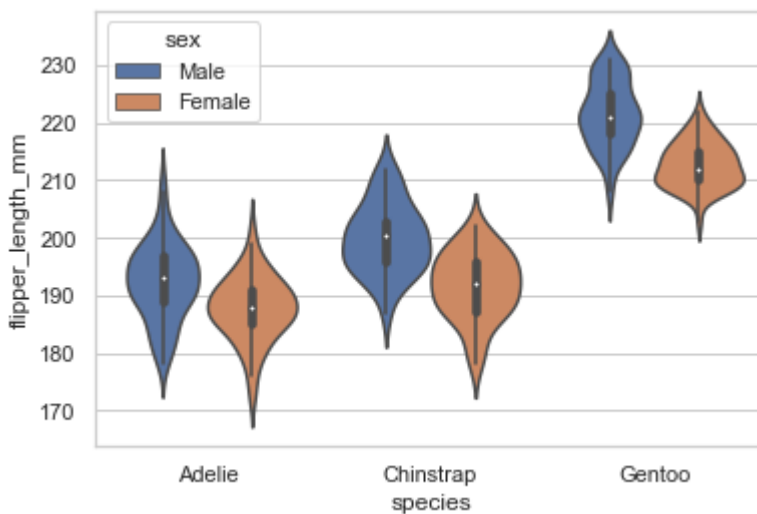
Out[14]: `<AxesSubplot:xlabel='flipper_length_mm', ylabel='species'>`



```
In [15]: # 06. Violinplot
# 데이터에 대한 분포 자체를 보여주는 plot.
# boxplot과 비슷하지만, 전체 분포에 대한 그림을 보여준다는 점에서 boxplot과 다르다

sns.violinplot(data=pen, y='flipper_length_mm', x='species', hue='sex')
```

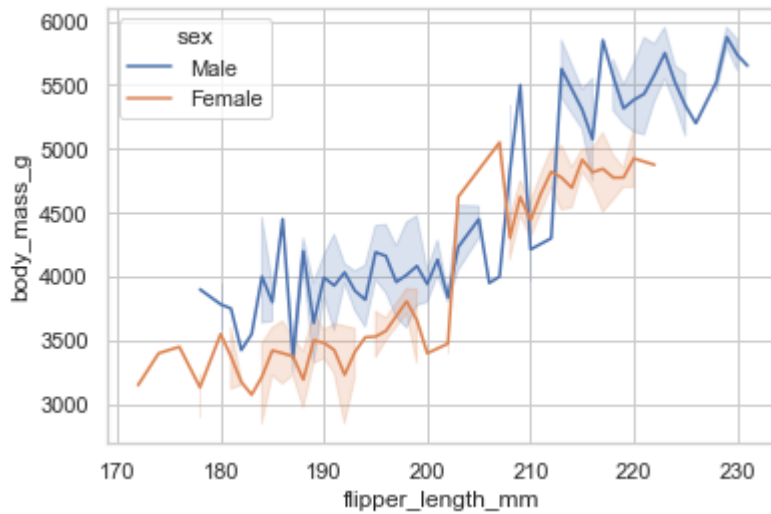
```
Out[15]: <AxesSubplot:xlabel='species', ylabel='flipper_length_mm'>
```



```
In [16]: # 07. Lineplot
# 특정 데이터를 x, y로 표시하여 관계를 확인할 수 있는 plot.
# 수치형 지표들 간의 경향을 파악할 때 많이 사용

sns.lineplot(data=pen, y='body_mass_g', x='flipper_length_mm', hue='sex')
```

```
Out[16]: <AxesSubplot:xlabel='flipper_length_mm', ylabel='body_mass_g'>
```



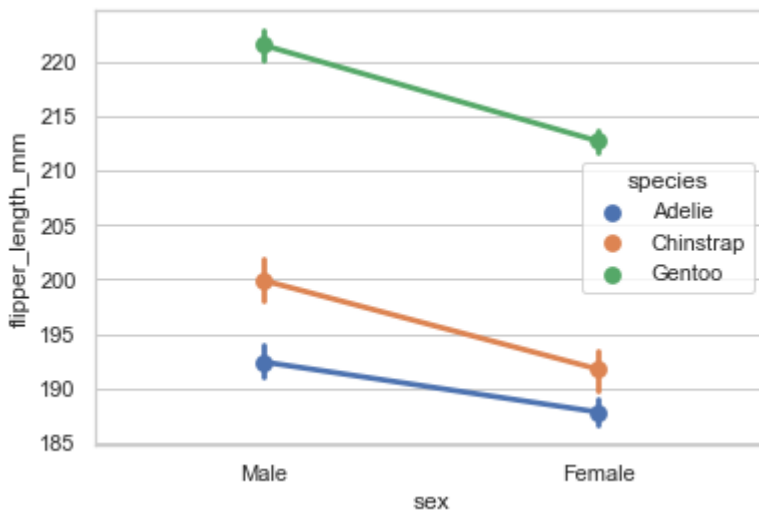
In [17]:

```
# 08. Pointplot
# 특정 수치 데이터를 error bar와 함께 출력해주는 plot.

sns.pointplot(data=pen, x='sex', y='flipper_length_mm', hue='species')
```

Out[17]:

```
<AxesSubplot:xlabel='sex', ylabel='flipper_length_mm'>
```



In [18]:

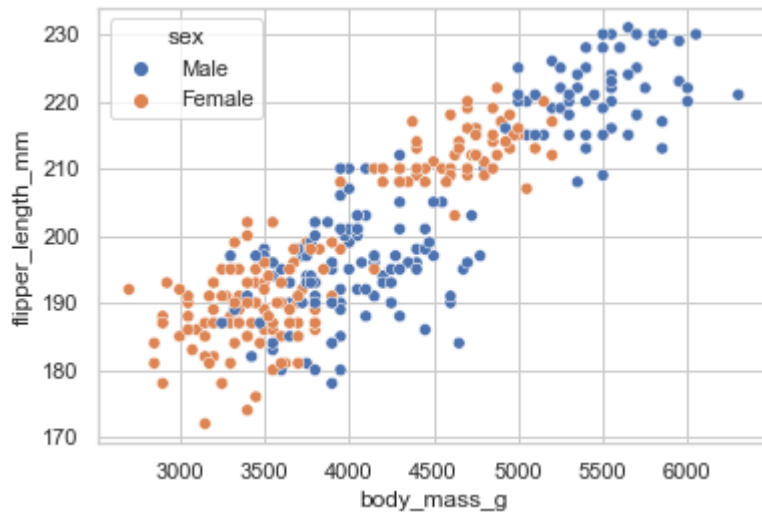
```
# 09. Scatterplot
# lineplot과 비슷하게 x, y에 대한 전체적인 분포를 확인하는 plot.
# lineplot은 경향성에 초점을 둔다면, scatterplot은 데이터 그 자체가 퍼져있는 모양에 중점

sns.scatterplot(data=pen, x='body_mass_g', y='flipper_length_mm', hue='sex')

# 여자 펭귄이 남자 펭귄에 비해 아래 쪽에 많은 점이 찍힌 것을 볼 수 있다.
# => 몸무게랑 발 사이즈가 더 작다.
```

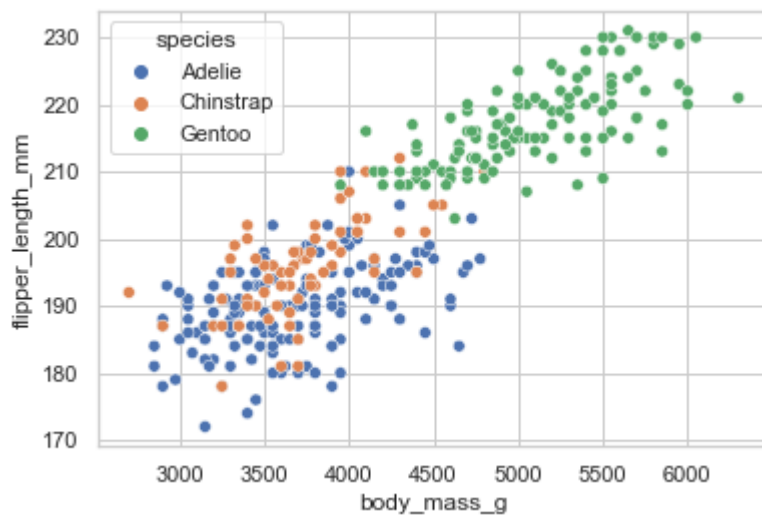
Out[18]:

```
<AxesSubplot:xlabel='body_mass_g', ylabel='flipper_length_mm'>
```



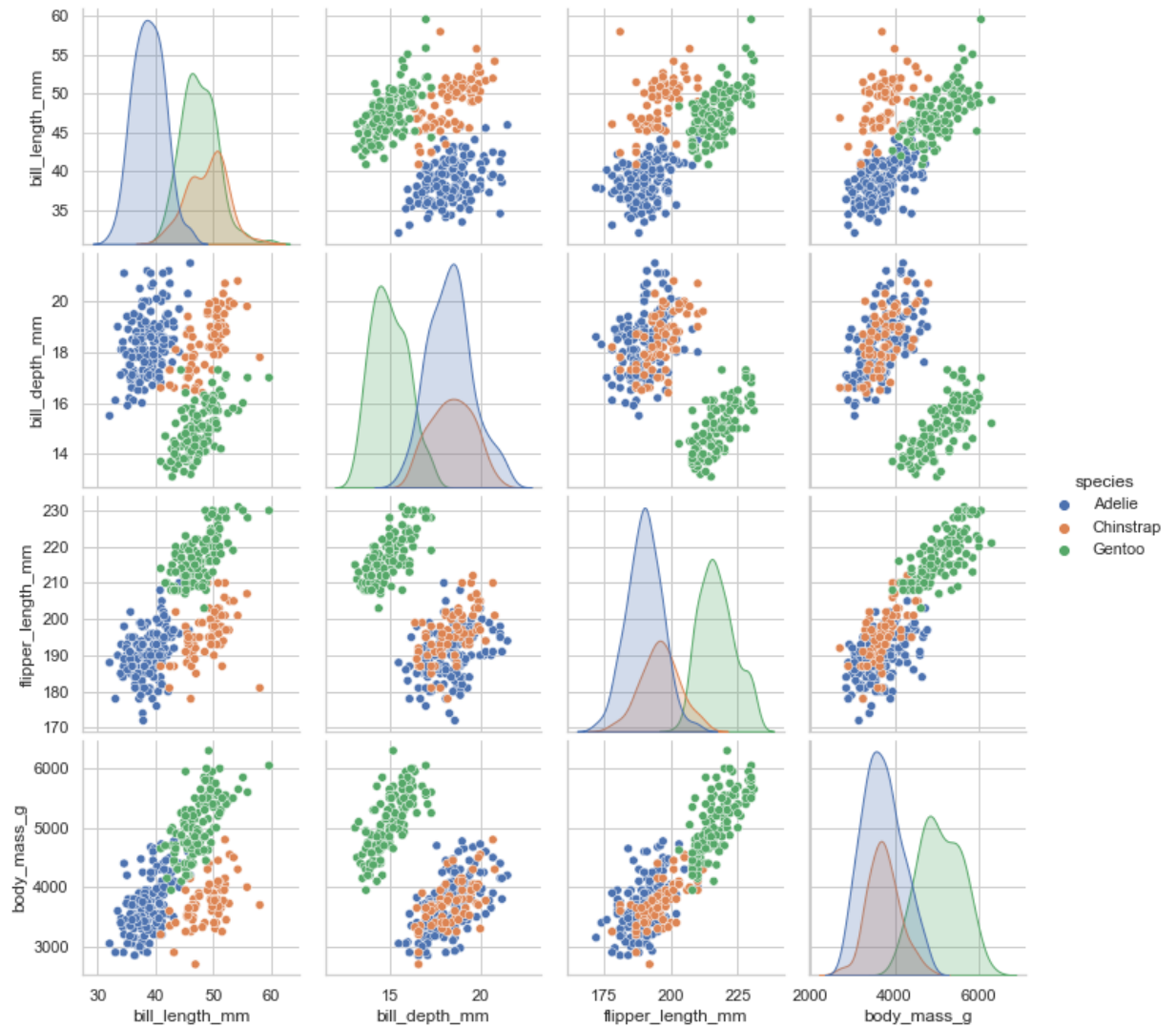
In [19]: `sns.scatterplot(data=pen, x='body_mass_g', y='flipper_length_mm', hue='species')`

Out[19]: `<AxesSubplot:xlabel='body_mass_g', ylabel='flipper_length_mm'>`



In [20]: `# 10. Pairplot`
`# 주어진 데이터의 각 feature들 사이의 관계를 표시하는 plot. (feature의 개수가 너무 많으면 관계`
`# scatterplot, FacetGrid, kdeplot을 이용하여 feature간의 관계를 잘 보여줌`
`sns.pairplot(data=pen, hue='species') # x, y를 따로 찍을 필요는 없음`

Out[20]: `<seaborn.axisgrid.PairGrid at 0x7fbd895658b0>`



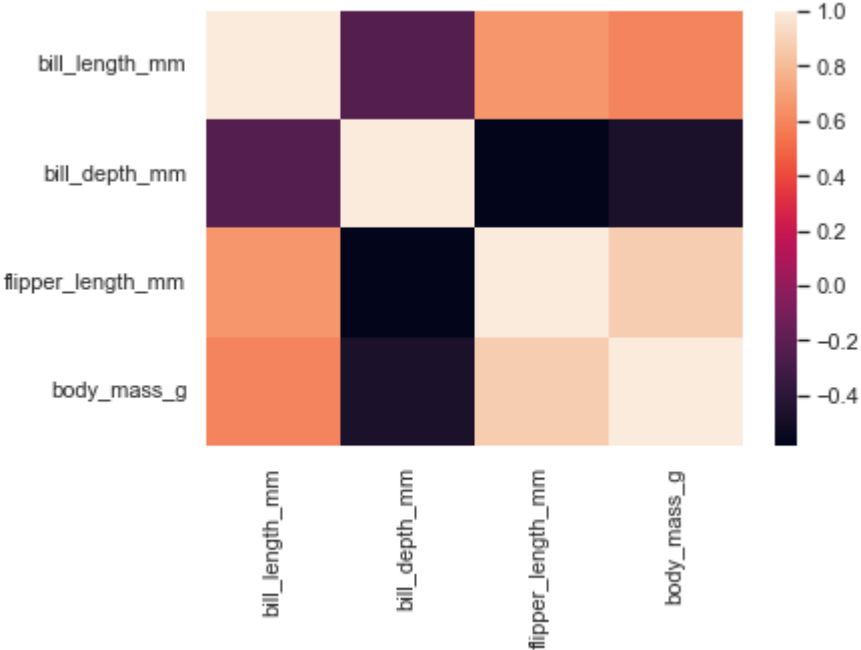
In [21]:

```
# 11. Heatmap
# 정사각형 그림에 데이터에 대한 정도 차이를 색 차이로 보여주는 plot.
# 열화상 카메라로 사물을 찍은 것처럼 정보의 차이를 보여줌. 그렇기 때문에 feature별로 수치 차이가
# pairplot과 비슷하게 feature간 관계를 시각화할 때 많이 사용됩니다. 만약 feature 수가 많은 경우

# 각 feature간 상관관계를 파악하기 위해 Correlation matrix를 만들
corr = pen.corr() # 자동으로 수치형 자료의 상관관계를 보여줌. 상관관계는 1 ~ -1의 값을 가짐

# penguin 데이터에 heatmap을 출력합니다
sns.heatmap(corr)
```

Out[21]: <AxesSubplot:>



```
In [ ]:
```