Learning python with Baba Ammar

Python Basics

My first Program

In [1]:

```
print(3+5)
print(14+10)
print("Hello to the World")
print("we are learning python with aamar")
```

```
8
24
Hello to the World
we are learning python with aamar
```

Operators in Python

In [2]:

```
print(3+2)
print(6*6)
print(6//6)
print(6/3)
print(2**3)
print(2**3/2*3/3+6-4)
```

```
5
36
1
2.0
8
6.0
```

PEMDAS Paraenthesis, Exponent, Multipy, Divide, Addition and Subtraction

String in Python

In [3]:

```
print("Hello to the World")
print("we are learning python with aamar")
print("what's up")
```

```
Hello to the World
we are learning python with aamar
what's up
```

Usage of Comments (Ctlr+/)

In [4]:

```
print("Hello to the World") #use of inverted commas
print("we are learning python with aamar") #Comments are used to explain the code
print('''learning in python''')
```

```
Hello to the World
we are learning python with aamar
learning in python
```

Loading [MathJax]/extensions/Safe.js

Variables are objects containing specific values Rules to Assign Variables 1- the variable shouls contain letters, numbers and underscores 2-Donot start with number 3-Spaces are not allowed 4-Donot use Keywords used in functions (break, mean, media, test etc) 5-short and descriptive 6-case sensitive (lower case letter should be used)

In [5]:

```
x=5
print(x)
y="we are learning python with ammaar"
print(y)
# Types/Class of Variables
print(type(x),type(y))
fruit=8
fruit="mangoes"
# del fruit  del is used to delete the variable
# Last updates varible value will show from integer to string
print(type(fruit))
```

```
5
we are learning python with ammaar
<class 'int'> <class 'str'>
<class 'str'>
```

Input Variables

In [6]:

```
fruit=8
print(fruit)
# input function simple
fruit=input("Enter the Name of your Favourite Fruite.....?")
print("Greeting! Your Favourite fruite is", fruit)
# Input Function 2nd Stage
name=input("name of fruit")
greetings="Hello"
print(greetings,name)
# another way of Input Function 2nd Stage
name=input("What is Your Name?  ")
print("Hello",name)
# 3rd Stage if input function
name1=input("Enter Your First Name? ")
name2=input("Enter Your Last Name? ")
age=input("how old are You? ")
print("Hello!", name1, name2, ", You are still young")
```

```
8
Enter the Name of your Favourite Fruite.....?Mango
Greeting! Your Favourite fruite is Mango
name of fruitApple
Hello Apple
What is Your Name?  Kashif
Hello Kashif
Enter Your First Name? Kashif
Enter Your Last Name? Rehman
how old are You? 30
Hello! Kashif Rehman , You are still young
```

logical operators are either true or false or yes or no, 0 or 1 equal to == not equal to != less than < greater than > less than and equal to <= greater than and equal to >= Q:is 6 is equal to 6??

Loading [MathJax]/extensions/Safe.js

In [8]:

```
print(6==6)
print(6!=6)
print(3<10)
print(15>7)
print(4<=10)
print(15>=10)
# # Application of logical operator
haris_age=7
eligible_school_age=5
print(haris_age==eligible_school_age)
#Input variable and conditional logics
haris_age=input("enter the age? ")
eligible_school_age=5
haris_age=int(haris_age) #string to integer conversion
print(haris_age==eligible_school_age)
```

```
True
False
True
True
True
True
False
enter the age? 5
True
```

Conversion Types

In [9]:

```
x=10
y=15.5
# print(type(y))
x*y
print(x*y,type(x*y))
# explicit typr conversion
age=input("enter your age? ")
#Float can not be convert into integer
# string can be convert into int
print(age,type(int(age)))
```

```
155.0 <class 'float'>
enter your age? 10
10 <class 'int'>
```

if, else and elif

In [10]:

```
# Q: if required school age is 5 then haris can join the school
haris_age=input("enter the age of haris? ")
haris_age=int(haris_age)
# haris_age=10
required_school_age=5
if haris_age==required_school_age:
    print("haris can join the school")
elif  haris_age > required_school_age:
    print("haris should join Higher School")
elif  haris_age <=2:
    print("haris is still Baby")
else:
s can not join School ")
```

Loading [MathJax]/extensions/Safe.js

```
enter the age of haris? 5
haris can join the school
```

functions

In [13]:

```
#if you want to print same value,operation more than once then it is suitable to create
# Define a function
# 1st way
def print_codanics():
    print("we learning python")
    print("we learning python")
    print("we learning python")
print_codanics()
# 2nd way
def print_codanics():
    text="we learning python"
    print(text)
    print(text)
    print(text)
print_codanics()
# 3rd way
def print_codanics(text):
    # text="we learning python"
    print(text)
    print(text)
    print(text)
print_codanics("we learning python")
#Defining a fnction with if and else statement
def school_calculator(age):
    if age==5:
        print("haris can join school")
    elif age<=2:
        print("haris is still baby")
    elif age>=10:
        print("haris should join highr school")
    else:
        print("haris can not join school")
school_calculator(8)
def future_age(age):
    new_age=age+20
    return new_age
# future_age(10)
# print(new_age)
future_predicted_age=future_age(10)
print(future_predicted_age)
```

```
we learning python
we learning python
we learning python
we learning python
we learning python
we learning python
we learning python
we learning python
we learning python
haris can not join school
30
```

Loading [MathJax]/extensions/Safe.js s

In [14]:

```
# while and for Loops
# While Loop
x=0
while (x<10):
    x=x+1
    print(x)
for x in range(8,15):
    print(x)
# Array
days=["Mon","Tue","Wed","Thur","Fri","Sat"]
for i in days:
    # if(i=="Fri"):break #Loop stops
    if(i=="Fri"):continue #skips i
    print(i)
```

```
1
2
3
4
5
6
7
8
9
10
8
9
10
11
12
13
14
Mon
Tue
Wed
Thur
Sat
```

Import Libraries

In [15]:

```

# if you want to print the value of pi

import math
print("The value of pi is ",math.pi)
import statistics

x=[50,100,150,200,250]
print(statistics.mean(x))
#numpy, pandas
```

```
The value of pi is  3.141592653589793
150
```

Trouble Shooting

print(we are learning python) #syntax error

Loading [MathJax]/extensions/Safe.js

(25/0) #runtime error

In [ ]:

```
# Learning python with Baba Ammar
## Python Basics
### My first Program

print(3+5)
print(14+10)
print("Hello to the World")
print("we are learning python with aamar")

**Operators in Python**

print(3+2)
print(6*6)
print(6//6)
print(6/3)
print(2**3)
print(2**3/2*3/3+6-4)

*PEMDAS*
*Paraenthesis, Exponent, Multipy, Divide, Addition and Subtraction*

**String in Python**

print("Hello to the World")
print("we are learning python with aamar")
print("what's up")

**Usage of Comments (Ctlr+/)**

print("Hello to the World") #use of inverted commas
print("we are learning python with aamar") #Comments are used to explain the code
print('''learning in python''')

**Variables are objects containing specific values**
Rules to Assign Variables
1- the variable shouls contain letters, numbers and underscores
2-Donot start with number
3-Spaces are not allowed
4-Donot use Keywords used in functions (break, mean, media, test etc)
5-short and descriptive
6-case sensitive (lower case letter should be used)

x=5
print(x)
y="we are learning python with ammaar"
print(y)
# Types/Class of Variables
print(type(x),type(y))
fruit=8
fruit="mangoes"
# del fruit  del is used to delete the variable
# Last updates varible value will show from integer to string
print(type(fruit))

**Input Variables**
```

Loading [MathJax]/extensions/Safe.js

```
fruit=8
print(fruit)
# input function simple
fruit=input("Enter the Name of your Favourite Fruite.....?")
print("Greeting! Your Favourite fruite is", fruit)
# Input Function 2nd Stage
name=input("name of fruit")
greetings="Hello"
print(greetings,name)
# another way of Input Function 2nd Stage
name=input("What is Your Name?  ")
print("Hello",name)
# 3rd Stage if input function
name1=input("Enter Your First Name? ")
name2=input("Enter Your Last Name? ")
age=input("how old are You? ")
print("Hello!", name1, name2, ", You are still young")
```

```
**logical operators are either true or false or yes or no, 0 or 1**
equal to                          ==
not equal to                      !=
less than                         <
greater than                      >
less than and equal to            <=
greater than and equal to         >=
Q:is 6 is equal to 6??
```

```
print(6==6)
print(6!=6)
print(3<10)
print(15>7)
print(4<=10)
print(15>=10)
# # Application of logical operator
haris_age=7
eligible_school_age=5
print(haris_age==eligible_school_age)
#Input variable and conditional logics
haris_age=input("enter the age? ")
eligible_school_age=5
haris_age=int(haris_age) #string to integer conversion
print(haris_age==eligible_school_age)
```

```
**Conversion Types**
```

```
x=10
y=15.5
# print(type(y))
x*y
print(x*y,type(x*y))
# explicit typr conversion
age=input("enter your age? ")
#Float can not be convert into integer
# string can be convert into int
print(age,type(int(age)))
```

```
**if, else and elif**
```

```
Loading [MathJax]/extensions/Safe.js d school age is 5 then haris can join the school
haris_age=input("enter the age of haris? ")
```

```
haris_age=int(haris_age)
# haris_age=10
required_school_age=5
if haris_age==required_school_age:
    print("haris can join the school")
elif  haris_age > required_school_age:
    print("haris should join Higher School")
elif  haris_age <=2:
    print("haris is still Baby")
else:
    print("haris can not join School ")

**functions**

#if you want to print same value,operation more than once then it is suitable to create
# Define a function
# 1st way
def print_codanics():
    print("we learning python")
    print("we learning python")
    print("we learning python")
print_codanics()
# 2nd way
def print_codanics():
    text="we learning python"
    print(text)
    print(text)
    print(text)
print_codanics()
# 3rd way
def print_codanics(text):
    # text="we learning python"
    print(text)
    print(text)
    print(text)
print_codanics("we learning python")
#Defining a fnction with if and else statement
def school_calculator(age):
    if age==5:
        print("haris can join school")
    elif age<=2:
        print("haris is still baby")
    elif age>=10:
        print("haris should join highr school")
    else:
        print("haris can not join school")
school_calculator(8)
def future_age(age):
    new_age=age+20
    return new_age
# future_age(10)
# print(new_age)
future_predicted_age=future_age(10)
print(future_predicted_age)

**while and for loops**

# while and for loops
```

Loading [MathJax]/extensions/Safe.js

```
x=0
```

```
while (x<10):
    x=x+1
    print(x)
for x in range(8,15):
    print(x)
# Array
days=["Mon","Tue","Wed","Thur","Fri","Sat"]
for i in days:
    # if(i=="Fri"):break #Loop stops
    if(i=="Fri"):continue #skips i
    print(i)

**Import Libraries**

# if you want to print the value of pi

import math
print("The value of pi is ",math.pi)
import statistics

x=[50,100,150,200,250]
print(statistics.mean(x))
#numpy, pandas

**Trouble Shooting**
#print(we are learning python) #syntax error

#(25/0)  #runtime error
```

-Indexing

In [1]:

```
#Make a string
a="roti salun"
a
```

Out[1]:

```
'roti salun'
```

In [2]:

```
a[0]
```

Out[2]:

```
'r'
```

In [3]:

```
a[1]
```

Out[3]:

```
'o'
```

In [4]:

```
a[3]
```

Out[4]:

```
'i'
```

In [5]:

```
a[5]
```

Loading [MathJax]/extensions/Safe.js

```
Out[5]: 's'
```

```
In [6]: #length of indeces
        len(a)
```

```
Out[6]: 10
```

```
In [7]: a[9]
```

```
Out[7]: 'n'
```

```
In [8]: a[0:4]
```

```
Out[8]: 'roti'
```

```
In [9]: #last index is inclusive
        a[0:9]
```

```
Out[9]: 'roti salu'
```

```
In [10]: a[-1]
```

```
Out[10]: 'n'
```

```
In [11]: a[-10:-1]
```

```
Out[11]: 'roti salu'
```

-String Methods

```
In [12]: food="biryani"
         food
```

```
Out[12]: 'biryani'
```

```
In [13]: len(food)
```

```
Out[13]: 7
```

```
In [14]: food.capitalize()
```

```
Out[14]: 'Biryani'
```

Loading [MathJax]/extensions/Safe.js

In [15]:
```
#lowecase
food.casefold()
```

Out[15]:
```
'biryani'
```

In [16]:
```
food.upper()
```

Out[16]:
```
'BIRYANI'
```

In [17]:
```
food.lower()
```

Out[17]:
```
'biryani'
```

In [18]:
```
food.replace("b","s")
```

Out[18]:
```
'siryani'
```

In [19]:
```
# Counting a specific alphbet in  astring
name="learnig python"
name
```

Out[19]:
```
'learnig python'
```

In [20]:
```
name.count("n")
```

Out[20]:
```
2
```

-Finding index number in string

In [21]:
```
name="My name is Kashif Rehman and learning Python with Baba aammar"
name
```

Out[21]:
```
'My name is Kashif Rehman and learning Python with Baba aammar'
```

In [22]:
```
name.find("B")
```

Out[22]:
```
50
```

How split a string

In [23]:
```
food="I love samosa, pakora, raita, biryani and karahi"
food
```

Out[23]:
```
'I love samosa, pakora, raita, biryani and karahi'
```

```
In [24]: food.split(",")

Out[24]: ['I love samosa', ' pakora', ' raita', ' biryani and karahi']
```

Basic data Structure in Python

1-Tuple

2-List

3-Dictionaries

4-Set

1-Tuple

- ordered collection of elements
- enclosed in paraenthesis ()
- Different elements can be stored
- once elements stored then can not be changed (immutable)

```
In [25]: tup1=(1, "python", True, 2.5)
         tup1

Out[25]: (1, 'python', True, 2.5)
```

```
In [26]: type(tup1)

Out[26]: tuple
```

-indexing in Tuple

```
In [27]: tup1[0]

Out[27]: 1
```

```
In [28]: tup1[2]

Out[28]: True
```

```
In [29]: #Last element is exclusive
         tup1[0:3]
```

Loading [MathJax]/extensions/Safe.js

```
ue)
```

In [30]:
```
# count of element in tuple
len(tup1)
```
Out[30]:
```
4
```

In [31]:
```
tup2=(10, "Life", "programing", 5.5)
tup2
```
Out[31]:
```
(10, 'Life', 'programing', 5.5)
```

In [32]:
```
tup1+tup2
```
Out[32]:
```
(1, 'python', True, 2.5, 10, 'Life', 'programing', 5.5)
```

In [33]:
```
tup1*2+tup2
```
Out[33]:
```
(1, 'python', True, 2.5, 1, 'python', True, 2.5, 10, 'Life', 'programing', 5.5)
```

In [34]:
```
tup3=(20, 70, 50, 100, 250)
tup3
```
Out[34]:
```
(20, 70, 50, 100, 250)
```

In [35]:
```
# Minimum Value
min(tup3)
```
Out[35]:
```
20
```

In [36]:
```
# Maximum Value
max(tup3)
```
Out[36]:
```
250
```

In [37]:
```
tup3*2
```
Out[37]:
```
(20, 70, 50, 100, 250, 20, 70, 50, 100, 250)
```

2-List

- Ordered collection of elements
- enclosed in squre brackets [ ]
- Values can be changed (Mutatable)

In [38]:
```
list1=[10, "kashif", False]
```

Out[38]:
```
[10, 'kashif', False]
```

In [39]:
```
type(list1)
```

Out[39]:
```
list
```

In [40]:
```
len(list1)
```

Out[40]:
```
3
```

In [41]:
```
list1[2]
```

Out[41]:
```
False
```

In [42]:
```
list2=[8, 10, "Engineering", "Teaching", "Fasle"]
list2
```

Out[42]:
```
[8, 10, 'Engineering', 'Teaching', 'Fasle']
```

In [43]:
```
list1+list2
```

Out[43]:
```
[10, 'kashif', False, 8, 10, 'Engineering', 'Teaching', 'Fasle']
```

In [44]:
```
list2*2
```

Out[44]:
```
[8,
 10,
 'Engineering',
 'Teaching',
 'Fasle',
 8,
 10,
 'Engineering',
 'Teaching',
 'Fasle']
```

In [45]:
```
list1.reverse()
list1
```

Out[45]:
```
[False, 'kashif', 10]
```

In [46]:
```
list1.append("study")
list1
```

Out[46]:
```
[False, 'kashif', 10, 'study']
```

In [47]:

Loading [MathJax]/extensions/Safe.js

```
"orange")
list1
```

Out[47]:
```
['orange', False, 'kashif', 10, 'study']
```

In [50]:
```
list1.remove("orange")
list1
```

Out[50]:
```
[False, 'kashif', 10, 'study']
```

In [53]:
```
list3=[10, 15, 2, 45, 225, 78, 500, 25]
list3
```

Out[53]:
```
[10, 15, 2, 45, 225, 78, 500, 25]
```

In [54]:
```
len(list3)
```

Out[54]:
```
8
```

In [56]:
```
list3.sort()
list3
```

Out[56]:
```
[2, 10, 15, 25, 45, 78, 225, 500]
```

In [57]:
```
lists=list1+list2
lists
```

Out[57]:
```
[False, 'kashif', 10, 'study', 8, 10, 'Engineering', 'Teaching', 'Fasle']
```

In [58]:
```
list3*2
```

Out[58]:
```
[2, 10, 15, 25, 45, 78, 225, 500, 2, 10, 15, 25, 45, 78, 225, 500]
```

3- Dictionaries

- an unsorted collection of elements
- key and vlaues
- curly braces or brackets { }
- values can be changed

In [60]:
```
# Daily use items and their prices
items={"Tea":500, "Soap":50, "Surf":100, "Tomato":80, "Potato":60}
items
```

Out[60]:
```
{'Tea': 500, 'Soap': 50, 'Surf': 100, 'Tomato': 80, 'Potato': 60}
```

Loading [MathJax]/extensions/Safe.js

```
type(items)
```

Out[61]:

```
dict
```

In [62]:

```
len(items)
```

Out[62]:

```
5
```

In [64]:

```
# Extract Data
keys=items.keys()
keys
```

Out[64]:

```
dict_keys(['Tea', 'Soap', 'Surf', 'Tomato', 'Potato'])
```

In [65]:

```
values=items.values()
values
```

Out[65]:

```
dict_values([500, 50, 100, 80, 60])
```

In [66]:

```
#Adding an element
items["milk"]=110
items
```

Out[66]:

```
{'Tea': 500, 'Soap': 50, 'Surf': 100, 'Tomato': 80, 'Potato': 60, 'milk': 110}
```

In [67]:

```
# Update the values
items["Tea"]=550
items
```

Out[67]:

```
{'Tea': 550, 'Soap': 50, 'Surf': 100, 'Tomato': 80, 'Potato': 60, 'milk': 110}
```

In [68]:

```
items2={"Rice":120, "Channy":70, "Biscuit":150}
items2
```

Out[68]:

```
{'Rice': 120, 'Channy': 70, 'Biscuit': 150}
```

In [70]:

```
# Cancatenate
items.update(items2)
items
```

Out[70]:

```
{'Tea': 550,
 'Soap': 50,
 'Surf': 100,
 'Tomato': 80,
 'Potato': 60,
 'milk': 110,
 'Rice': 120,
 'Channy': 70,
 'Biscuit': 150}
```

Loading [MathJax]/extensions/Safe.js

4- Set

- unordered and undexed
- curly brackets are used
- no Duplicates are allowed

In [71]:

```
s1={10, 15.5, "Multan", "Medical", "Kashif", "Baba"}
s1
```

Out[71]:

```
{10, 15.5, 'Baba', 'Kashif', 'Medical', 'Multan'}
```

In [73]:

```
s1.add("Baba2")
s1
```

Out[73]:

```
{10, 15.5, 'Baba', 'Baba2', 'Kashif', 'Medical', 'Multan'}
```

In [74]:

```
s1.remove("Baba2")
s1
```

Out[74]:

```
{10, 15.5, 'Baba', 'Kashif', 'Medical', 'Multan'}
```

In [75]:

```
s2={12,45,78,12}
s2
```

Out[75]:

```
{12, 45, 78}
```

In [76]:

```
s1.update(s2)
s1
```

Out[76]:

```
{10, 12, 15.5, 45, 78, 'Baba', 'Kashif', 'Medical', 'Multan'}
```

In [78]:

```
# Difference
s1.difference(s2)
```

Out[78]:

```
{10, 15.5, 'Baba', 'Kashif', 'Medical', 'Multan'}
```

In [80]:

```
s1.discard(10)
s1
```

Out[80]:

```
{12, 15.5, 45, 78, 'Baba', 'Kashif', 'Medical', 'Multan'}
```

Import Libraries

seaborn automatically installs these libraries

- numpy *scipy*pandas *matplotlib

In [1]:

```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

phool=sns.load_dataset("iris")
phool
```

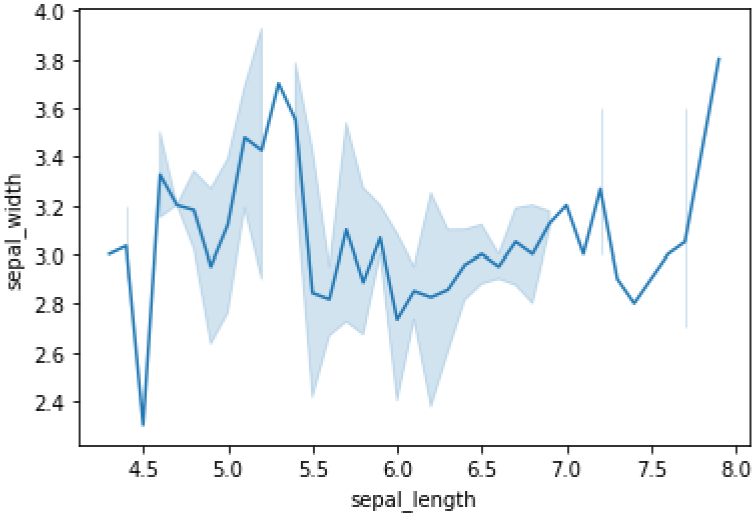
Out[1]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [2]:

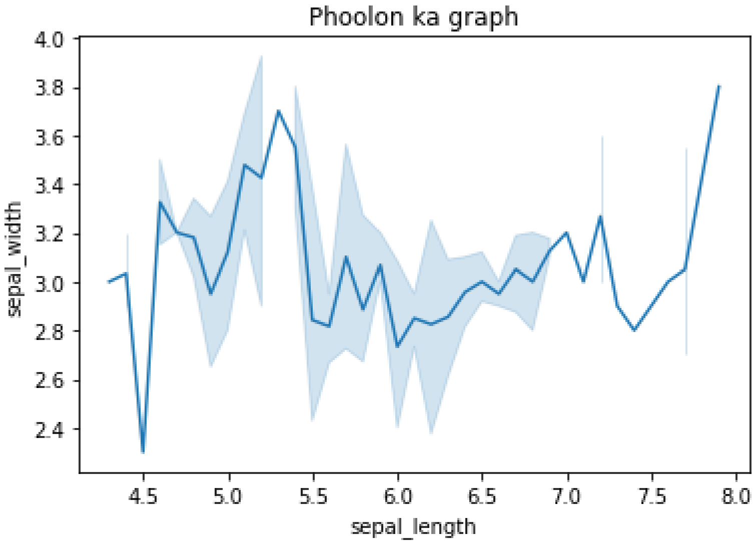
```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

phool=sns.load_dataset("iris")
phool
# Draw a Line Plot
sns.lineplot(x="sepal_length", y="sepal_width", data=phool)
plt.show()
```



Adding Titles in Graph

In [3]:

```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

phool=sns.load_dataset("iris")
phool
# Draw a Line Plot
sns.lineplot(x="sepal_length", y="sepal_width", data=phool)
plt.title("Phoolon ka graph")
plt.show()
```



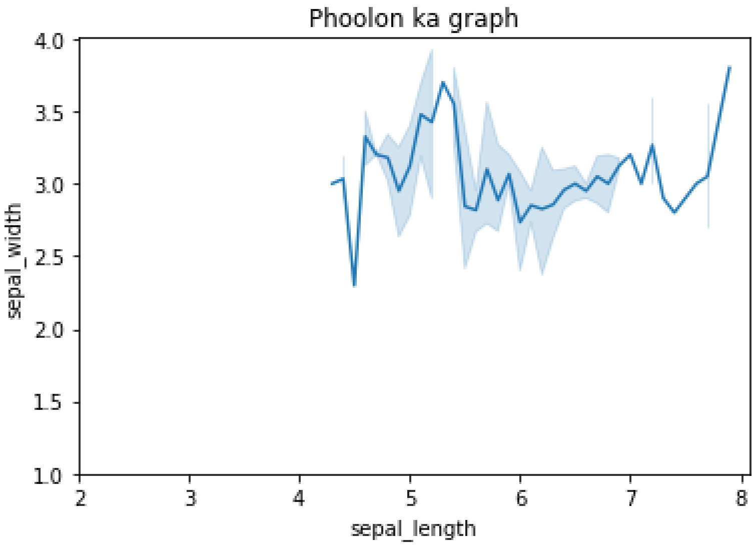
Adding limits in Graph

In [4]:

```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
```

Loading [MathJax]/extensions/Safe.js

```
phool=sns.load_dataset("iris")
phool
# Draw a Line Plot
sns.lineplot(x="sepal_length", y="sepal_width", data=phool)
plt.title("Phoolon ka graph")
plt.xlim(2)
plt.ylim(1)
plt.show()
```

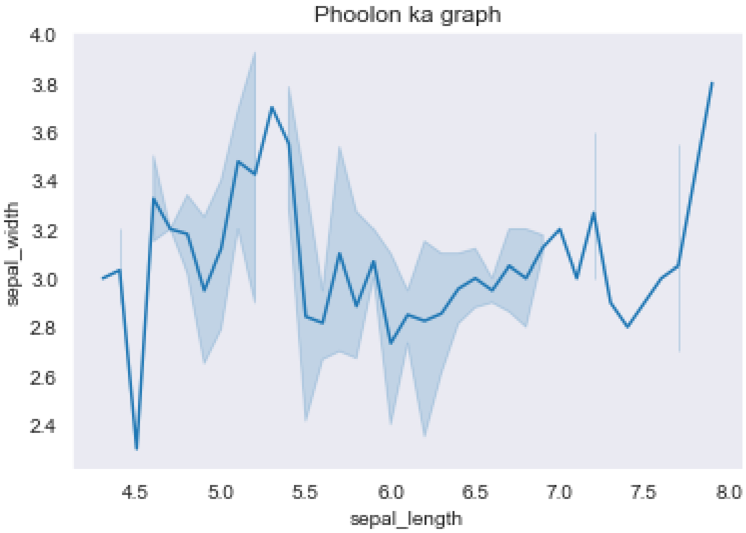


# Set Styles

- Darkgrid
- whitegrid
- Dark
- White
- ticks

In [9]:

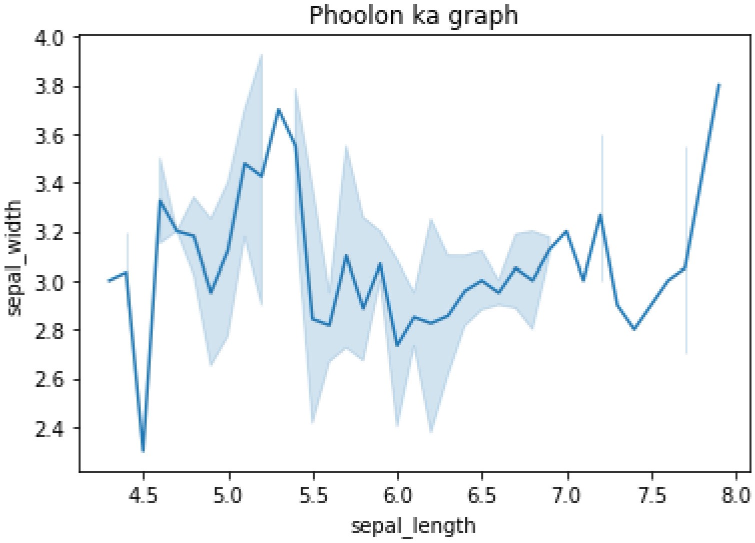
```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data
phool=sns.load_dataset("iris")
phool
# Draw a Line Plot
sns.lineplot(x="sepal_length", y="sepal_width", data=phool)
plt.title("Phoolon ka graph")
# Set Style
sns.set_style("dark")
sns.set_style(style=None, rc=None)
plt.show()
```



Size of Figure

In [6]:

```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt

# Load Data

phool=sns.load_dataset("iris")
phool
# Draw a Line Plot
sns.lineplot(x="sepal_length", y="sepal_width", data=phool)
plt.title("Phoolon ka graph")
plt.figure(figsize=(8,6))
plt.show()
```



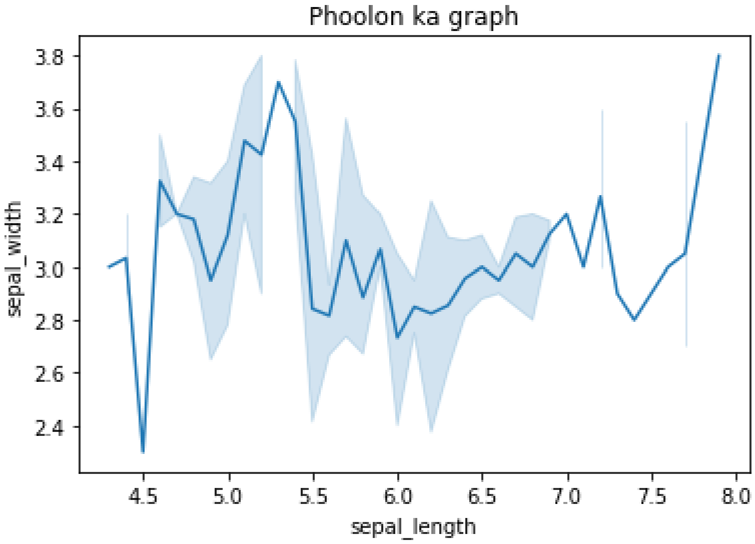
```
<Figure size 576x432 with 0 Axes>
```

In [7]:

```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
```

Loading [MathJax]/extensions/Safe.js

```
# Load Data

phool=sns.load_dataset("iris")
phool
# Draw a Line Plot
sns.lineplot(x="sepal_length", y="sepal_width", data=phool)
plt.title("Phoolon ka graph")
plt.figure(figsize=(10,8))
plt.show()
```

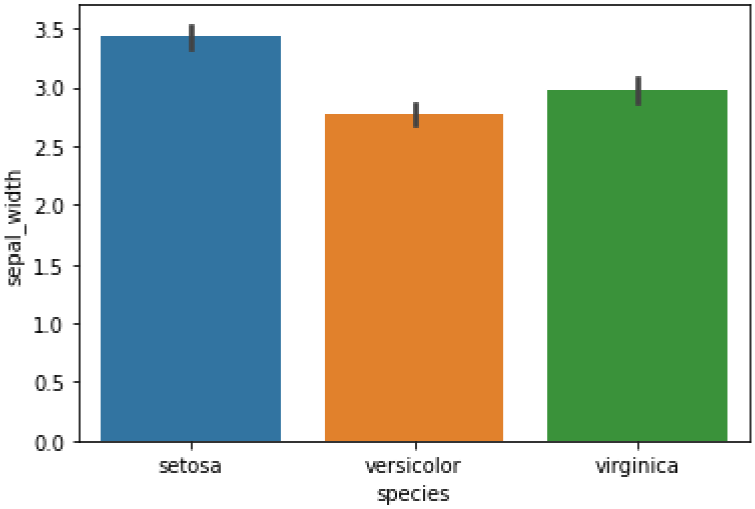


```
<Figure size 720x576 with 0 Axes>
```

Bar Plot

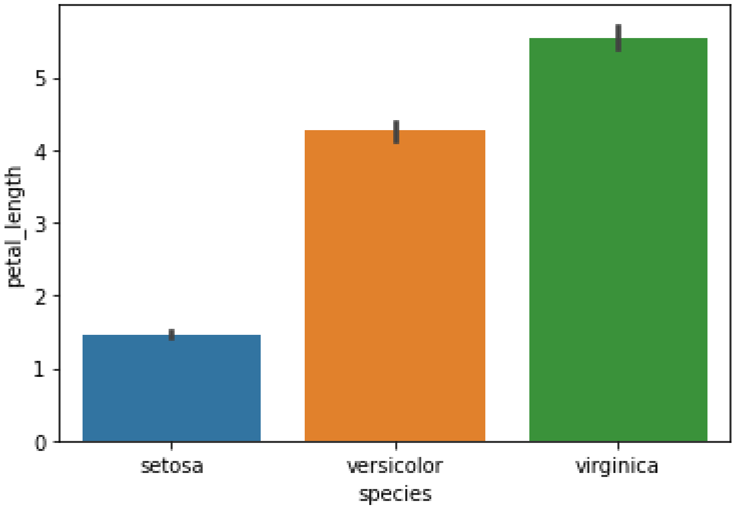
In [1]:

```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

phool=sns.load_dataset("iris")
phool
# Draw a Line Plot
sns.barplot(x="species", y="sepal_width", data=phool)
plt.show()
```



In [2]:

```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

phool=sns.load_dataset("iris")
phool
# Draw a Line Plot
sns.barplot(x="species", y="petal_length", data=phool)
plt.show()
```



In [3]:

```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
```

Out[3]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	dec
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	Na
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	

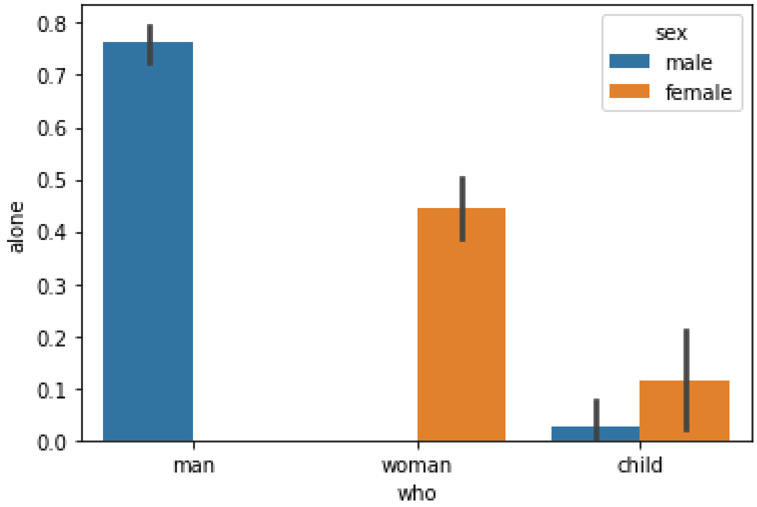
Loading [MathJax]/extensions/Safe.js

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	dec
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	Na
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	Na
...	...	...	...	...	...	...	...	...	...	...	...	
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	Na
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	Na
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	Na

891 rows × 15 columns

In [5]:

```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
sns.barplot(x="who", y="alone", hue="sex", data=kashti)
plt.show()
```

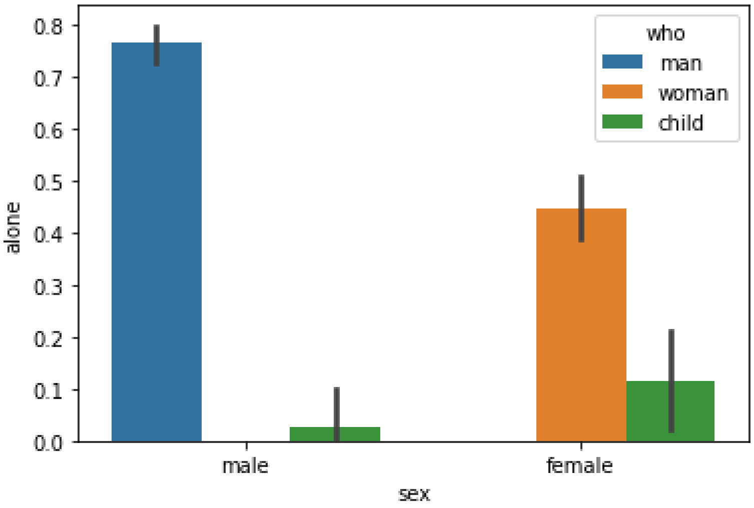


In [6]:

```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
```

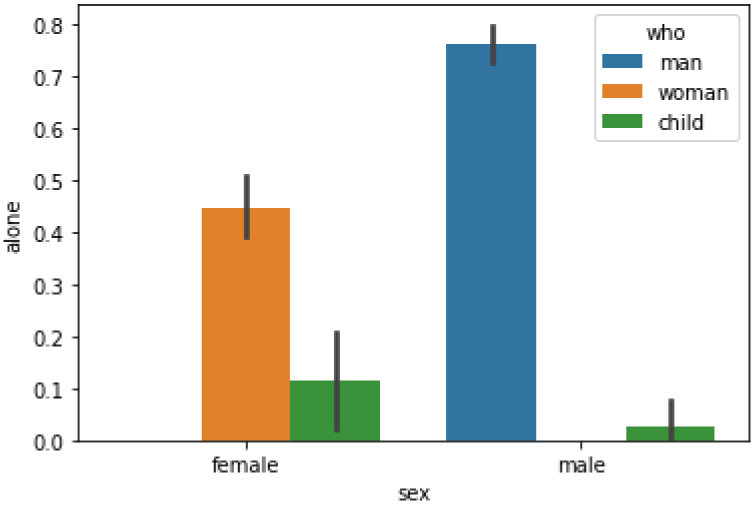
Loading [MathJax]/extensions/Safe.js

```
sns.barplot(x="sex", y="alone", hue="who", data=kashti)
plt.show()
```



In [8]:

```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
sns.barplot(x="sex", y="alone", hue="who", data=kashti, order=["female", "male"])
plt.show()
```

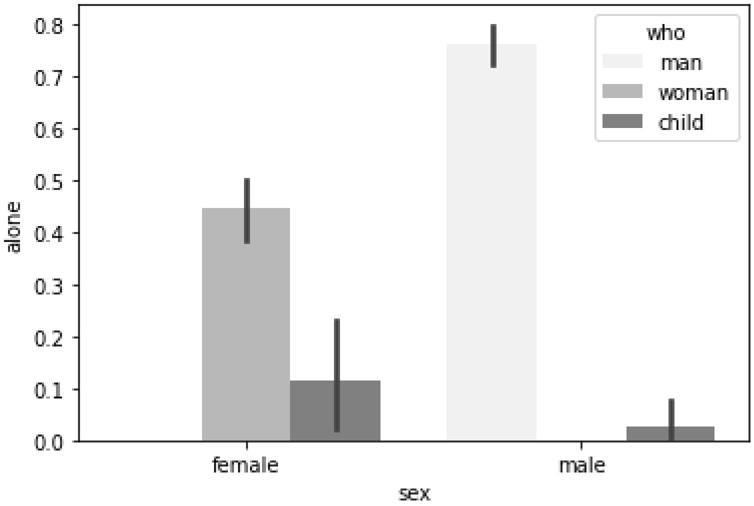


In [10]:

```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
```

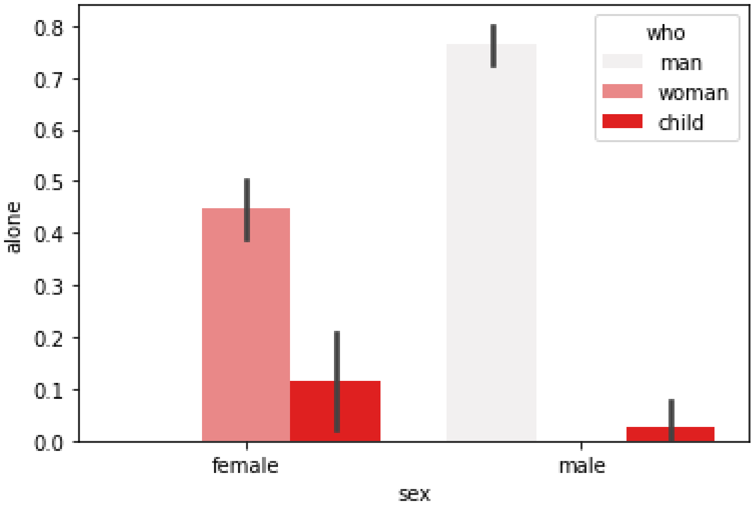
Loading [MathJax]/extensions/Safe.js

```
sex", y="alone", hue="who", data=kashti, order=["female", "male"], color
plt.show()
```



In [11]:

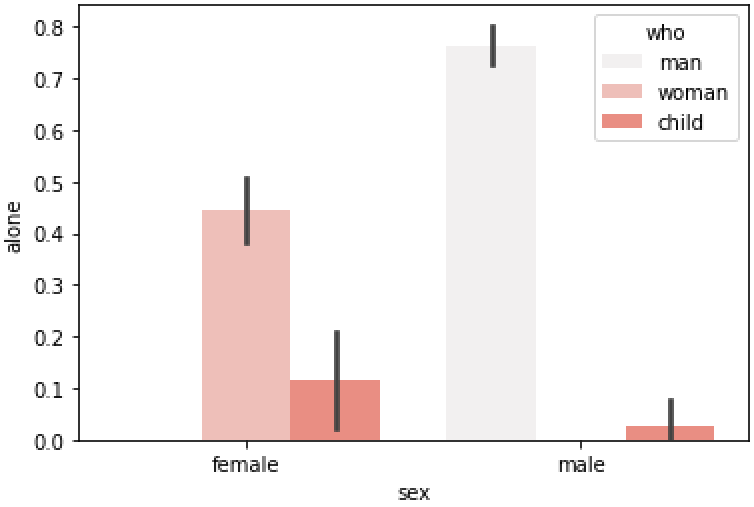
```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
sns.barplot(x="sex", y="alone", hue="who", data=kashti, order=["female", "male"], color
plt.show()
```



In [12]:

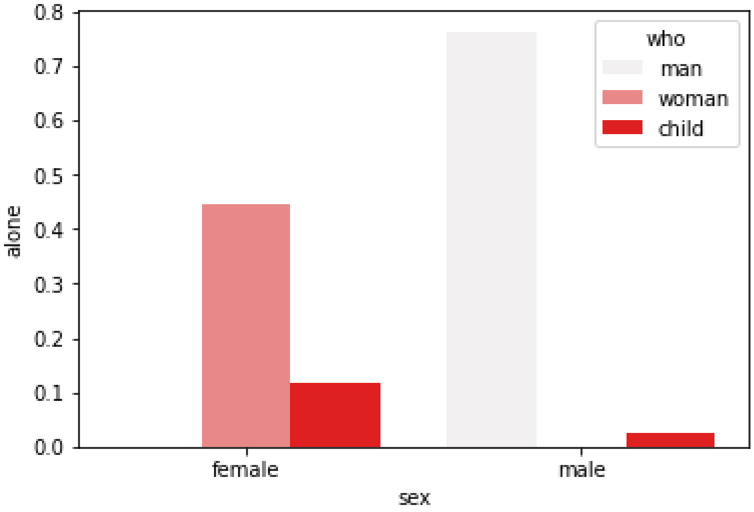
```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
sns.barplot(x="sex", y="alone", hue="who", data=kashti, order=["female", "male"], color
plt.show()
```

Loading [MathJax]/extensions/Safe.js



In [13]:

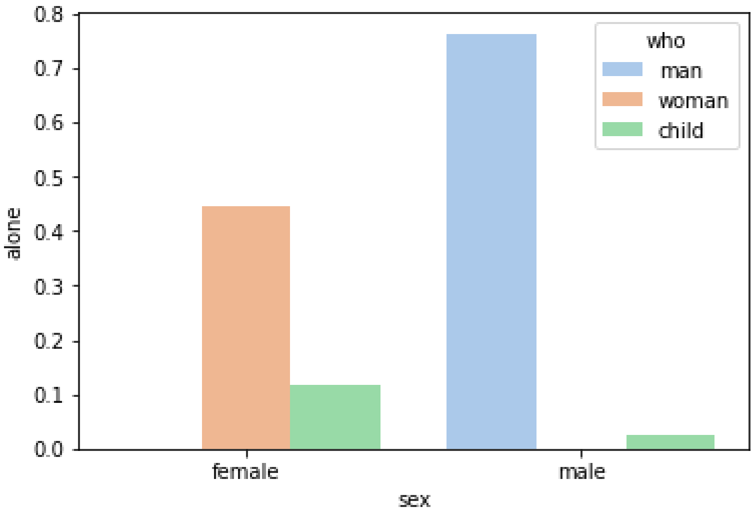
```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
sns.barplot(x="sex", y="alone", hue="who", data=kashti, order=["female", "male"], color
plt.show()
```



In [14]:

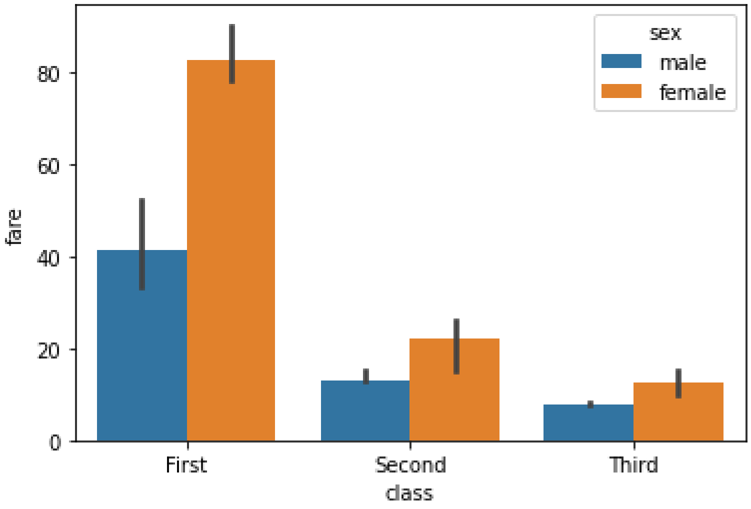
```
# Import Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
sns.barplot(x="sex", y="alone", hue="who", data=kashti, order=["female", "male"], color
plt.show()
```

Loading [MathJax]/extensions/Safe.js



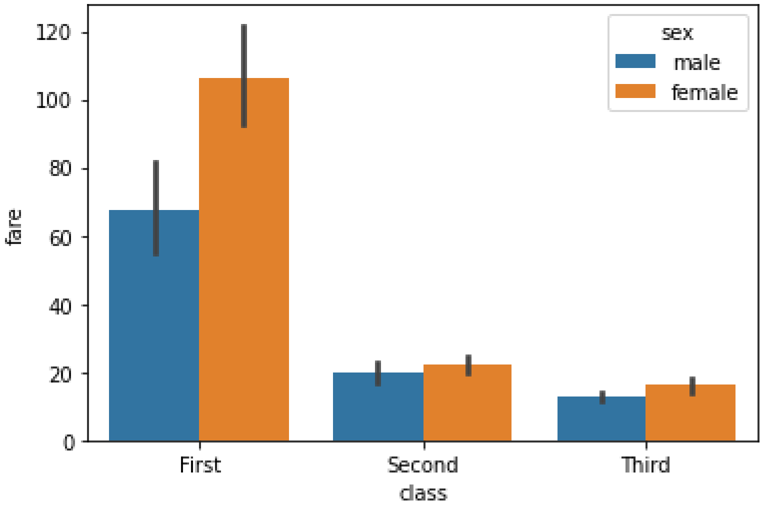
In [19]:

```
# Import Libraries
import seaborn as sns
import numpy
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
sns.barplot(x="class", y="fare", hue="sex", data=kashti, estimator=median)
plt.show()
```



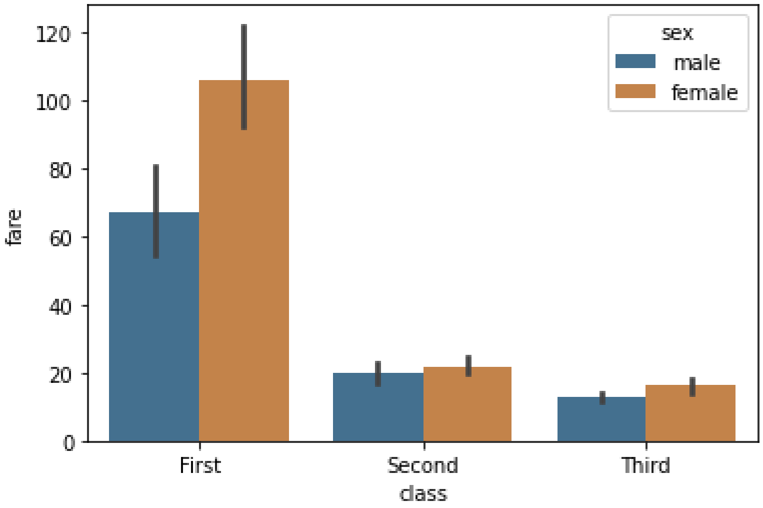
In [18]:

```
# Import Libraries
import seaborn as sns
import numpy
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
sns.barplot(x="class", y="fare", hue="sex", data=kashti, estimator=mean)
```



In [20]:

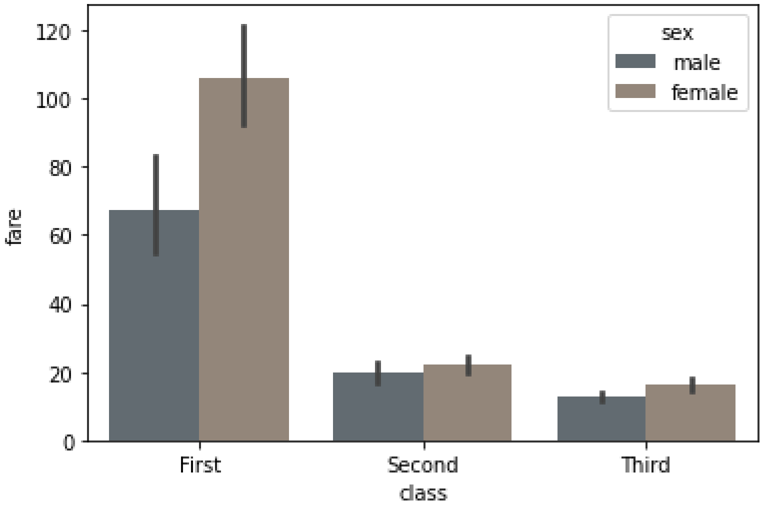
```
# Import Libraries
import seaborn as sns
import numpy
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
sns.barplot(x="class", y="fare", hue="sex", data=kashti, estimator=mean, saturation=0.5
plt.show()
```



In [21]:

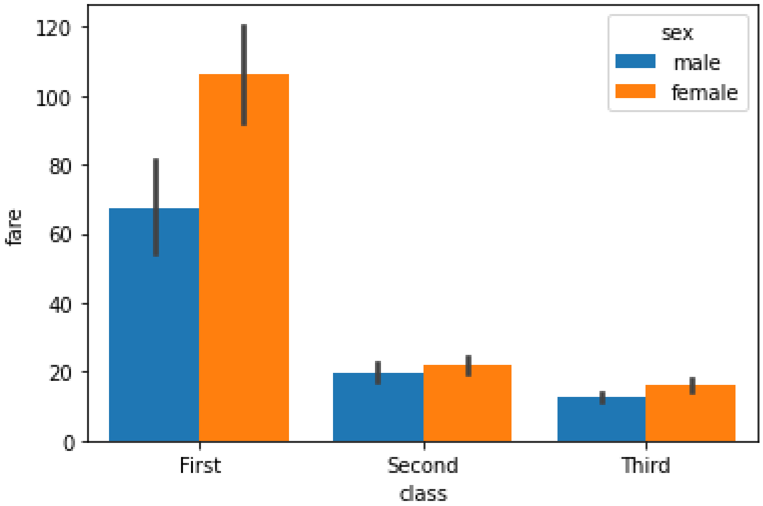
```
# Import Libraries
import seaborn as sns
import numpy
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
sns.barplot(x="class", y="fare", hue="sex", data=kashti, estimator=mean, saturation=0.1
```

Loading [MathJax]/extensions/Safe.js



In [22]:

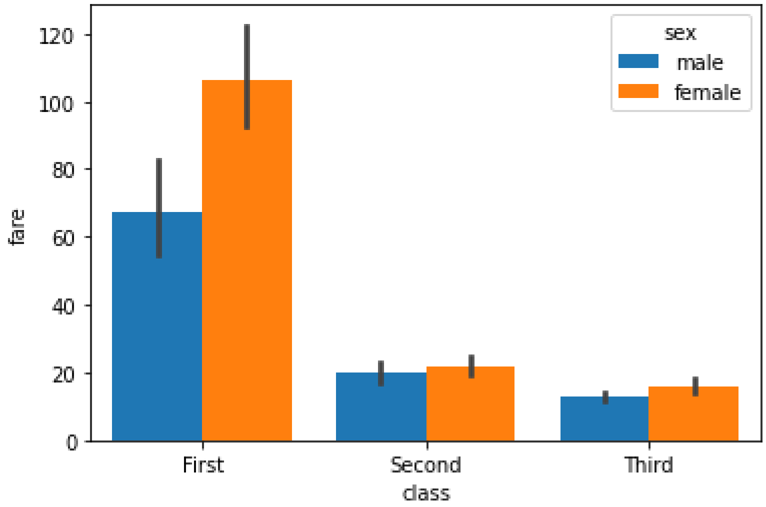
```
# Import Libraries
import seaborn as sns
import numpy
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
sns.barplot(x="class", y="fare", hue="sex", data=kashti, estimator=mean, saturation=1)
plt.show()
```



In [23]:

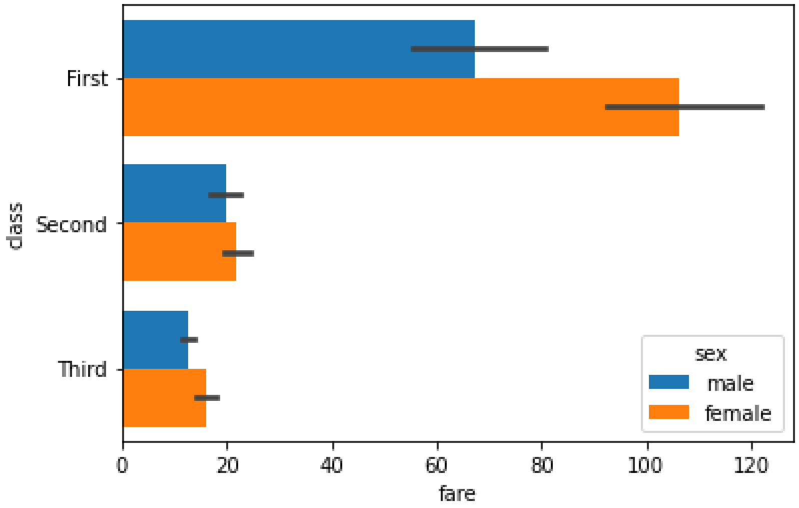
```
# Import Libraries
import seaborn as sns
import numpy
import matplotlib.pyplot as plt
# Load Data

kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
sns.barplot(x="class", y="fare", hue="sex", data=kashti, estimator=mean, saturation=2)
```

Loading [MathJax]/extensions/Safe.js



In [24]:

```
# Horizontal Plot
# Import Libraries
import seaborn as sns
import numpy
import matplotlib.pyplot as plt
# Load Data
kashti=sns.load_dataset("titanic")
kashti
# Draw a Line Plot
sns.barplot(x="fare", y="class", hue="sex", data=kashti, estimator=mean, saturation=2)
plt.show()
```



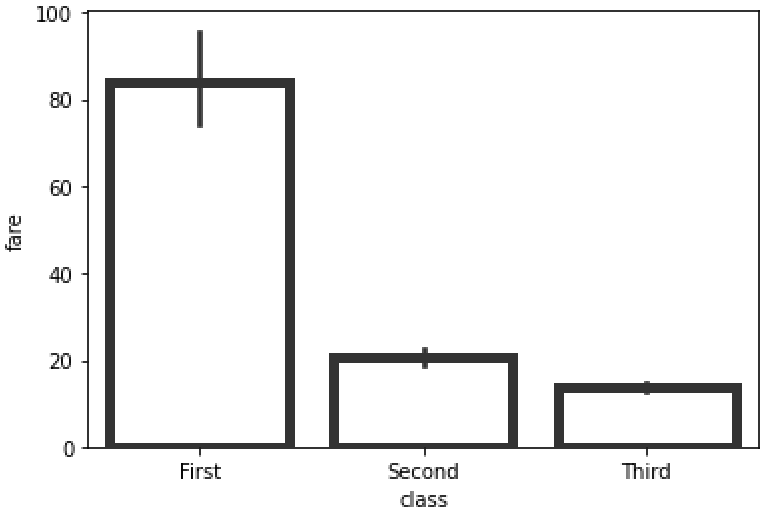
In [27]:

```
# Import the required Library
import seaborn as sns
import matplotlib.pyplot as plt
# read the titanic.csv file from seaborn library
kashti=sns.load_dataset("titanic")
sns.barplot(x="class", y="fare", data=kashti, linewidth=5, facecolor=(1,1,1,0), errcolo
```

Out[27]:

```
<AxesSubplot:xlabel='class', ylabel='fare'>
```

Loading [MathJax]/extensions/Safe.js

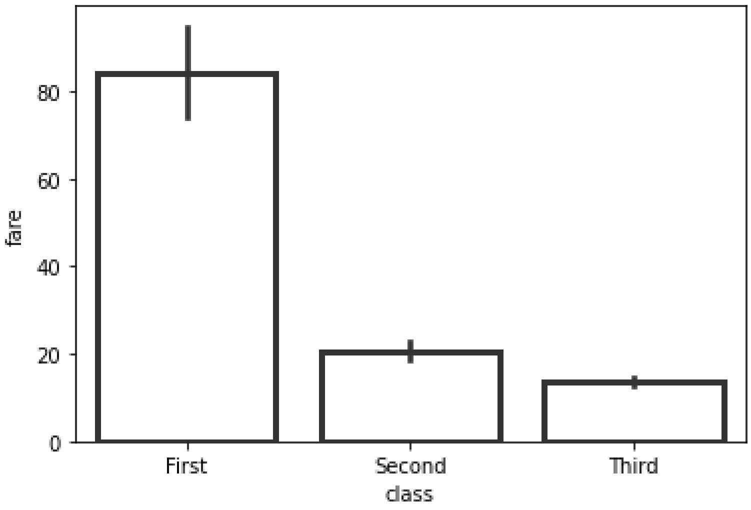


In [28]:

```
# Import the required Library
import seaborn as sns
import matplotlib.pyplot as plt
# read the titanic.csv file from seaborn library
kashti=sns.load_dataset("titanic")
sns.barplot(x="class", y="fare", data=kashti, linewidth=3, facecolor=(1,1,1,0), errcolo
```

Out[28]:

```
<AxesSubplot:xlabel='class', ylabel='fare'>
```



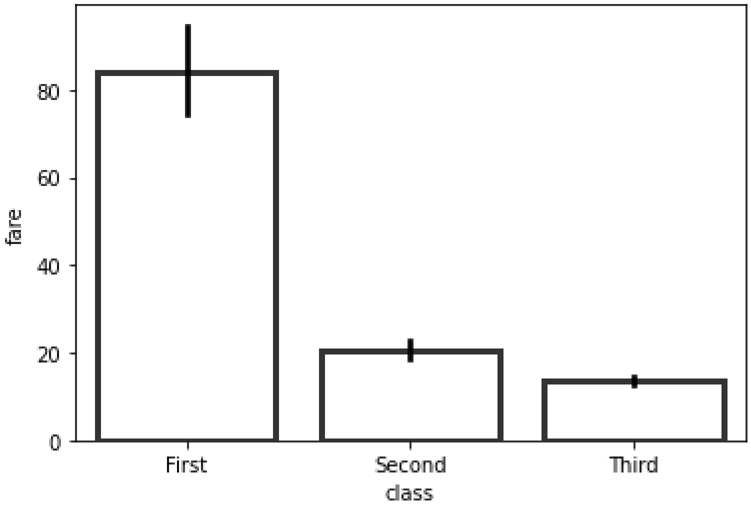
In [29]:

```
# Import the required Library
import seaborn as sns
import matplotlib.pyplot as plt
# read the titanic.csv file from seaborn library
kashti=sns.load_dataset("titanic")
sns.barplot(x="class", y="fare", data=kashti, linewidth=3, facecolor=(1,1,1,0), errcolo
```

Out[29]:

```
<AxesSubplot:xlabel='class', ylabel='fare'>
```

Loading [MathJax]/extensions/Safe.js

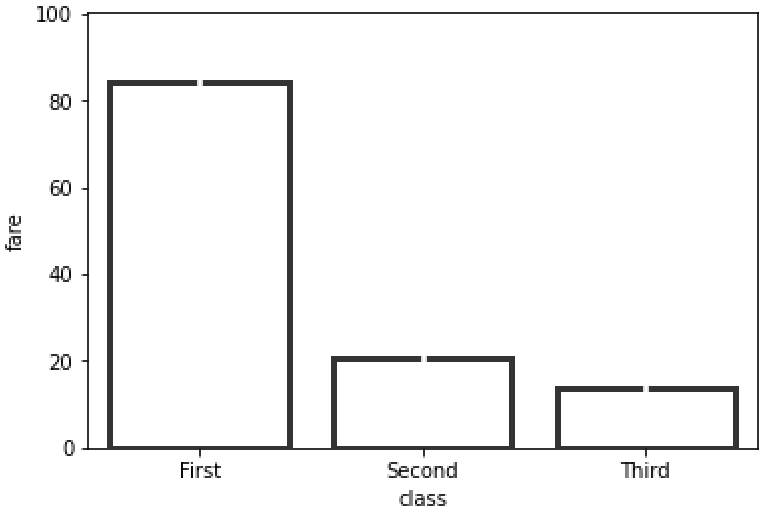


In [30]:

```
# Import the required Library
import seaborn as sns
import matplotlib.pyplot as plt
# read the titanic.csv file from seaborn library
kashti=sns.load_dataset("titanic")
sns.barplot(x="class", y="fare", data=kashti, linewidth=3, facecolor=(1,1,1,0), errcolo
```

Out[30]:

```
<AxesSubplot:xlabel='class', ylabel='fare'>
```



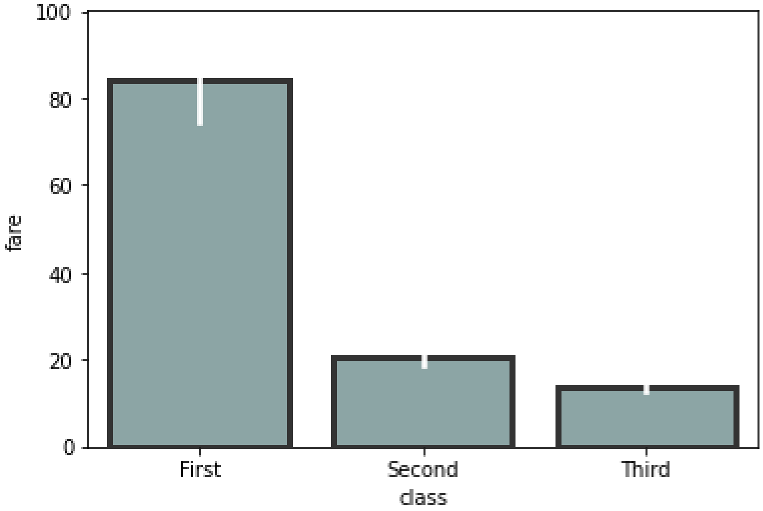
In [32]:

```
# Import the required Library
import seaborn as sns
import matplotlib.pyplot as plt
# read the titanic.csv file from seaborn library
kashti=sns.load_dataset("titanic")
sns.barplot(x="class", y="fare", data=kashti, linewidth=3, facecolor=(0.1,0.3,0.3,0.5),
```

Out[32]:

```
<AxesSubplot:xlabel='class', ylabel='fare'>
```

Loading [MathJax]/extensions/Safe.js



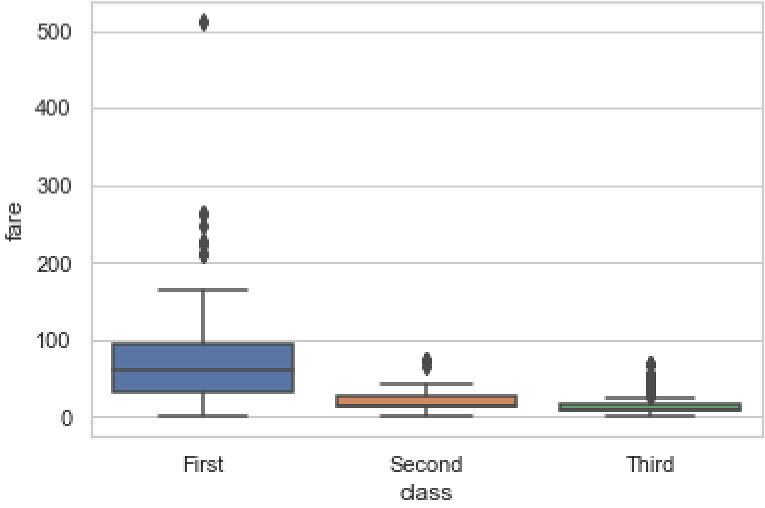
Boxplot

In [1]:

```
# Import Library
import seaborn
# Canvas (Baloon Board)
seaborn.set(style="whitegrid")
# Load Data
kashti=seaborn.load_dataset("titanic")
seaborn.boxplot(x="class",y="fare", data=kashti)
```

Out[1]:

```
<AxesSubplot:xlabel='class', ylabel='fare'>
```

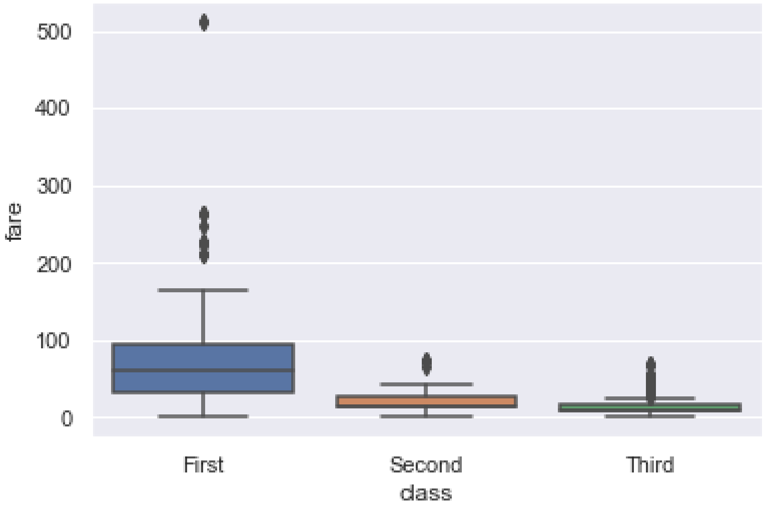


In [2]:

```
# Import Library
import seaborn
# Canvas (Baloon Board)
seaborn.set(style="darkgrid")
# Load Data
kashti=seaborn.load_dataset("titanic")
seaborn.boxplot(x="class",
                y="fare",
                data=kashti)
```

```
<AxesSubplot:xlabel='class', ylabel='fare'>
```

Out[2]:



In [5]:

```
import seaborn
seaborn.set(style='whitegrid')
df=seaborn.load_dataset('titanic')
df
```

Out[5]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	dec
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	Na
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	Na
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	Na
...	...	...	...	...	...	...	...	...	...	...	...	
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	Na
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	Na
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	Na

891 rows × 15 columns

In [7]:

```
import seaborn
seaborn.set(style='whitegrid')
df=seaborn.load_dataset('tips')
df
```

```
---------------------------------------------------------------------------
ConnectionAbortedError                    Traceback (most recent call last)
Loading [MathJax]/extensions/Safe.js urllib\request.py in do_open(self, http_class, req, **http_conn_args)
```

```
   1345                 try:
-> 1346                     h.request(req.get_method(), req.selector, req.data, headers,
   1347                               encode_chunked=req.has_header('Transfer-encoding'))

~\anaconda3\lib\http\client.py in request(self, method, url, body, headers, encode_chunk
ed)
   1278         """Send a complete request to the server."""
-> 1279         self._send_request(method, url, body, headers, encode_chunked)
   1280

~\anaconda3\lib\http\client.py in _send_request(self, method, url, body, headers, encode
_chunked)
   1324             body = _encode(body, 'body')
-> 1325         self.endheaders(body, encode_chunked=encode_chunked)
   1326

~\anaconda3\lib\http\client.py in endheaders(self, message_body, encode_chunked)
   1273             raise CannotSendHeader()
-> 1274         self._send_output(message_body, encode_chunked=encode_chunked)
   1275

~\anaconda3\lib\http\client.py in _send_output(self, message_body, encode_chunked)
   1033         del self._buffer[:]
-> 1034         self.send(msg)
   1035

~\anaconda3\lib\http\client.py in send(self, data)
    973             if self.auto_open:
--> 974                 self.connect()
    975             else:

~\anaconda3\lib\http\client.py in connect(self)
   1447
-> 1448             self.sock = self._context.wrap_socket(self.sock,
   1449                                                   server_hostname=server_hostnam
e)

~\anaconda3\lib\ssl.py in wrap_socket(self, sock, server_side, do_handshake_on_connect,
 suppress_ragged_eofs, server_hostname, session)
    499         # ctx._wrap_socket()
--> 500         return self.sslsocket_class._create(
    501             sock=sock,

~\anaconda3\lib\ssl.py in _create(cls, sock, server_side, do_handshake_on_connect, suppr
ess_ragged_eofs, server_hostname, context, session)
   1039                         raise ValueError("do_handshake_on_connect should not be
 specified for non-blocking sockets")
-> 1040                     self.do_handshake()
   1041             except (OSError, ValueError):

~\anaconda3\lib\ssl.py in do_handshake(self, block)
   1308                 self.settimeout(None)
-> 1309             self._sslobj.do_handshake()
   1310         finally:

ConnectionAbortedError: [WinError 10053] An established connection was aborted by the so
ftware in your host machine
```

```
of the above exception, another exception occurred:
```

```
URLError                                  Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_10892/3864610840.py in <module>
      1 import seaborn
      2 seaborn.set(style='whitegrid')
----> 3 df=seaborn.load_dataset('tips')
      4 df

~\anaconda3\lib\site-packages\seaborn\utils.py in load_dataset(name, cache, data_home, *
*kws)
    592         cache_path = os.path.join(get_data_home(data_home), os.path.basename(ur
l))
    593         if not os.path.exists(cache_path):
--> 594             if name not in get_dataset_names():
    595                 raise ValueError(f"'{name}' is not one of the example
datasets.")
    596             urlretrieve(url, cache_path)

~\anaconda3\lib\site-packages\seaborn\utils.py in get_dataset_names()
    518     """
    519     url = "https://github.com/mwaskom/seaborn-data"
--> 520     with urlopen(url) as resp:
    521         html = resp.read()
    522 

~\anaconda3\lib\urllib\request.py in urlopen(url, data, timeout, cafile, capath, cadefau
lt, context)
    212     else:
    213         opener = _opener
--> 214     return opener.open(url, data, timeout)
    215 
    216 def install_opener(opener):

~\anaconda3\lib\urllib\request.py in open(self, fullurl, data, timeout)
    515 
    516         sys.audit('urllib.Request', req.full_url, req.data, req.headers, req.ge
t_method())
--> 517         response = self._open(req, data)
    518 
    519         # post-process response

~\anaconda3\lib\urllib\request.py in _open(self, req, data)
    532 
    533         protocol = req.type
--> 534         result = self._call_chain(self.handle_open, protocol, protocol +
    535                                   '_open', req)
    536         if result:

~\anaconda3\lib\urllib\request.py in _call_chain(self, chain, kind, meth_name, *args)
    492         for handler in handlers:
    493             func = getattr(handler, meth_name)
--> 494             result = func(*args)
    495             if result is not None:
    496                 return result

~\anaconda3\lib\urllib\request.py in https_open(self, req)
   1387 
   1388         def https_open(self, req):
-> 1389             return self.do_open(http.client.HTTPSConnection, req,
                        context=self._context, check_hostname=self._check_hostname)
   1391 
```

```
~\anaconda3\lib\urllib\request.py in do_open(self, http_class, req, **http_conn_args)
   1347                                encode_chunked=req.has_header('Transfer-encoding'))
   1348             except OSError as err: # timeout error
-> 1349                 raise URLError(err)
   1350             r = h.getresponse()
   1351         except:

URLError: <urlopen error [WinError 10053] An established connection was aborted by the s
oftware in your host machine>
```

In [1]:

```
weight=input("enter your weight")
```

```
enter your weight72
```

In [2]:

```
weight=float(weight)
```

In [3]:

```
height=input("enter your height")
```

```
enter your height1.73
```

In [4]:

```
height=float(height)
```

In [5]:

```
name=input("your name please? ")
```

```
your name please? Kashif
```

In [6]:

```
BMI=weight/height**2
```

In [8]:

```
print("your name is ",name ,"and BMI is ,",BMI)
```

```
your name is  Kashif and BMI is , 24.0569347455645
```

Assignment_1

Box_plot

In [1]:

```
import pandas as pd
df=pd.read_csv("data1.csv")
df
```

Out[1]:

Loading [MathJax]/extensions/Safe.js

	Gender	Location	Age	Qualification_completed	field_of_study	Purpose_for_chilla	What are you?	Blc grc
0	Male	Pakistan	36-40	Masters	Natural Sciences	to boost my skill set	Unemplyed	
1	Male	Pakistan	26-30	Bachelors	CS/IT	to boost my skill set	Student	
2	Male	Pakistan	31-35	Masters	Enginnering	Switch my field of study	Employed	
3	Female	Pakistan	31-35	Masters	CS/IT	to boost my skill set	Employed	
4	Female	Pakistan	26-30	Masters	Enginnering	to boost my skill set	Student	
...	...	...	...	...	...	...	...	
370	Male	Pakistan	26-30	Masters	Enginnering	to boost my skill set	Employed	
371	Male	Pakistan	31-35	Bachelors	Enginnering	to boost my skill set	Employed	
372	Male	Pakistan	21-25	Bachelors	CS/IT	to boost my skill set	Employed	
373	Male	Pakistan	26-30	Masters	Enginnering	to boost my skill set	Employed	
374	Female	Pakistan	31-35	Masters	Mathematics	Switch my field of study	Unemplyed	

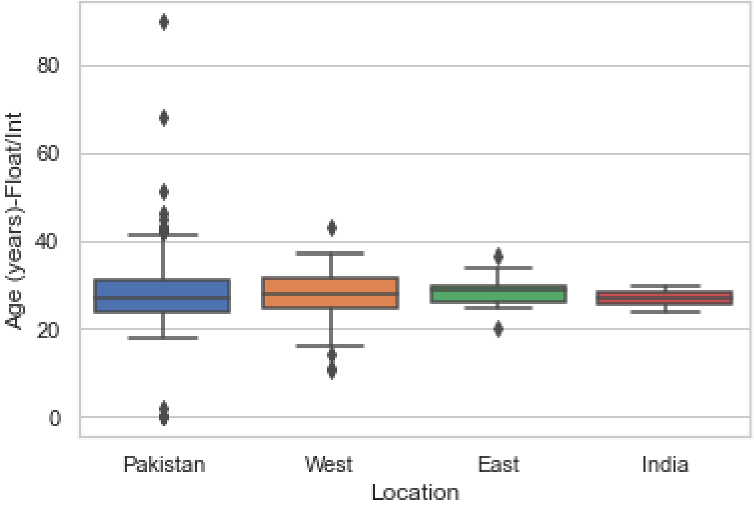
375 rows × 23 columns

In [2]:

```
import pandas as pd
import seaborn as sns
df=pd.read_csv("data1.csv")
sns.set(style='whitegrid')
sns.boxplot(x='Location', y='Age (years)-Float/Int', data=df, saturation=1)
```

Out[2]:

```
<AxesSubplot:xlabel='Location', ylabel='Age (years)-Float/Int'>
```

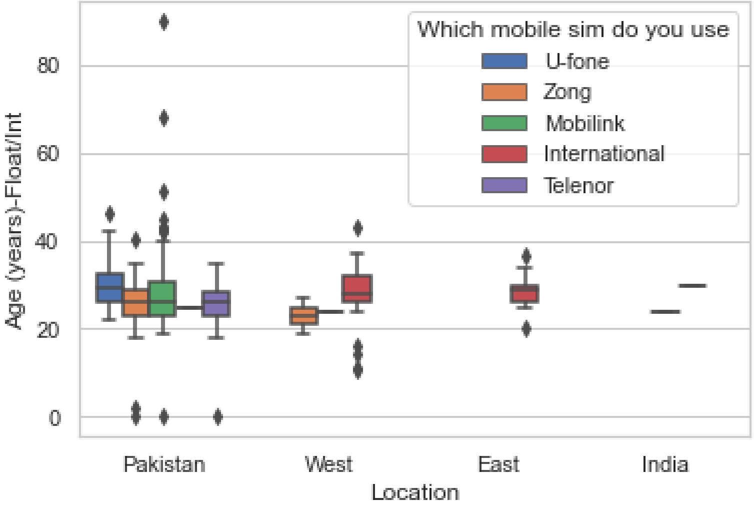


In [3]:

```
import pandas as pd
import numpy
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv("data1.csv")
sns.set(style='whitegrid')
sns.boxplot(x='Location', y='Age (years)-Float/Int', hue='Which mobile sim do you use',
```

Out[3]:

```
<AxesSubplot:xlabel='Location', ylabel='Age (years)-Float/Int'>
```



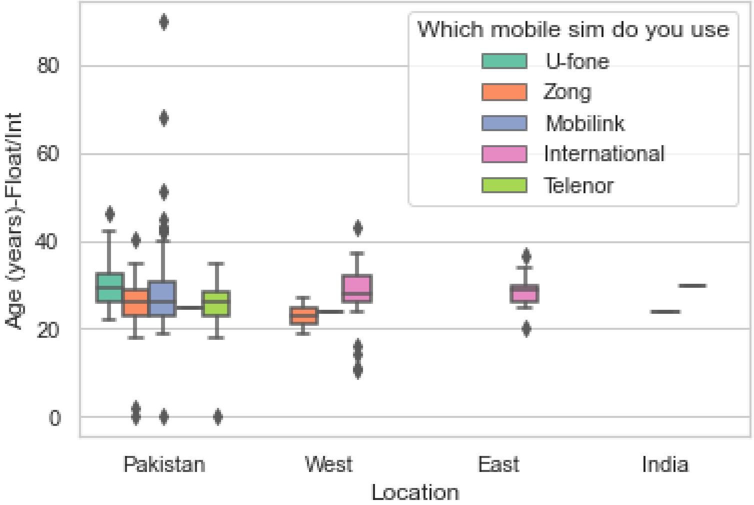
In [4]:

```
import pandas as pd
import numpy
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv("data1.csv")
sns.set(style='whitegrid')
sns.boxplot(x='Location', y='Age (years)-Float/Int', hue='Which mobile sim do you use',
```

Out[4]:

```
<AxesSubplot:xlabel='Location', ylabel='Age (years)-Float/Int'>
```

Loading [MathJax]/extensions/Safe.js

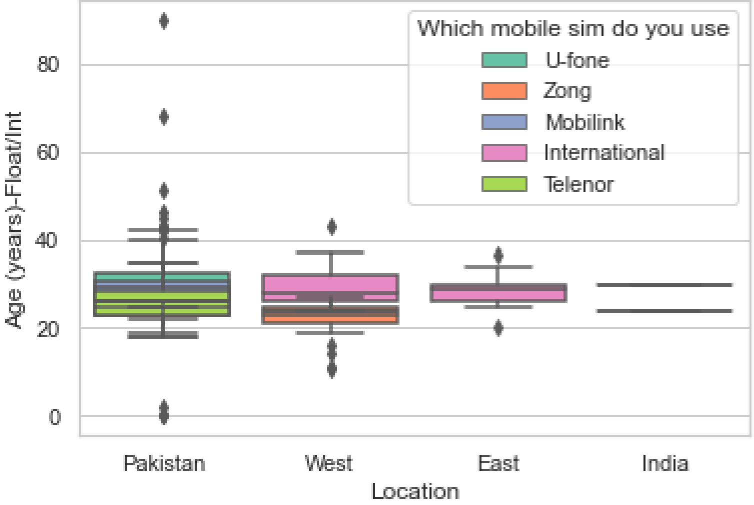


In [5]:

```
import pandas as pd
import numpy
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv("data1.csv")
sns.set(style='whitegrid')
sns.boxplot(x='Location', y='Age (years)-Float/Int', hue='Which mobile sim do you use',
```

Out[5]:

```
<AxesSubplot:xlabel='Location', ylabel='Age (years)-Float/Int'>
```



In [6]:

```
import pandas as pd
import seaborn as sns
df=pd.read_csv("data1.csv")
sns.set(style='whitegrid')
sns.boxplot(x='Gender', y='Age (years)-Float/Int', data=df, saturation=1, color=)
```

```
  File "C:\Users\User\AppData\Local\Temp/ipykernel_8772/3878837766.py", line 5
    sns.boxplot(x='Gender', y='Age (years)-Float/Int', data=df, saturation=1, color=)
                                                                                     ^
SyntaxError: invalid syntax
```

In [ ]:

Loading [MathJax]/extensions/Safe.js

```
s pd
import numpy
import seaborn as sns
df=pd.read_csv("data1.csv")
df.head()
```

In [ ]:

```
import pandas as pd
import numpy
import seaborn as sns
df=pd.read_csv("data1.csv")
sns.set(style='whitegrid')
sns.boxplot(x='Gender', y='Age (years)-Float/Int', data=df, saturation=1, color='red',s
```

In [ ]:

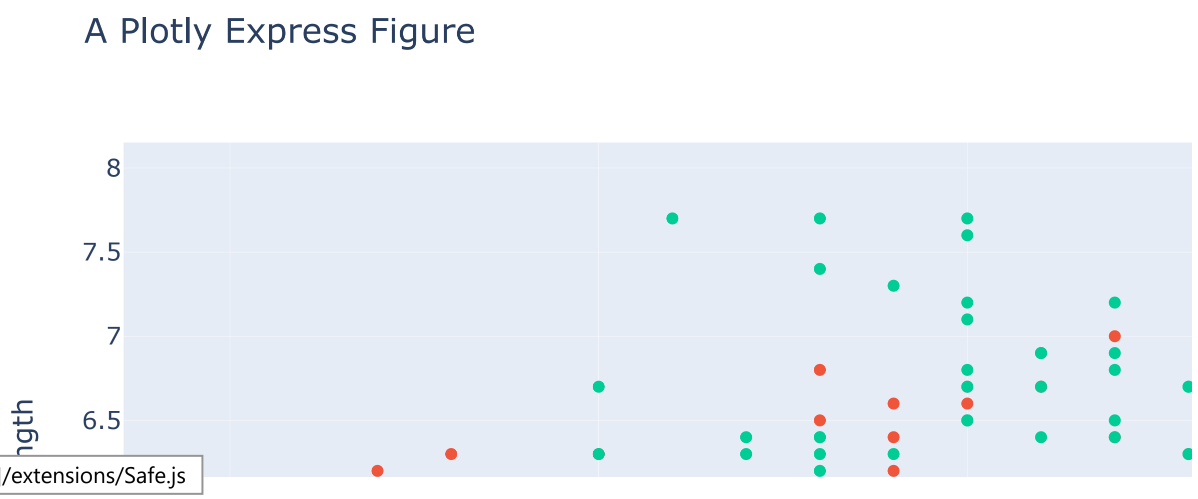
```
import pandas as pd
import numpy
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv("data1.csv")
sns.set(style='whitegrid')
sns.boxplot(x='Gender', y='Age (years)-Float/Int', data=df, saturation=1, color='red',s
# Show Label
plt.xlabel('gender of participant')
plt.ylabel('Age of participant')
plt.title('Plot of Assignment Data')
plt.show()
```

In [ ]:

```
pip install plotly
```

In [7]:

```
import plotly.express as px

df = px.data.iris()
fig = px.scatter(df, x="sepal_width", y="sepal_length", color="species", title="A Plotl

# If you print the figure, you'll see that it's just a regular figure with data and Lay
# print(fig)

fig.show()
```



Loading [MathJax]/extensions/Safe.js

In [9]:

```
import plotly.express as px

df = px.data.election()
geojson = px.data.election_geojson()

fig = px.choropleth_mapbox(df, geojson=geojson, color="Bergeron",
                           locations="district", featureidkey="properties.district",
                           center={"lat": 45.5517, "lon": -73.7073},
                           mapbox_style="carto-positron", zoom=9)
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

Loading [MathJax]/extensions/Safe.js

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```