In [82]:
```
#Import Libraries

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
sns.set_style('whitegrid') #seaborn style to set the default for all plots
```

In [83]:
```
#Extracting CSV Dataset From System using Pandas Library

data = pd.read_csv("C:\Projects\Heart Disease Diagnostic Analysis\Heart Disease data 2.csv")
data
```

Out[83]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

In [84]:
```
#All Columns

data.columns
```

Out[84]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

Attribute Information:

age: The person's age in years

sex: 1 = male, 0 = female

cp:chest pain type (4 values)

trestbps: resting blood pressure

chol:serum cholestoral in mg/dl

fbs:fasting blood sugar > 120 mg/dl ; 1 = true; 0 = false

restecg: resting electrocardiographic results (values 0,1,2)

thalach:maximum heart rate achieved

exang:exercise induced angina (1 = yes; 0 = no)

oldpeak = ST depression induced by exercise relative to rest

slope: the slope of the peak exercise ST segment

ca: number of major vessels (0-3) colored by flourosopy

thal: 1= normal; 2 = fixed defect; 3= reversable defect

target: 0 = no, 1 = yes

In [85]:
```
#Check NULL

data.isnull().sum()
```

Out[85]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

NO MISSING Values

Percentage of people having Heart Disease

In [86]:
```
target=data.groupby('target').size()
target
```

Out[86]:
```
target
0    499
1    526
dtype: int64
```

In [87]:
```
# Numerical to Categorical

def heart_disease(row):
    if row == 0:
        return 'Absence'
    elif row == 1:
        return 'Presence'
```

In [88]:
```
#new column - Heart_Disease

data['Heart_Disease']=data['target'].apply(heart_disease)
data.head() #The head() method returns the first 5 rows if a number is not specified.
```

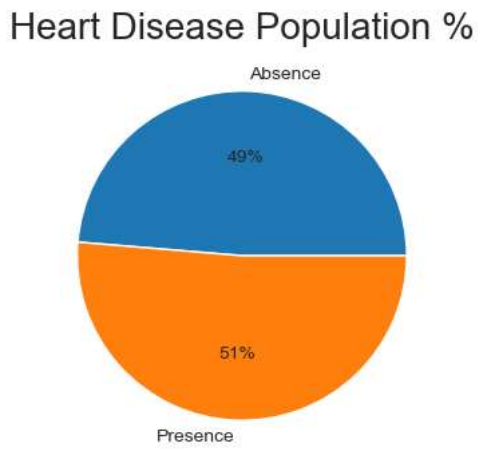
Out[88]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	Heart_Disease
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0	Absence
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0	Absence
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0	Absence
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0	Absence
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0	Absence

In [89]:
```
hd=data.groupby('Heart_Disease')['target'].count()
print(hd)
```

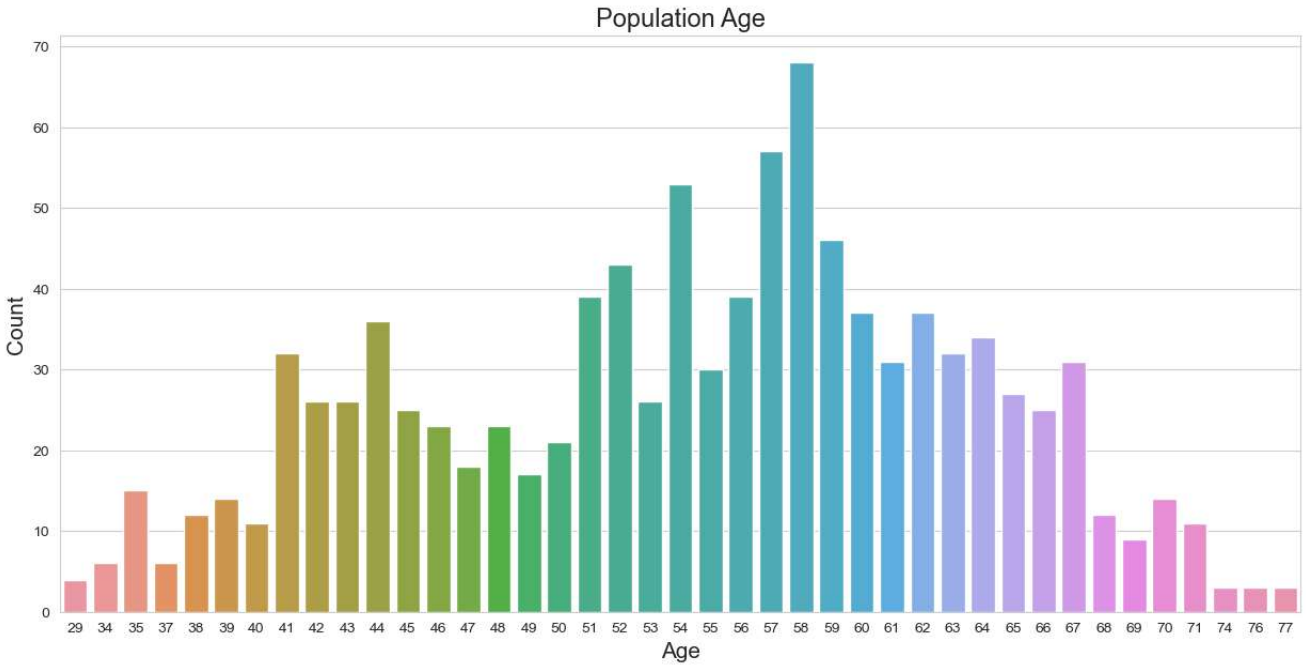
```
Heart_Disease
Absence     499
Presence    526
Name: target, dtype: int64
```

In [90]:
```
#Pie Chart of Heart Disease Population % via MatplotLib
#autopct:to label the wedges with their numeric value ; if 0.2f% then 2 decimal place

plt.figure(figsize=(9,4))
plt.pie(hd, labels=['Absence','Presence'], autopct='%0.0f%%')
plt.title('Heart Disease Population %', fontsize=20)
plt.show()
```



AGE:

In [91]:
```
#Countplot-Population Age via MatplotLib and Seaborn

plt.figure(figsize=(15,7))
sns.countplot(x='age', data=data)
plt.title('Population Age', fontsize=17)

plt.xlabel('Age', fontsize=15)
plt.ylabel('Count', fontsize=15)
plt.show()
```



In [92]:
```
#Statistical Analysis

Min_Age=data['age'].min()
Max_Age=data['age'].max()
Mean_Age=data['age'].mean()

print("Minimum Age =",Min_Age)
print("Maximum Age =",Max_Age)
print("Mean Age =",Mean_Age)
```

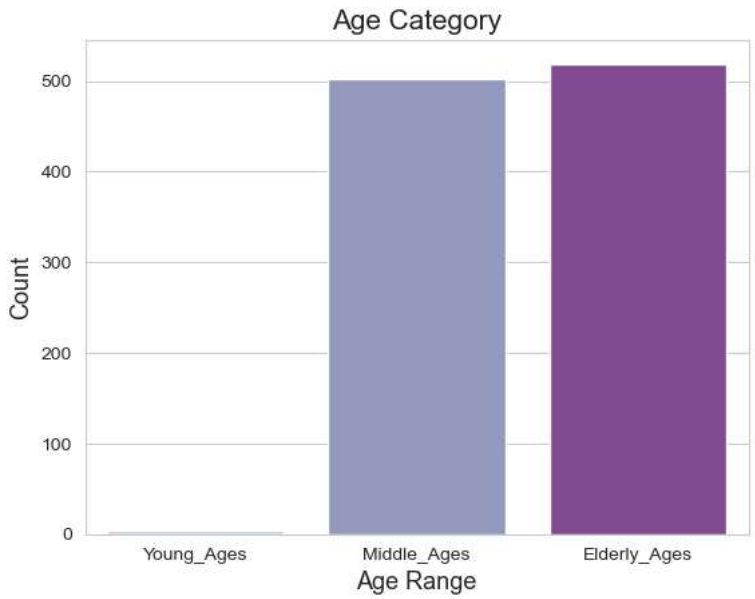
```
Minimum Age = 29
Maximum Age = 77
Mean Age = 54.43414634146342
```

the best analysis can be divided into the elderly,middle-aged, young people by looking at the age ranges.

In [93]:
```
#Categorical Analysis

Young_Ages = data [(data['age']<30)]
Middle_Ages = data [(data['age']>=30) & (data['age']<=55)]
Elderly_Ages = data [(data['age']>55)]

print('Young Ages =',len(Young_Ages))
print('Middle Ages =',len(Middle_Ages))
print('Elderly Ages =',len(Elderly_Ages))
```

```
Young Ages = 4
Middle Ages = 502
Elderly Ages = 519
```

In [94]:
```
#Bar Plot

sns.barplot(x=['Young_Ages','Middle_Ages','Elderly_Ages'], y=[len(Young_Ages), len(Middle_Ages), len(Elderly_Ages)], palette='BuP

plt.title('Age Category', fontsize=15)
plt.xlabel('Age Range', fontsize=13)
plt.ylabel('Count', fontsize=13)
plt.show()
```



In [95]:
```
#Numerical Data into Categorical Data

def gender(row):
    if row==1:
        return 'Male'
    elif row==0:
        return 'Female'
```

In [96]:
```
#new column - sex1

data['sex1']=data['sex'].apply(gender)
data.head()
```

Out[96]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	Heart_Disease	sex1
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0	Absence	Male
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0	Absence	Male
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0	Absence	Male
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0	Absence	Male
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0	Absence	Female

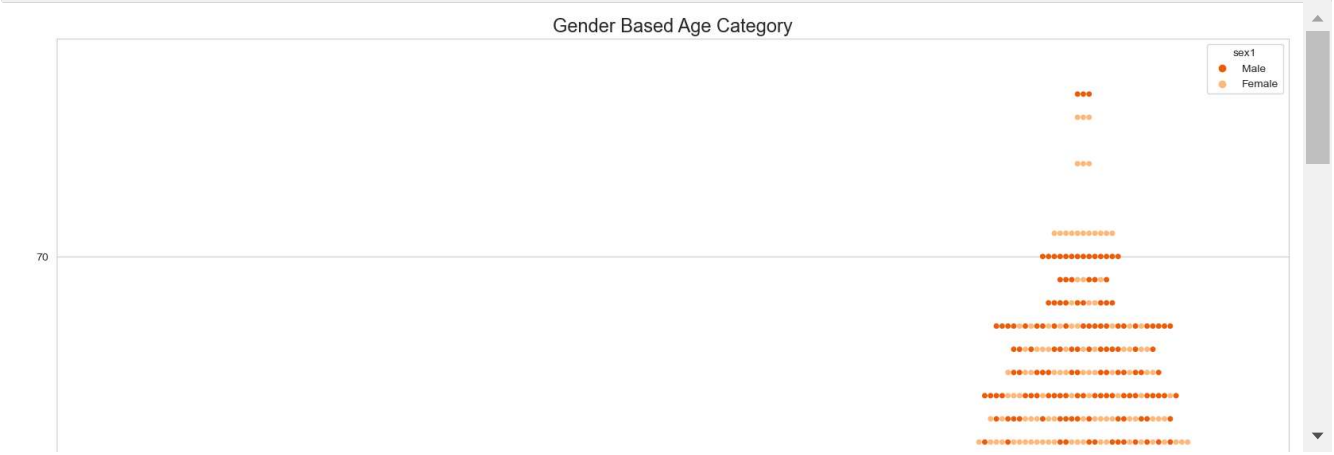
In [97]:
```
#Numerical Data into Categorical Data

def age_range(row):
    if row<30:
        return 'Young Age'
    elif row>=30 and row<=55:
        return 'Middle Age'
    elif row>55:
        return 'Elder Age'
```

In [98]:
```
#new column - Age_Range

data['Age_Range']=data['age'].apply(age_range)
data.head()
```

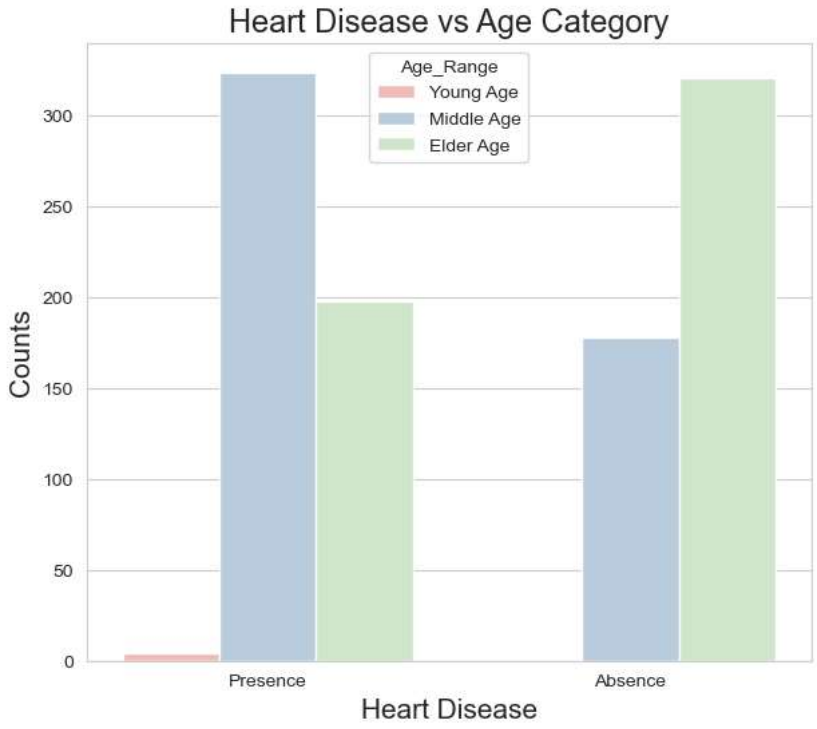
Out[98]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	Heart_Disease	sex1	Age_Range
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0	Absence	Male	Middle Age
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0	Absence	Male	Middle Age
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0	Absence	Male	Elder Age
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0	Absence	Male	Elder Age
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0	Absence	Female	Elder Age

In [109]:
```
#Swarm Plot of Gender Based Age via MatplotLib and Seaborn

plt.figure(figsize=(20,20))
sns.swarmplot(x='Age_Range', y='age', hue='sex1', data=data, order=['Young Age','Middle Age','Elder Age'], palette='Oranges_r')
plt.title('Gender Based Age Category', fontsize=17)
plt.xlabel('Age Category', fontsize=15)
plt.ylabel('Age', fontsize=15)
plt.show()
```

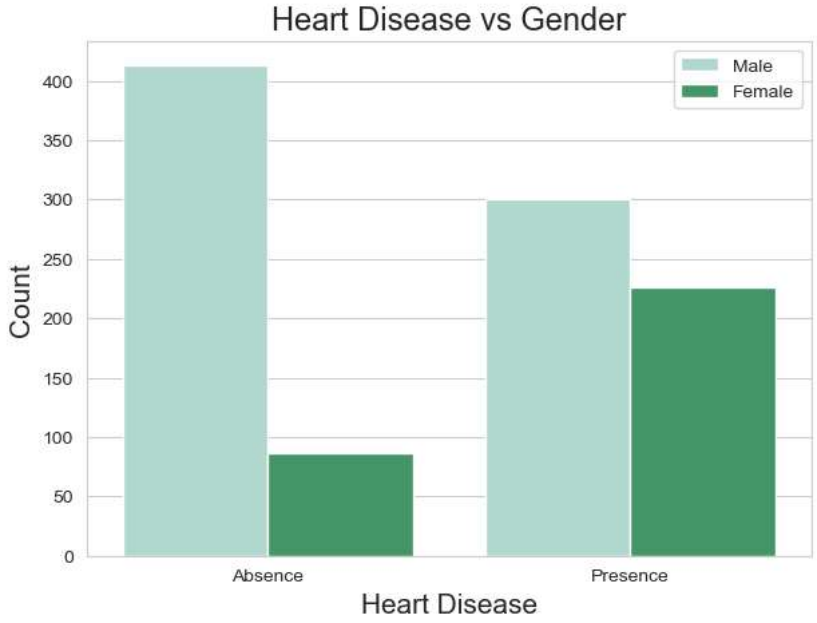


In [117]:
```
#Count Plot of Heart Disease Based On Age via MatplotLib and Seaborn

plt.figure(figsize=(7,6))
hue_order=['Young Age', 'Middle Age', 'Elder Age']
sns.countplot(x='Heart_Disease', hue='Age_Range', data=data, order=['Presence','Absence'], hue_order=hue_order, palette='Pastel1'
plt.title('Heart Disease vs Age Category', fontsize=17)
plt.xlabel('Heart Disease', fontsize=15)
plt.ylabel('Counts', fontsize=15)
plt.show()
```



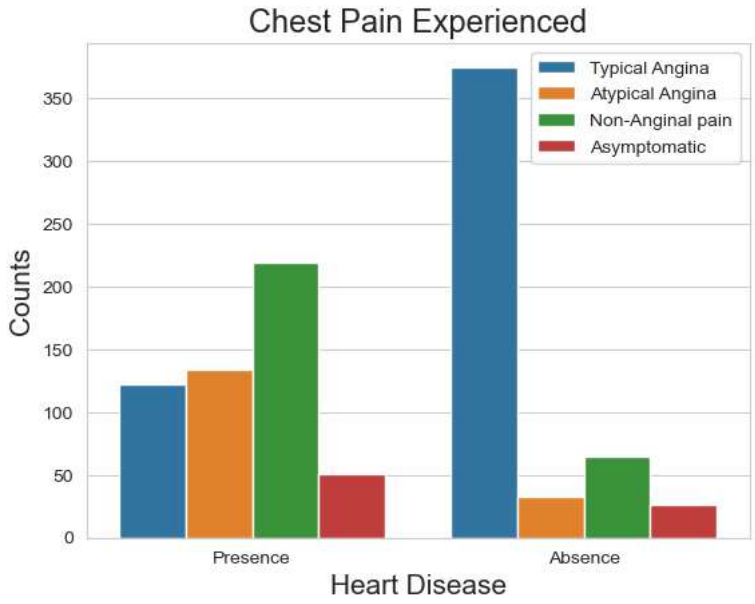
Middle Age People are most affected by Heart Disease AND Young Age People are mostly FREE from any kind of Heart Disease

In [116]:
```
#Count Plot of Heart Disease Based on Gender via MatplotLib and Seaborn

plt.figure(figsize=(7,5))
sns.countplot(x=data['Heart_Disease'], hue='sex1', data=data, palette='BuGn')
plt.xlabel('Heart Disease', fontsize=15)
plt.ylabel('Count',fontsize=15)
plt.legend(labels=['Male','Female'])
plt.title('Heart Disease vs Gender',fontsize=17)
plt.show()
```



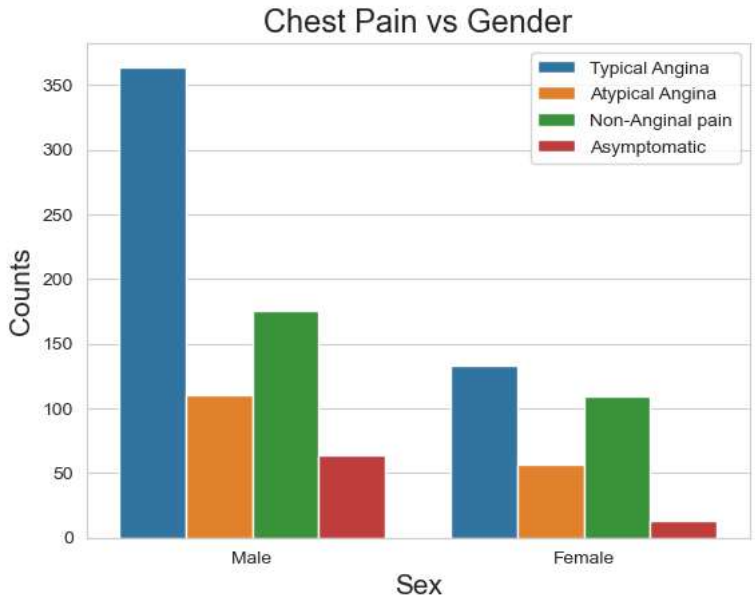
Male are more prone to Heart Disease

In [112]:
```
#Count Plot of Chest Pain Experienced using MatplotLib and Seaborn

sns.countplot(x=data['Heart_Disease'], hue='cp', data=data, order=['Presence','Absence'])
plt.title('Chest Pain Experienced', fontsize=17)
plt.xlabel('Heart Disease',fontsize=15)
plt.ylabel('Counts',fontsize=15)
plt.legend(labels=['Typical Angina','Atypical Angina','Non-Anginal pain','Asymptomatic'])
plt.show()
```



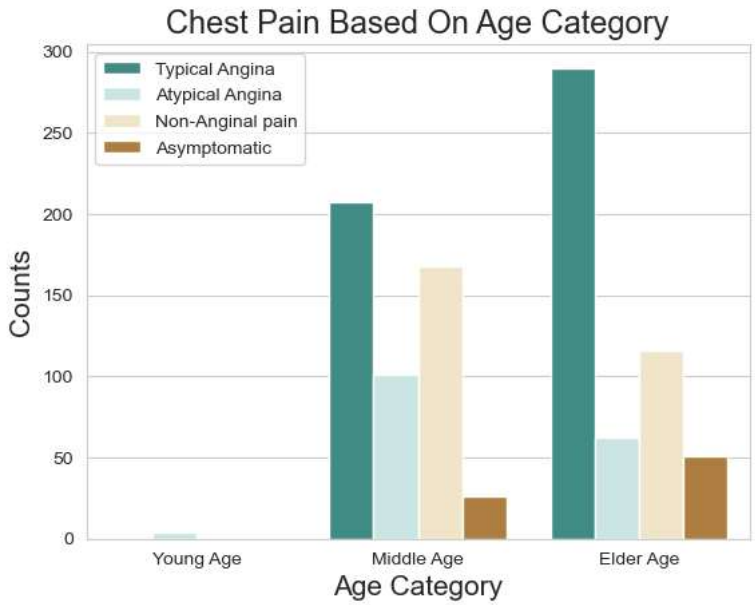
It seems people having non anginal pain have a higher chance of heart disease

In [122]:
```
#Count Plot of Chest Pain Based On Gender via MatplotLib and Seaborn

sns.countplot(x=data['sex1'], hue='cp', data=data)
plt.title('Chest Pain vs Gender', fontsize=17)
plt.xlabel('Sex', fontsize=15)
plt.ylabel('Counts', fontsize=15)
plt.legend(labels=['Typical Angina','Atypical Angina','Non-Anginal pain','Asymptomatic'])
plt.show()
```



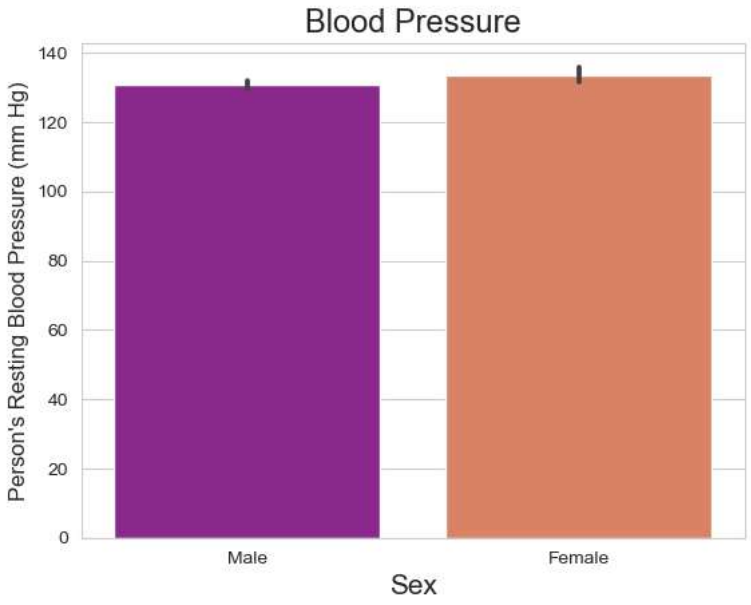
Higher number of men are suffering from Typical Angina

In [121]:
```
#Count Plot of Chest Pain Based On Age via MatplotLib and Seaborn

sns.countplot(x=data['Age_Range'], hue='cp', data=data, order=['Young Age', 'Middle Age', 'Elder Age'], palette='BrBG_r')
plt.title('Chest Pain Based On Age Category', fontsize=17)
plt.xlabel('Age Category', fontsize=15)
plt.ylabel('Counts', fontsize=15)
plt.legend(labels=['Typical Angina','Atypical Angina','Non-Anginal pain','Asymptomatic'])
plt.show()
```



High number of Typical Angina in Elderly age

In [124]:
```
#Bar Plot Creation of Person's Resting Blood Pressure (mm Hg)

sns.barplot(x='sex1', y='trestbps', data=data, palette='plasma')
plt.title("Blood Pressure", fontsize=17)
plt.xlabel('Sex',fontsize=15)
plt.ylabel("Person's Resting Blood Pressure (mm Hg)", fontsize=12)
plt.show()
```

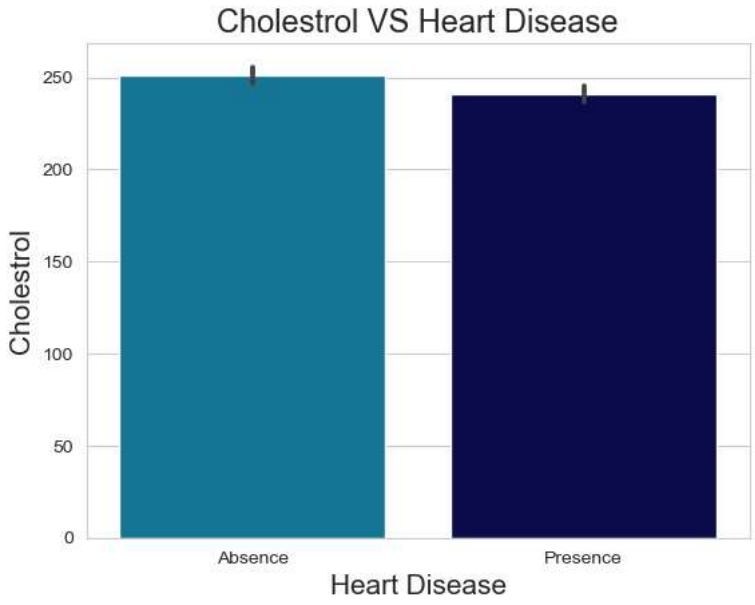


Blood Pressure is almost equal in Male and Female

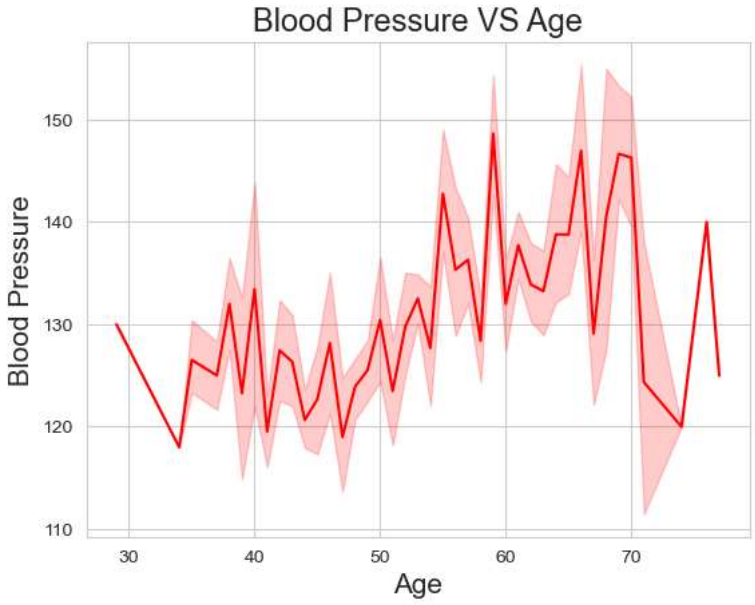
In [125]:
```
#Bar Plot of Cholestrol Level Based On Gender via MatplotLib and Seaborn

sns.barplot(x='sex1', y='chol', data=data, palette='turbo')
plt.title("Cholestrol Level Based On Gender", fontsize=17)
plt.xlabel('Sex',fontsize=15)
plt.ylabel("Cholestrol", fontsize=15)
plt.show()
```

female have higher cholesterol than male

In [126]:
```
#Bar Plot of Cholestrol VS Heart Disease via MatplotLib and Seaborn

sns.barplot(x='Heart_Disease', y='chol', data=data, palette='ocean_r')
plt.title('Cholestrol VS Heart Disease', fontsize=17)
plt.xlabel('Heart Disease', fontsize=15)
plt.ylabel('Cholestrol', fontsize=15)
plt.show()
```

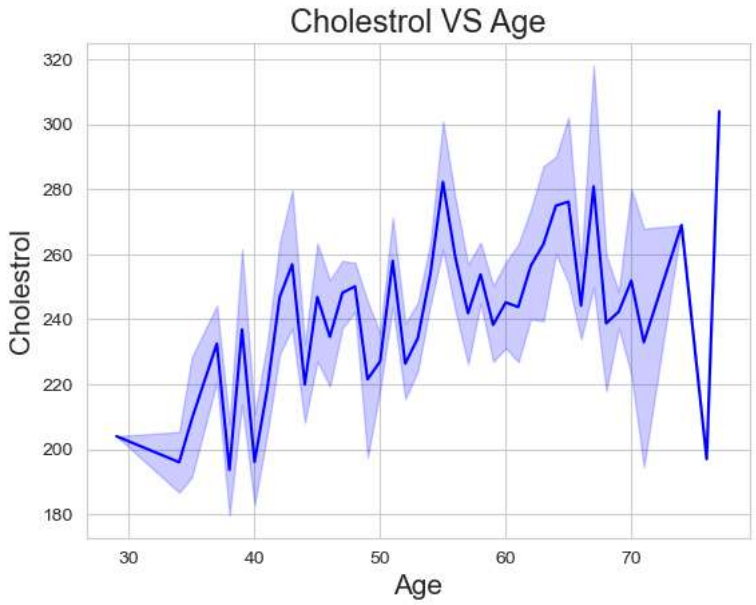


In [129]:
```
#Line Plot of Blood Pressure VS Age via MatplotLib and Seaborn

sns.lineplot(x='age', y='trestbps', data=data, color='r')
plt.title('Blood Pressure VS Age', fontsize=17)
plt.xlabel('Age', fontsize=15)
plt.ylabel('Blood Pressure', fontsize=15)
plt.show()
```



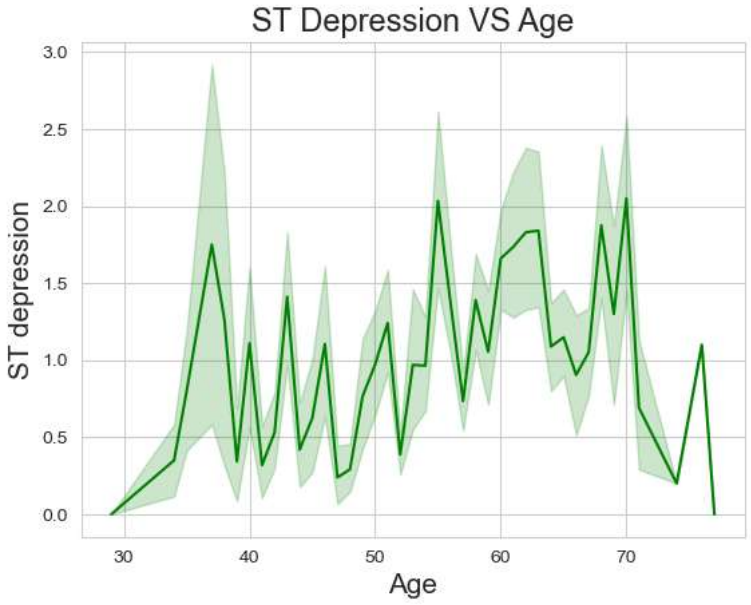
Blood Pressure increases between age of 50 to 60 and reaches peak in elderly people.

In [130]:
```
#Line Plot of Cholestrol VS Age via MatplotLib and Seaborn

sns.lineplot(x='age', y='chol', data=data, color='b')
plt.title('Cholestrol VS Age', fontsize=17)
plt.xlabel('Age', fontsize=15)
plt.ylabel('Cholestrol', fontsize=15)
plt.show()
```



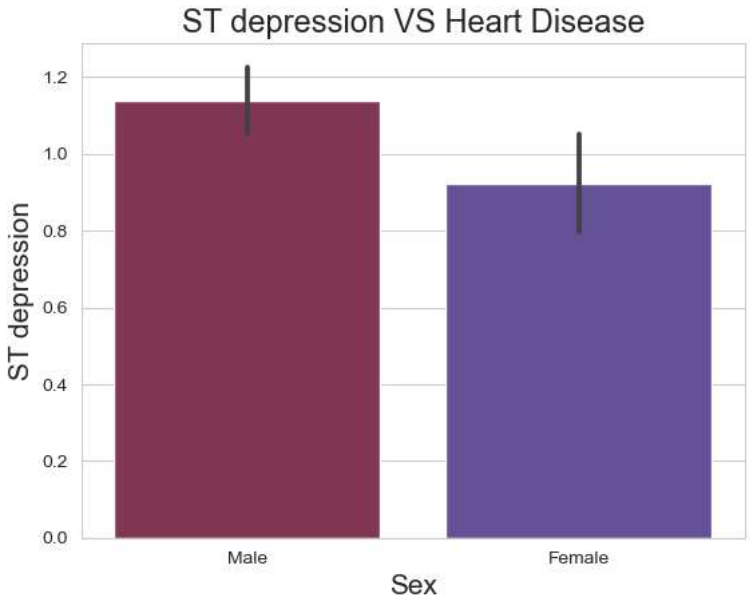
Similarly Cholestrol Increases in the age group of 50-60 and reaches peak in elderly age.

In [133]:
```
#Line Plot of ST Depression VS Age via MatplotLib and Seaborn

sns.lineplot(x='age', y='oldpeak', data=data, color='g')
plt.title('ST Depression VS Age', fontsize=17)
plt.xlabel('Age', fontsize=15)
plt.ylabel('ST depression', fontsize=15)
plt.show()
```



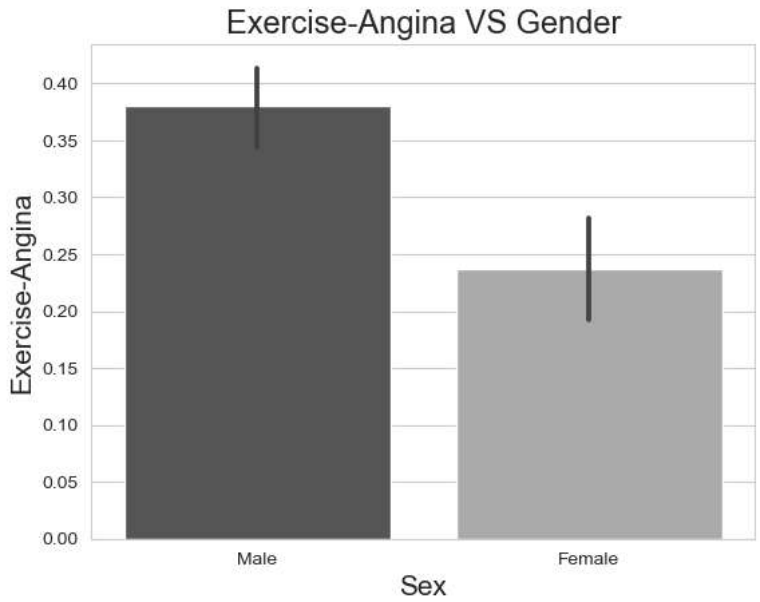
ST depression is high in age group: 30-40

In [137]:
```
#Bar Plot of ST depression VS Heart Disease via MatplotLib and Seaborn

sns.barplot(x='sex1', y='oldpeak', data=data, palette='twilight_r')
plt.title('ST depression VS Heart Disease', fontsize=17)
plt.xlabel('Sex', fontsize=15)
plt.ylabel('ST depression', fontsize=15)
plt.show()
```



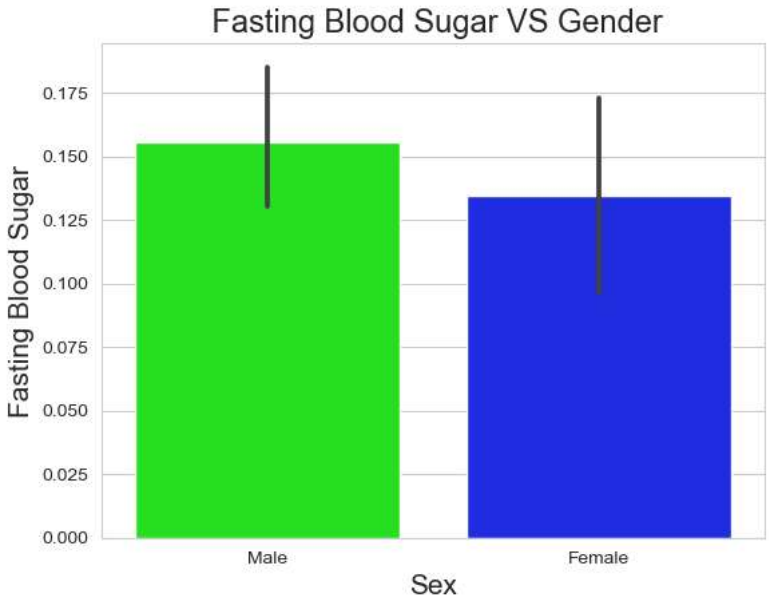
Comparatively male are prone to ST depression than females

In [142]:
```
#Bar Plot of Exercise With Angina VS Gender via MatplotLib and Seaborn

sns.barplot(x='sex1', y='exang', data=data, palette='binary_r')
plt.title('Exercise-Angina VS Gender', fontsize=17)
plt.xlabel('Sex', fontsize=15)
plt.ylabel('Exercise-Angina', fontsize=15)
plt.show()
```



-> Male suffer from Angina during exercise

In [143]:
```
#Bar Plot of Fasting Blood Sugar VS Gender via MatplotLib and Seaborn

sns.barplot(y='fbs', x='sex1', data=data, palette='hsv')
plt.title(' Fasting Blood Sugar VS Gender', fontsize=17)
plt.xlabel('Sex', fontsize=15)
plt.ylabel('Fasting Blood Sugar', fontsize=15)
plt.show()
```



Male have high no. of Fasting Blood Sugar over 120

In [156]:
```
#Heatmap Creation using Seaborn

plt.figure(figsize=(16, 9))
sns.heatmap(data.corr(numeric_only=True), annot=True, linewidth=3)
plt.show()
```

In [ ]: