Telecom Churn Analysis

In [119]:
```
# Import Libraries !!!

import matplotlib.pyplot as plt       # Plot Graphs and Graphical representatio.
import pandas as pd                   # Apply DataFrame (rows and columns) based operations.
%matplotlib inline
                                      # Graphs shows below the cell not in other window or editor.
import numpy as np                    # Use for Numerical operations perform in DataFrame
import math                           # Use For ceil() function.
import seaborn as sns
```

In [3]:
```
df = pd.read_csv("telecom-churn.csv")    # Read the dataset file in .csv formate.
```

In [4]:
```
df.head(2)
```

Out[4]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecu
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes

2 rows × 21 columns

In [5]:
```
df.shape
```

Out[5]: (7043, 21)

In [6]:
```
df.columns
```

Out[6]:
```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'],
      dtype='object')
```

In [7]:
```
df.isnull().values.any()       # Check the Missing Values in records
```

Out[7]:
```
False
```

Change the Attribute(TotalCharges) String to Float.

In [8]:
```
df["TotalCharges"]=df.TotalCharges.convert_objects(convert_numeric=True)
```

```
C:\Users\Sachin\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: FutureWarning: convert_objects is deprec
ated.  To re-infer data dtypes for object columns, use Series.infer_objects()
For all other conversions use the data-type specific converters pd.to_datetime, pd.to_timedelta and pd.to_num
eric.
  """Entry point for launching an IPython kernel.
```

Data Cleansing: Redesign the data into a usable/functional format and correct/remove any bad data !!!

We will Change 'No phone service' to 'No' for column "MultipleLines"

In [9]:
```
df["MultipleLines"].unique()
```

Out[9]:
```
array(['No phone service', 'No', 'Yes'], dtype=object)
```

In [10]:
```
df.MultipleLines.replace(['No phone service'],["No"],inplace = True )
df["MultipleLines"].unique()
```

Out[10]: array(['No', 'Yes'], dtype=object)

We will Change 'No internet service' to 'No' for column "OnlineSecurity" .

In [11]:
```
df["OnlineSecurity"].unique()
```

Out[11]: array(['No', 'Yes', 'No internet service'], dtype=object)

In [12]:
```
df["OnlineSecurity"].replace(['No internet service'],['No'],inplace= True)
df["OnlineSecurity"].unique()
```

Out[12]: array(['No', 'Yes'], dtype=object)

We will Change 'No internet service' to 'No' for column "OnlineBackup" .

In [13]:
```
df["OnlineBackup"].unique()
```

Out[13]: array(['Yes', 'No', 'No internet service'], dtype=object)

In [14]:
```
df["OnlineBackup"].replace(['No internet service'],['No'],inplace= True)
df["OnlineBackup"].unique()
```

Out[14]: array(['Yes', 'No'], dtype=object)

We will Change 'No internet service' to 'No' for column "DeviceProtection"

In [15]:
```
df["DeviceProtection"].unique()
```

Out[15]: array(['No', 'Yes', 'No internet service'], dtype=object)

In [16]:
```
df["DeviceProtection"].replace(['No internet service'],['No'],inplace= True)
df["DeviceProtection"].unique()
```

Out[16]:
```
array(['No', 'Yes'], dtype=object)
```

We will Change 'No internet service' to 'No' for column "TechSupport"

In [17]:
```
df["TechSupport"].unique()
```

Out[17]:
```
array(['No', 'Yes', 'No internet service'], dtype=object)
```

In [18]:
```
df["TechSupport"].replace(['No internet service'],['No'],inplace= True)
df["TechSupport"].unique()
```

Out[18]:
```
array(['No', 'Yes'], dtype=object)
```

We will Change 'No internet service' to 'No' for column "StreamingTV"

In [19]:
```
df["StreamingTV"].unique()
```

Out[19]:
```
array(['No', 'Yes', 'No internet service'], dtype=object)
```

In [20]:
```
df["StreamingTV"].replace(['No internet service'],['No'],inplace= True)
df["StreamingTV"].unique()
```

Out[20]:
```
array(['No', 'Yes'], dtype=object)
```

We will Change 'No internet service' to 'No' for column "StreamingMovies"

In [21]:
```
df["StreamingMovies"].unique()
```

Out[21]:
```
array(['No', 'Yes', 'No internet service'], dtype=object)
```

In [22]:
```
df["StreamingMovies"].replace(['No internet service'],['No'],inplace= True)
df["StreamingMovies"].unique()
```

Out[22]:
```
array(['No', 'Yes'], dtype=object)
```

Change the Tenure (Months wise) to tenure (Year wise).

In [23]:
```
df["tenure"].nunique()
```

Out[23]:
```
73
```

In [24]:
```
df.tenure.describe()
```

Out[24]:
```
count    7043.000000
mean       32.371149
std        24.559481
min         0.000000
25%         9.000000
50%        29.000000
75%        55.000000
max        72.000000
Name: tenure, dtype: float64
```

Minimum tenure Month is 0 and Maximum tenure Month is 72. So we change the tenure in months to Year wise duration.

In [25]:
```
df['tenure_yr']=df["tenure"]/12                    # Months covert in Years.
```

In [26]:
```
df.head(2)
```

Out[26]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecu
0	7590-VHVEG	Female	0	Yes	No	1	No	No	DSL	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes

2 rows × 22 columns

In [27]:
```
df["tenure_year"]= np.ceil(df.tenure_yr)        # Months covert in Years range.
```

In [28]:
```
df.head(2)
```

Out[28]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecu
0	7590-VHVEG	Female	0	Yes	No	1	No	No	DSL	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes

2 rows × 23 columns

Remove Columns we do not need for the analysis .

In [29]:
```
df=df.drop(["customerID"], axis=1)     # customerID has does not use in analysis
                                           # axis= 1(columns) or 0(index/rows)
```

How many Churn of total customer.

In [30]:
```
gen=df.groupby(["Churn"]).count();gen
```

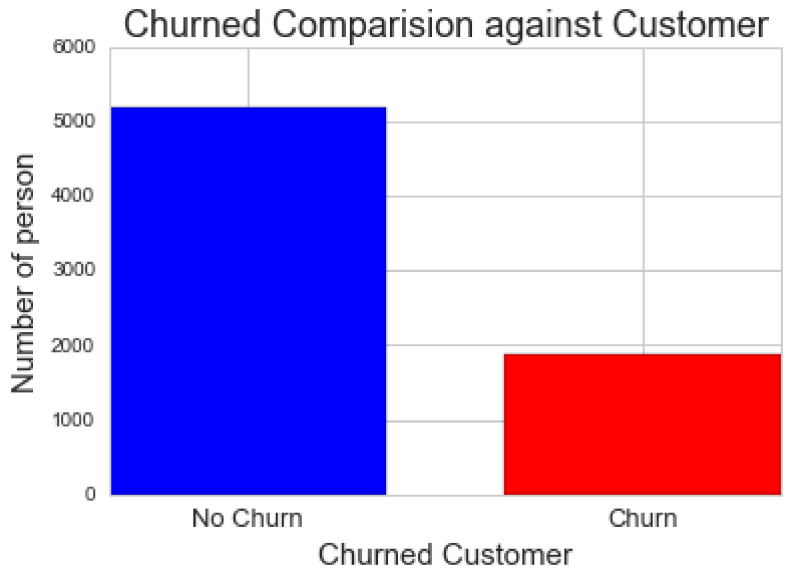
Out[30]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	On
Churn										
No	5174	5174	5174	5174	5174	5174	5174	5174	5174	51
Yes	1869	1869	1869	1869	1869	1869	1869	1869	1869	18

2 rows × 21 columns

In [120]:
```
plt.bar(["No","Yes"],gen["gender"].tolist() , align="center", width=0.7 ,color=['blue', 'red'],alpha=1 )
plt.title("Churned Comparision against Customer",size=18)
plt.ylabel("Number of person",size=15)
plt.xlabel("Churned Customer",size=15)
plt.xticks(["No","Yes"],["No Churn", "Churn"],size=13)
sns.set(style="whitegrid")
df["Churn"].value_counts()
```

Out[120]:
```
No     5174
Yes    1869
Name: Churn, dtype: int64
```



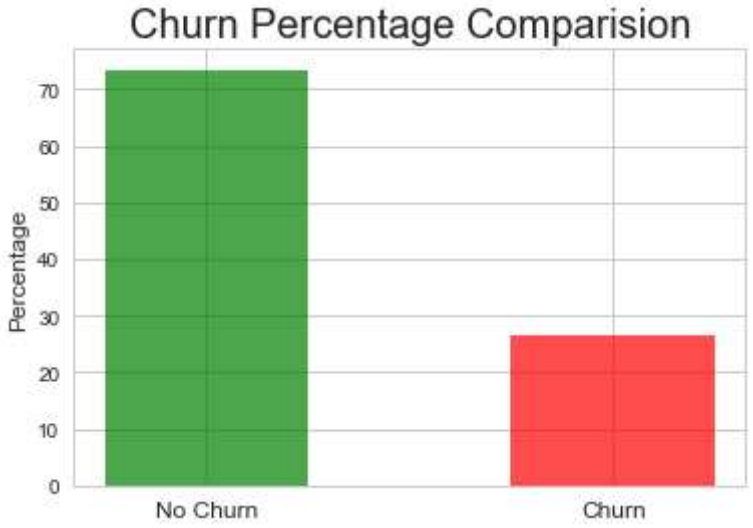
The number of Churn Customer is less.

Bar Plots at percentage wise of attributes using matplotlib !!!

Churn Customer total percentage comparision.

In [32]:
```
churn_per= df.Churn.value_counts() / len(df.Churn) *100
plt.bar(["No","Yes"],churn_per , align="center", width=0.5 ,color=['green', 'red'],alpha=0.7 )
plt.title("Churn Percentage Comparision",size=20)
plt.xticks([0,1],["No Churn","Churn"],size=12)
plt.ylabel("Percentage",size=12)
churn_per
```

Out[32]:
```
No     73.463013
Yes    26.536987
Name: Churn, dtype: float64
```

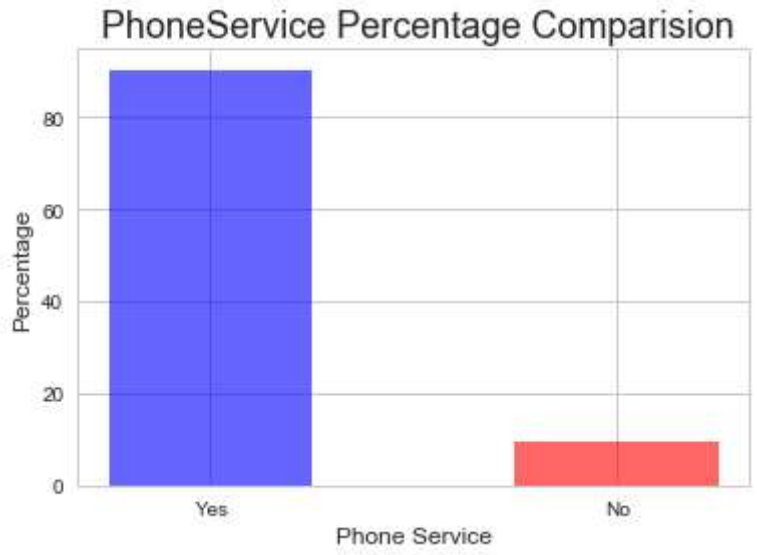


The Churn Customer is 26.53% of the total record. We Focused on that person.

How many percentage of customer use Phone Service or Not ?

In [33]:
```
PhoneService_per= df.PhoneService.value_counts() / len(df.PhoneService) *100
plt.bar(["No","Yes"],PhoneService_per,align='center', width=0.5 ,color=['blue', 'red'], alpha=0.6 )
plt.title(" PhoneService Percentage Comparision" , size=18)
plt.xlabel("Phone Service" , size=12)
plt.xticks([0,1],["Yes","No"])
plt.ylabel("Percentage",size=12)
PhoneService_per
```

Out[33]:
```
Yes    90.316626
No      9.683374
Name: PhoneService, dtype: float64
```

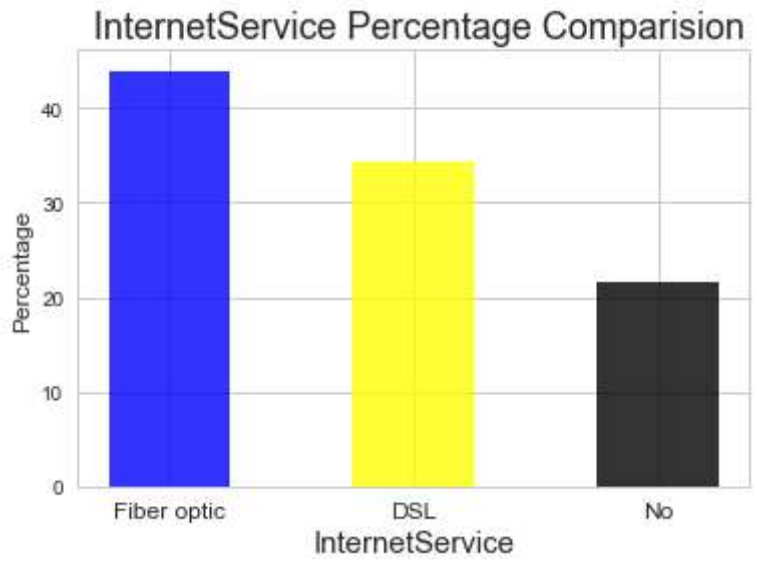


The Phone Service usage by Customer has maximum in nature (above 90%) , its very less number of customer have does not use to Phone Service.

Which Internet Service use by customer ,percentage comparision of services.

In [34]:
```
InternetService_per= df.InternetService.value_counts() / len(df.InternetService) *100
plt.bar(["Fiber optic","DSL","No"],InternetService_per,align='center',width=0.5 ,color=['blue', 'yellow','bla
ck'],alpha=0.8 )
plt.title(" InternetService Percentage Comparision" , size=18)
plt.xlabel("InternetService" , size=15)
plt.xticks([0,1,2],["Fiber optic","DSL","No"],size=12)
plt.ylabel("Percentage",size=12)
InternetService_per
```

Out[34]:
```
Fiber optic    43.958540
DSL            34.374556
No             21.666903
Name: InternetService, dtype: float64
```



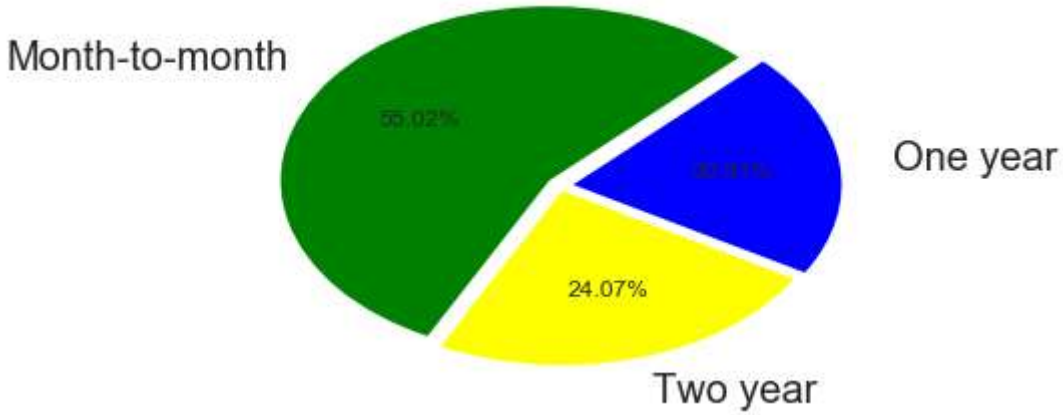
In InternetService there are Two types of service provided first FIBER OPTICS and another is DSL(Digital Subscriber Line), in these Mostly use of FIBER OPTICS service at 44% or DSL service at 34.37% and remaining has 21.67% customer has not used to InternetServices.

Which Contract Service mostly take by customer ? percentage wise comparision of Contract.

In [35]:
```
df['Contract'].value_counts()
```

Out[35]:
```
Month-to-month    3875
Two year          1695
One year          1473
Name: Contract, dtype: int64
```

In [51]:
```
patches, texts, autotexts=plt.pie(df['Contract'].value_counts(), labels=["Month-to-month","Two year","One yea
r"],shadow=False, autopct= '%1.2f%%',explode=(0.05,0.03,0.05),labeldistance=1.2,startangle=45 ,colors=["Gree
n","Yellow","Blue"]);
texts[0].set_fontsize(20)
texts[1].set_fontsize(20)
texts[2].set_fontsize(20)
plt.show()
```

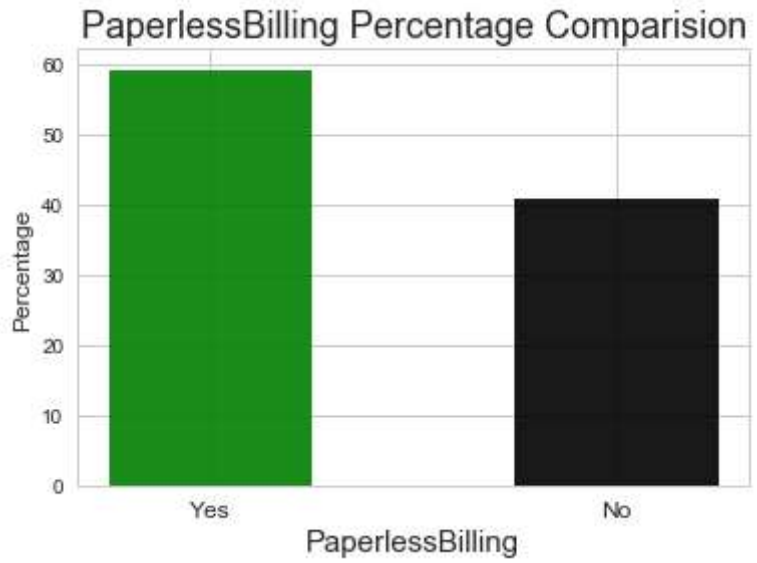


In Contract Service , Month-to-Month Services more prefferd as compare to One year or Two year plan.Month-to-Month service more than 55% customer has use.

How many percentage of customer use Paperless Billing Service or Not ?

In [74]:
```
PaperlessBilling_per= df.PaperlessBilling.value_counts() / len(df.PaperlessBilling) *100
plt.bar(["No","Yes"],PaperlessBilling_per,align='center', width=0.5 ,color=['green','black' ], alpha=0.9)
plt.title("PaperlessBilling Percentage Comparision" , size=18)
plt.xlabel("PaperlessBilling" , size=15)
plt.xticks([0,1],["Yes","No"],size=12)
plt.ylabel("Percentage",size=12)
PaperlessBilling_per
```

Out[74]:
```
Yes    59.221922
No     40.778078
Name: PaperlessBilling, dtype: float64
```



PaperlessBilling Service has 60 % Customer has to be prefferd.

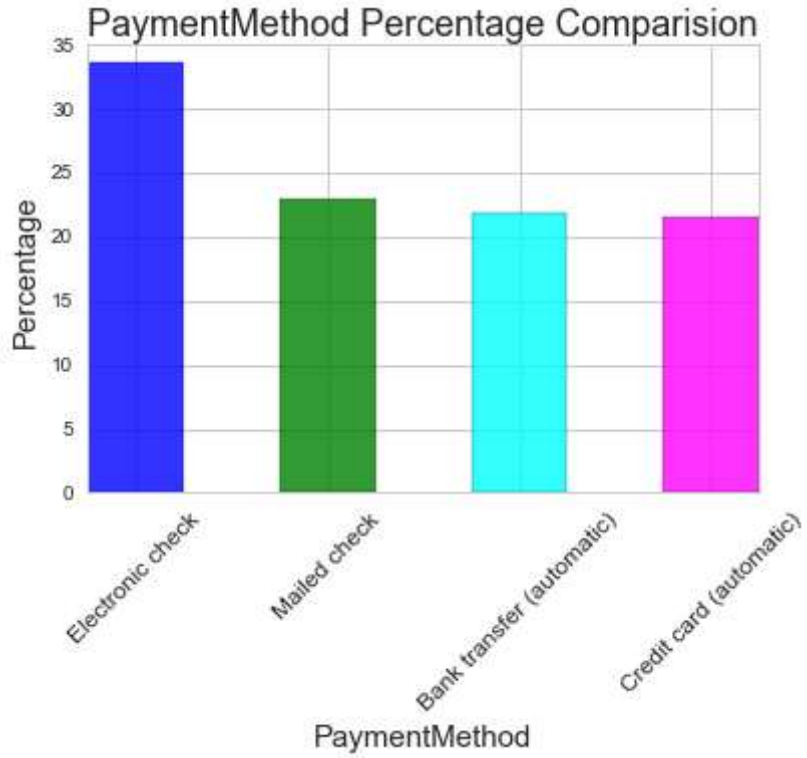
Which Payement Methods mostly take by customer ? percentage wise comparision of Payment Methods.

In [75]:
```
df['PaymentMethod'].value_counts()
```

Out[75]:
```
Electronic check             2365
Mailed check                 1612
Bank transfer (automatic)    1544
Credit card (automatic)      1522
Name: PaymentMethod, dtype: int64
```

In [121]:
```
PaymentMethod_per= df.PaymentMethod.value_counts() / len(df.PaymentMethod) *100
plt.bar(["Electronic check","Mailed check","Bank transfer (automatic)","Credit card (automatic)"],PaymentMeth
od_per,align='center', width=0.5 ,color=['blue', 'green','cyan','magenta'], alpha=0.8)
plt.title("PaymentMethod Percentage Comparision" , size=18)
plt.xlabel("PaymentMethod" , size=15)
plt.xticks([0,1,2,3],["Electronic check","Mailed check","Bank transfer (automatic)","Credit card (automati
c)"],size=12, rotation=45)
plt.ylabel("Percentage",size=15)
PaymentMethod_per
```

Out[121]:
```
Electronic check             33.579441
Mailed check                 22.887974
Bank transfer (automatic)    21.922476
Credit card (automatic)      21.610109
Name: PaymentMethod, dtype: float64
```



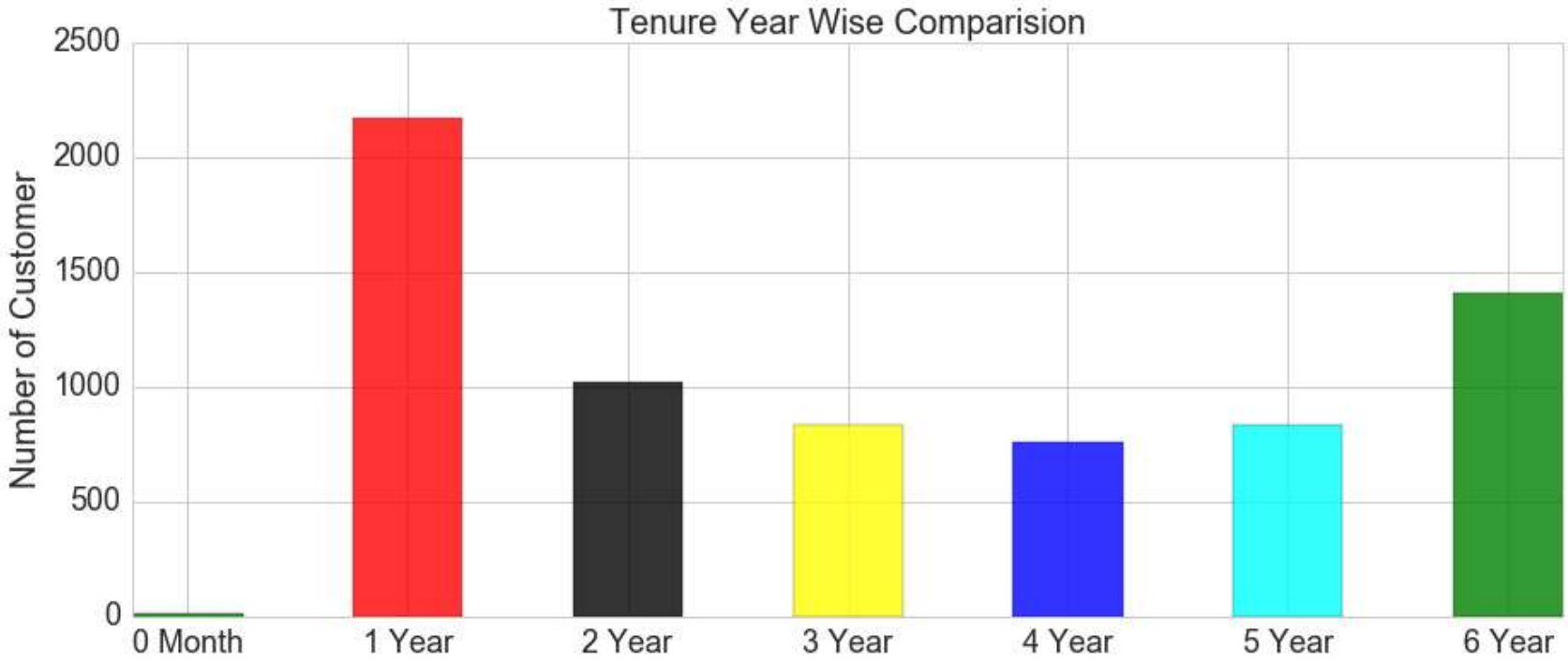
In Payment Methods options Electronic Check service has more prefferd as compare to other options.

How many customer are continue with services at year wise show ?

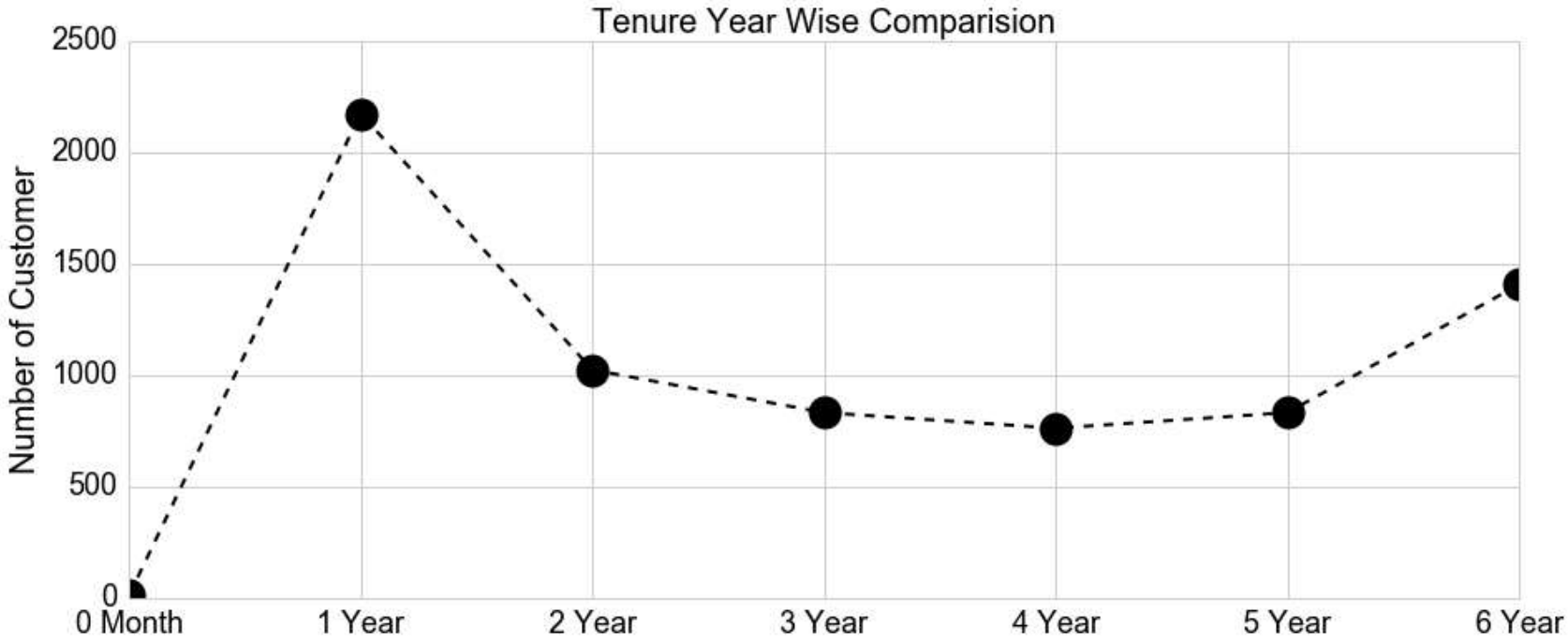
In [57]:
```
d1=df.tenure_year.value_counts().to_frame()
TY=d1.sort_index()
```

In [126]:
```
fig, ax = plt.subplots()
fig.set_size_inches(15,6)
plt.bar(["0","1","2","3","4","5","6"],TY.tenure_year,align='center', width=0.5 ,color=["green","Red","Blac
k","Yellow","Blue","Cyan","green"] ,alpha=0.8)
plt.title("Tenure Year Wise Comparision" , size=20)
plt.xticks([0,1,2,3,4,5,6],["0 Month","1 Year","2 Year","3 Year","4 Year","5 Year","6 Year"],size=18)
plt.ylabel("Number of Customer",size=20)
plt.yticks(size=18)
```

Out[126]:
```
(array([   0.,  500., 1000., 1500., 2000., 2500.]),
 <a list of 6 Text yticklabel objects>)
```



In [127]:
```
fig, ax = plt.subplots()
fig.set_size_inches(15,6)
fig.set_facecolor('w')
plt.plot(TY,"go--",linewidth=2, marker='o',markersize=20 ,color="black")
plt.title("Tenure Year Wise Comparision" , size=20,color="black")
plt.xticks([0,1,2,3,4,5,6],["0 Month","1 Year","2 Year","3 Year","4 Year","5 Year","6 Year"],size=18,color="b
lack")
plt.ylabel("Number of Customer",size=20,color="black")
plt.yticks(size=18,color="black");
```



Maximum Customer are services used at tenure at upto 12 Months of overall customer.

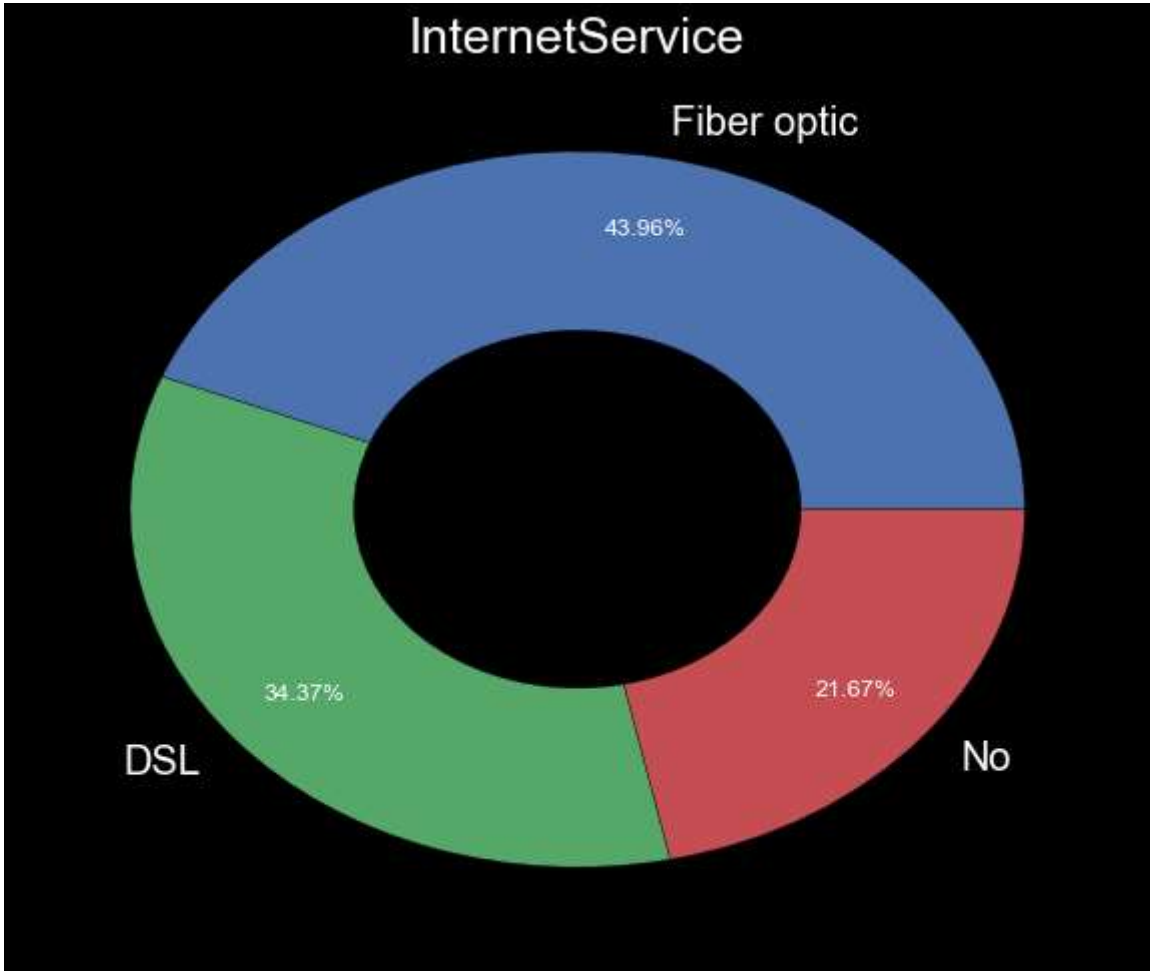
Different Graphs shows of combination of attributes using matplotlib and Seaborn !!!

In [128]:
```
df1= df.groupby(["Churn",'InternetService']).count()
df1
```

Out[128]:

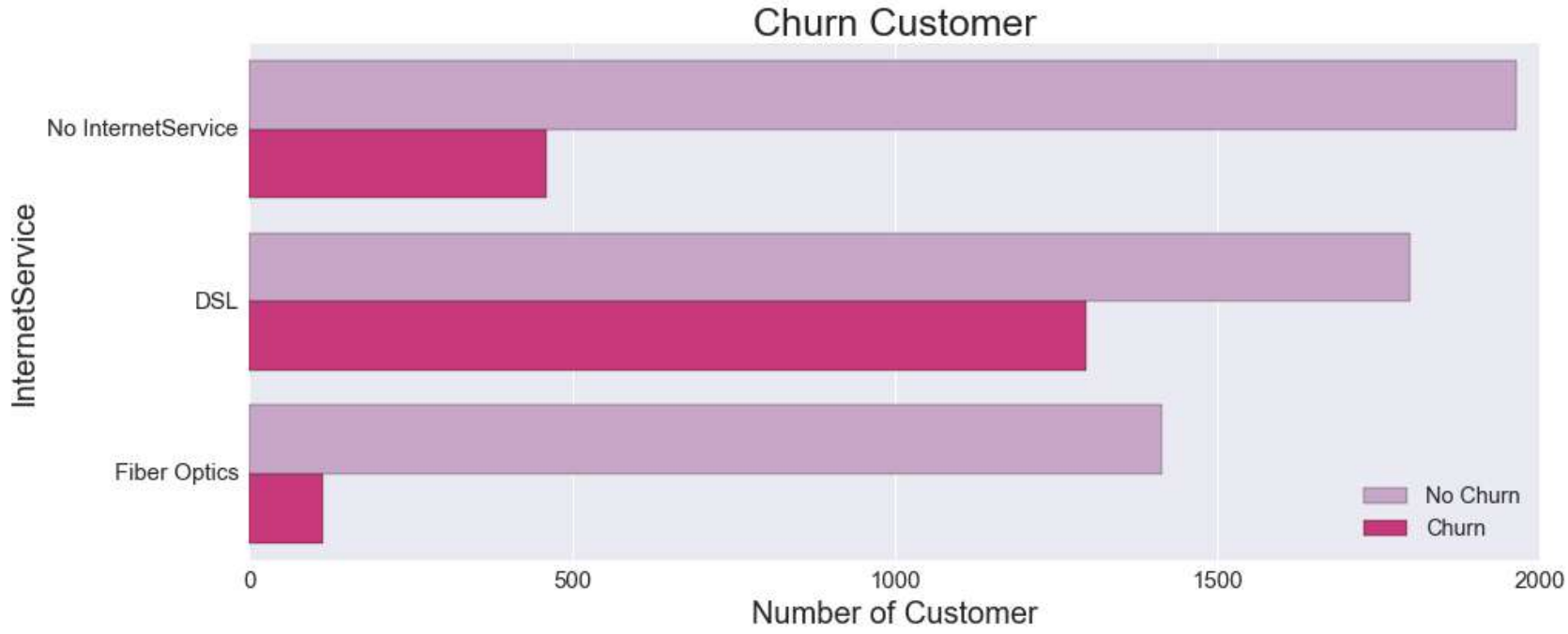
		gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	OnlineSecurity	On
Churn	**InternetService**									
No	**DSL**	1962	1962	1962	1962	1962	1962	1962	1962	196
	Fiber optic	1799	1799	1799	1799	1799	1799	1799	1799	179
	No	1413	1413	1413	1413	1413	1413	1413	1413	141
Yes	**DSL**	459	459	459	459	459	459	459	459	459
	Fiber optic	1297	1297	1297	1297	1297	1297	1297	1297	129
	No	113	113	113	113	113	113	113	113	113

In [130]:
```
df.InternetService.value_counts()/len(df.InternetService) *100
size_of_groups=df['InternetService'].value_counts()
# create a figure and set different background
fig = plt.figure()
fig.set_size_inches(10,8)
fig.patch.set_facecolor('black')

# Change color of text
plt.rcParams['text.color'] = 'white'


 # Create a pieplot
patches, texts, autotexts=plt.pie(size_of_groups,labels=["Fiber optic","DSL","No"], autopct= '%1.2f%%',pctdis
tance=0.8,labeldistance=1.1)
plt.title('InternetService',size=25)
texts[0].set_fontsize(20)
texts[1].set_fontsize(20)
texts[2].set_fontsize(20)
# add a circle at the center
my_circle=plt.Circle( (0,0), 0.5, color='black')
p=plt.gcf()
p.gca().add_artist(my_circle)
plt.show()
```



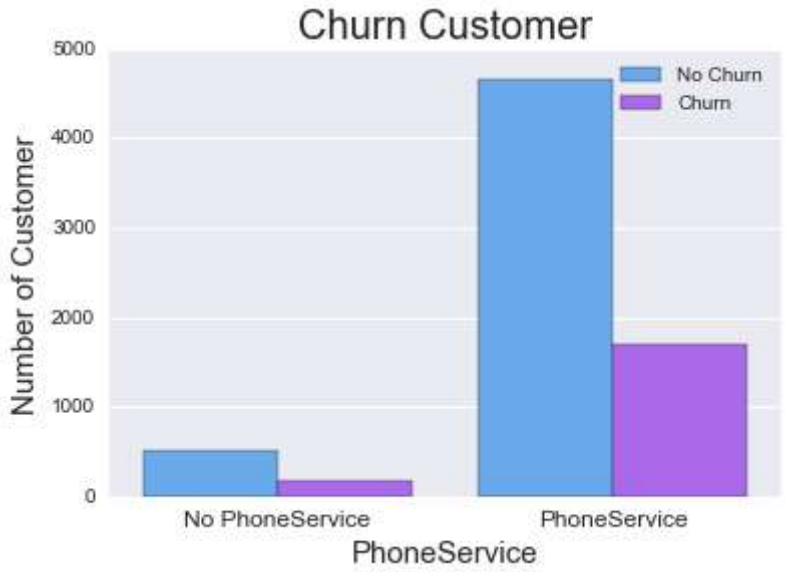
Internet Service has Two types at provides by the provider, in this FIBER OPTICS Service has more preffered as compare to DSL(Digital Subscriber Line)

In [141]:
```
fig, ax = plt.subplots()
fig.set_size_inches(15,6)
sns.countplot(y='InternetService', data=df,hue='Churn',palette="PuRd")
plt.yticks([0,1,2], ['No InternetService','DSL',"Fiber Optics"],size=12)
plt.title("Churn Customer",size=25)
sns.set(style="darkgrid")
plt.legend(labels=['No Churn','Churn'],loc="best", fontsize="large")
plt.xlabel("Number of Customer",size=20)
plt.ylabel("InternetService",size=20)
plt.xticks(size=15)
plt.yticks(size=15);
```



In Internet Services has Maximum Churn Customer at used DSL service, that service has required as more improvemrnt or other option FIBER OPTICS has more advertised or make a user friendly.

In [143]:
```
sns.countplot(x='PhoneService', data=df,hue='Churn',palette="cool")
plt.xticks([0,1], ['No PhoneService','PhoneService'],size=12)
plt.title("Churn Customer",size=20)
sns.set(style="darkgrid")
plt.legend(labels=['No Churn','Churn'])
plt.xlabel("PhoneService",size=15)
plt.ylabel("Number of Customer",size=15);
```



Phone Service as service is most used as compare to Not use , Churn customer has not more effected by Phone Service.

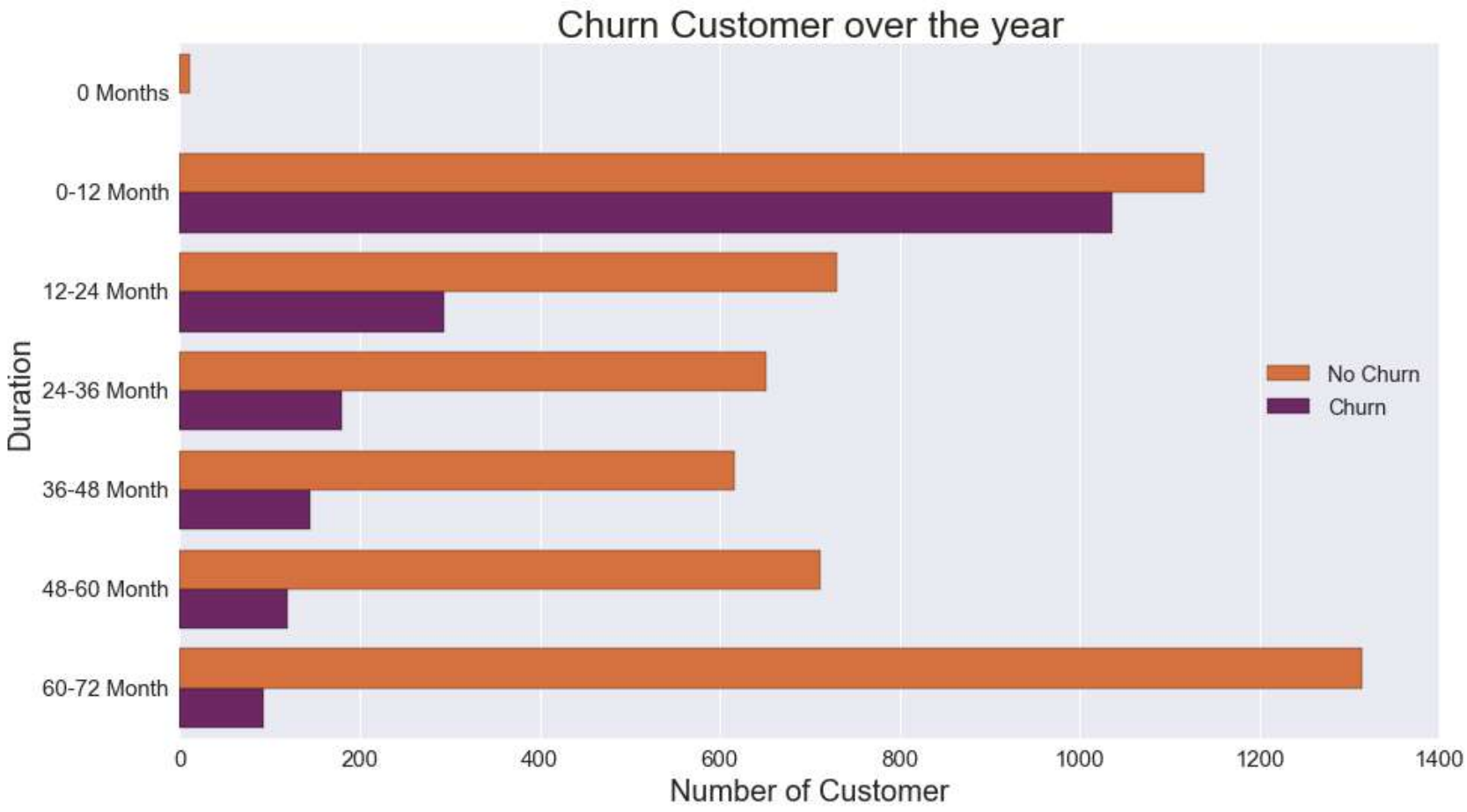
In [144]:
```
df2= df.groupby(["Churn",'tenure_year']).count()
df2
```

Out[144]:

		gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	Onlin
Churn	**tenure_year**									
No	**0.0**	11	11	11	11	11	11	11	11	11
	1.0	1138	1138	1138	1138	1138	1138	1138	1138	1138
	2.0	730	730	730	730	730	730	730	730	730
	3.0	652	652	652	652	652	652	652	652	652
	4.0	617	617	617	617	617	617	617	617	617
	5.0	712	712	712	712	712	712	712	712	712
	6.0	1314	1314	1314	1314	1314	1314	1314	1314	1314
Yes	**1.0**	1037	1037	1037	1037	1037	1037	1037	1037	1037
	2.0	294	294	294	294	294	294	294	294	294
	3.0	180	180	180	180	180	180	180	180	180
	4.0	145	145	145	145	145	145	145	145	145
	5.0	120	120	120	120	120	120	120	120	120
	6.0	93	93	93	93	93	93	93	93	93

In [148]:
```
fig, ax = plt.subplots()
fig.set_size_inches(15, 8.27)
sns.despine()
sns.countplot(y='tenure_year', data=df,hue='Churn',palette='inferno_r')
plt.yticks([0,1,2,3,4,5,6], ['0 Months','0-12 Month','12-24 Month','24-36 Month',\
                             '36-48 Month','48-60 Month','60-72 Month'],size=15)
plt.title("Churn Customer over the year",size=25)
sns.set(style="darkgrid")
plt.legend(labels=['No Churn','Churn'],loc="best",fontsize="large")
plt.xlabel("Number of Customer",size=20)
plt.ylabel("Duration",size=20)
plt.xticks(size=15)
```

```
Out[148]: (array([   0.,  200.,  400.,  600.,  800., 1000., 1200., 1400.]),
           <a list of 8 Text xticklabel objects>)
```



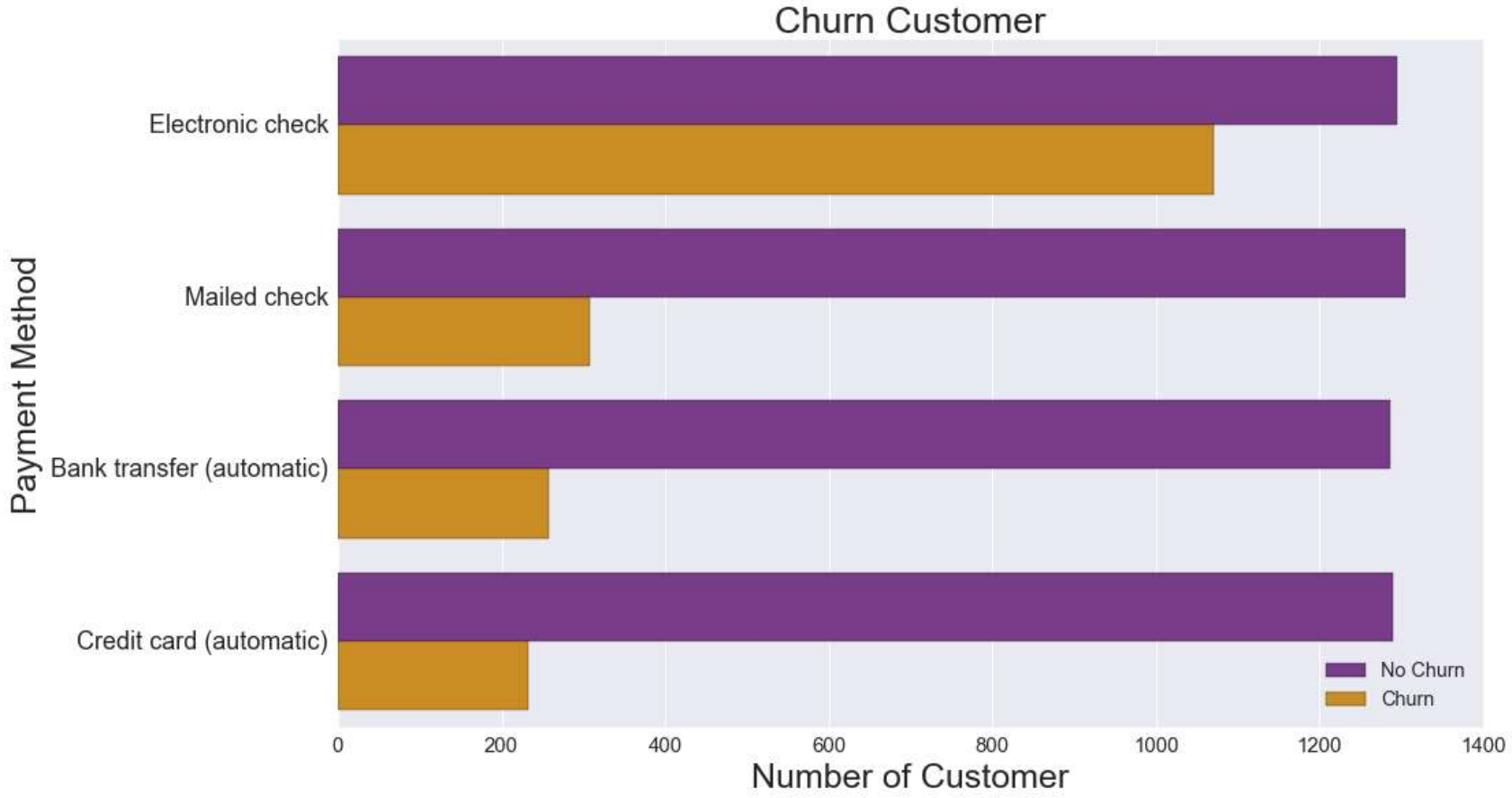
In Churn Customer has Maximum in One year Tenure Range , these range customer has provide more offers and Stay tunned with the services and take time to used the services.

In [149]:
```
df3= df.groupby(["PaymentMethod","Churn"]).count()
df3
```

Out[149]:

		gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	O
PaymentMethod	**Churn**									
Bank transfer (automatic)	**No**	1286	1286	1286	1286	1286	1286	1286	1286	1
	Yes	258	258	258	258	258	258	258	258	2
Credit card (automatic)	**No**	1290	1290	1290	1290	1290	1290	1290	1290	1
	Yes	232	232	232	232	232	232	232	232	2
Electronic check	**No**	1294	1294	1294	1294	1294	1294	1294	1294	1
	Yes	1071	1071	1071	1071	1071	1071	1071	1071	1
Mailed check	**No**	1304	1304	1304	1304	1304	1304	1304	1304	1
	Yes	308	308	308	308	308	308	308	308	3

In [156]:
```
fig, ax = plt.subplots()
fig.set_size_inches(15, 9)
sns.despine()
sns.countplot(y='PaymentMethod', data=df,hue='Churn',palette="CMRmap")
plt.yticks([0,1,2,3], ["Electronic check","Mailed check","Bank transfer (automatic)",\
                       "Credit card (automatic)"],size=18)
plt.title("Churn Customer",size=27)
sns.set(style="darkgrid")
plt.legend(labels=['No Churn','Churn'],loc="lower right", fontsize="large")
plt.xlabel("Number of Customer",size=25)
plt.ylabel("Payment Method",size=25)
plt.xticks(size=15);
```



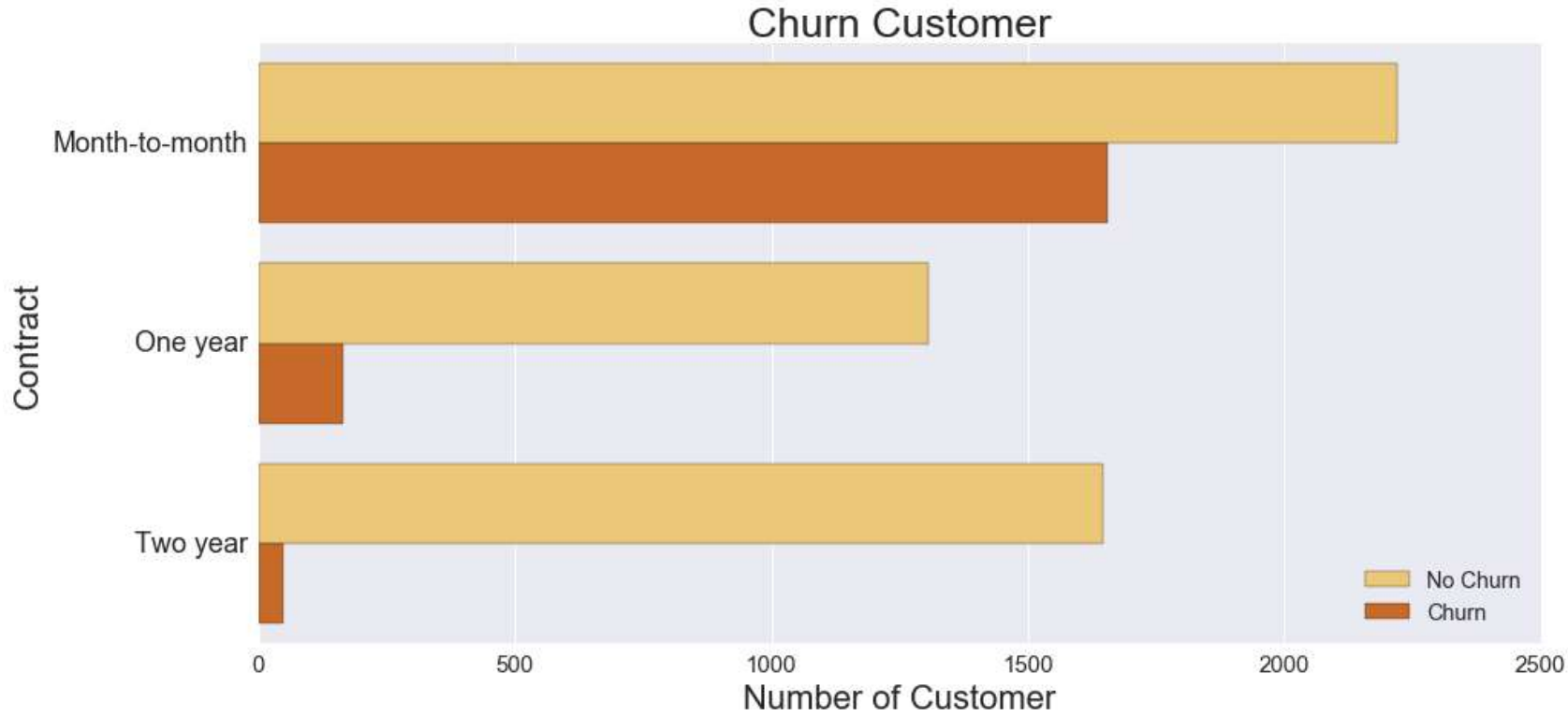
The Payment Methods Options has Electronic check payment method has maximum Churn Customers , this payment method option have not offered in future. Other options as Bank transfer (automatic) and Credit card (automatic) is more easy payment modes can preffered.

In [157]:
```
df4= df.groupby(["Contract","Churn"]).count()
df4
```

Out[157]:

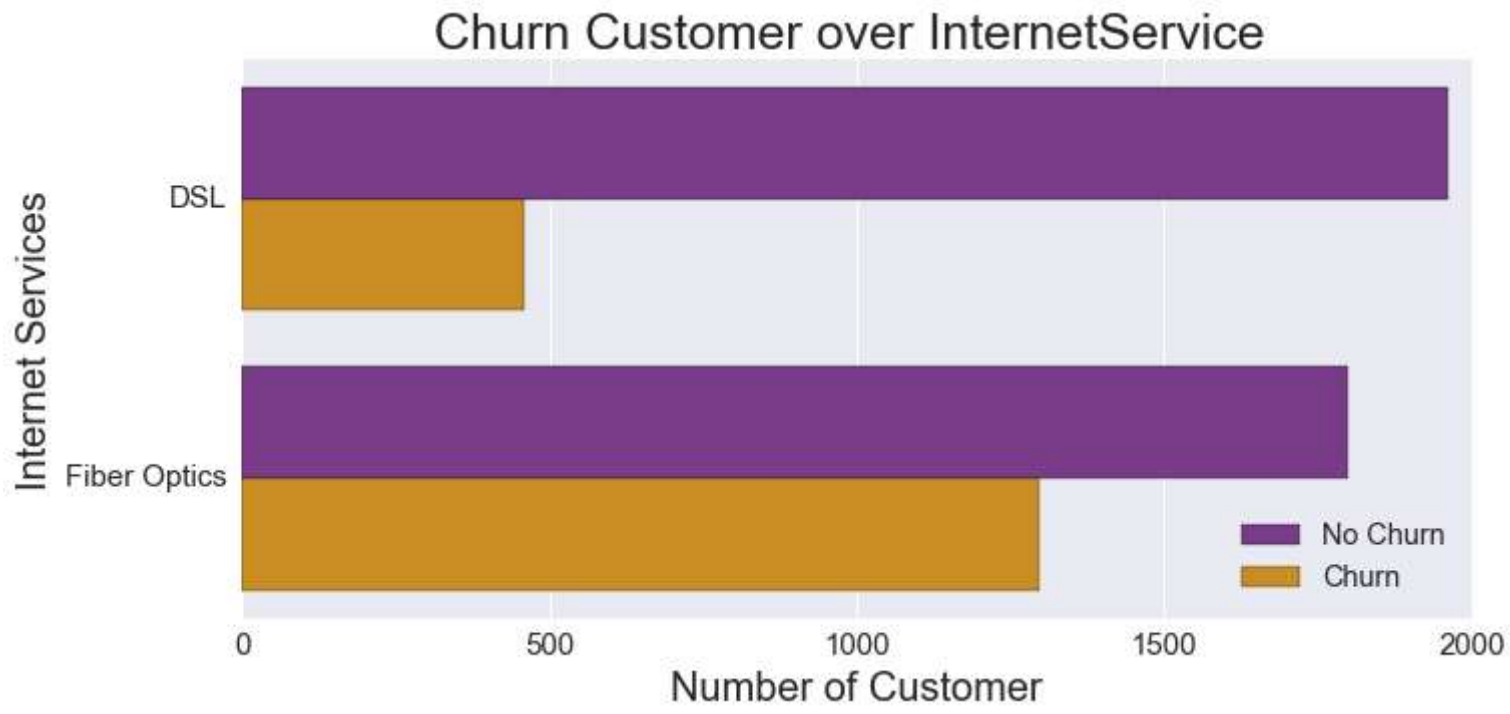
		gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSe
Contract	**Churn**									
Month-to-month	**No**	2220	2220	2220	2220	2220	2220	2220	2220	2220
	Yes	1655	1655	1655	1655	1655	1655	1655	1655	1655
One year	**No**	1307	1307	1307	1307	1307	1307	1307	1307	1307
	Yes	166	166	166	166	166	166	166	166	166
Two year	**No**	1647	1647	1647	1647	1647	1647	1647	1647	1647
	Yes	48	48	48	48	48	48	48	48	48

In [159]:
```
fig, ax = plt.subplots()
fig.set_size_inches(15, 7)
sns.despine()
sns.countplot(y='Contract', data=df,hue='Churn',palette="YlOrBr")
plt.yticks([0,1,2], ["Month-to-month","One year","Two year"],size=18)
plt.title("Churn Customer",size=27)
sns.set(style="darkgrid")
plt.legend(labels=['No Churn','Churn'], loc="best", fontsize="large")
plt.xlabel("Number of Customer",size=22)
plt.ylabel("Contract",size=22)
plt.xticks(size=15);
```



In Contract Service , Month-to-Month Services more prefferd as compare to One year or Two year plan , But Maximum Churn Customer has also in Month-to-Month Service, So the new customer has added or those customer who has tenure range is 0-12 Months these customer has provide One Year or Two year plan has to suggested added more attractive offers ,than customer choose One or Two Year plan , than Churn Rate is decreases.

In [162]:
```
fig, ax = plt.subplots()
fig.set_size_inches(11, 5)
sns.countplot(y='InternetService',hue="Churn", data=df[df.InternetService != 'No' ],palette="CMRmap")
plt.yticks([0,1],['DSL',"Fiber Optics"],size=15)
plt.title("Churn Customer over InternetService ",size=25)
sns.set(style="darkgrid")
plt.legend(labels=['No Churn','Churn'], loc="best", fontsize="large")
plt.xlabel("Number of Customer",size=20)
plt.ylabel("Internet Services",size=20)
plt.xticks(size=15);
```

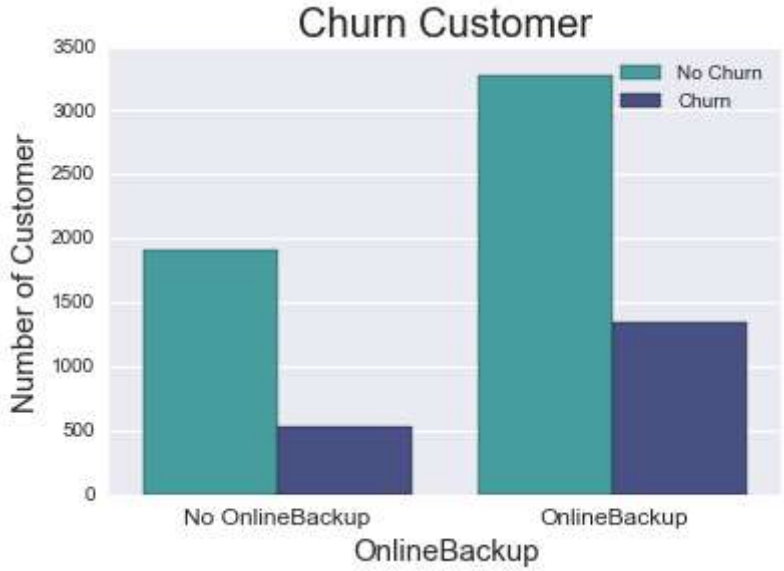


In Internet Services has Maximum Churn Customer at used DSL service, that service has required as more improvemrnt or other option FIBER OPTICS has more advertised or make a user friendly.

In [164]:
```
sns.countplot(x='OnlineSecurity', data=df,hue='Churn',palette="nipy_spectral")
plt.xticks([0,1], ['No OnlineSecurity','OnlineSecurity'],size=12)
plt.title("Churn Customer",size=20)
sns.set(style="whitegrid")
plt.legend(labels=['No Churn','Churn'])
plt.xlabel('OnlineSecurity',size=15)
plt.ylabel("Number of Customer",size=15);
```

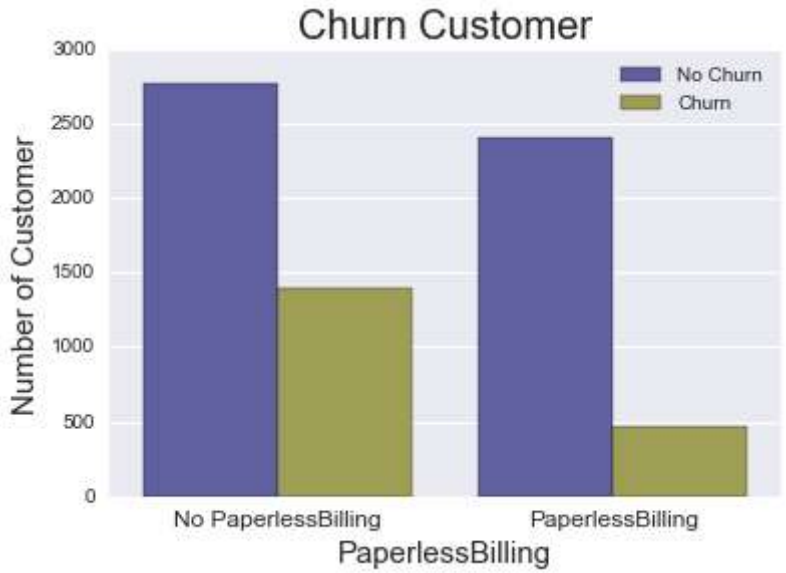
Online Security services has those who service are used has minimum number of Churned customer , Online Security Service has more improve or added facility to other customers than make more profit or decrease Churn Rate.

In [166]:
```
sns.countplot(x='OnlineBackup', data=df,hue='Churn',palette="mako_r")
plt.xticks([0,1], ['No OnlineBackup','OnlineBackup'],size=12)
plt.title("Churn Customer",size=20)
sns.set(style="darkgrid")
plt.legend(labels=['No Churn','Churn'])
plt.xlabel('OnlineBackup',size=15)
plt.ylabel("Number of Customer",size=15);
```



Online Backup services has more number are customer is used and its Churn rate as more as compare to other services. Then this service has required to more improvement.

In [167]:
```
sns.countplot(x='PaperlessBilling', data=df,hue='Churn',palette="gist_stern")
plt.xticks([0,1], ['No PaperlessBilling','PaperlessBilling'],size=12)
plt.title("Churn Customer",size=20)
sns.set(style="darkgrid")
plt.legend(labels=['No Churn','Churn'])
plt.xlabel('PaperlessBilling',size=15)
plt.ylabel("Number of Customer",size=15);
```



Paperless Billing service has very important service ,this service has very less Churn Rate as compare to those who do not use the serices .Then this service has make more user friendly or easy.

Preprocesseing: 1) Feature Selection

In [168]:
```
df.head(1)
```

Out[168]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineB
0	Female	0	Yes	No	1	No	No	DSL	No	Yes

1 rows × 22 columns

In [169]:
```
df.columns
```

Out[169]:
```
Index(['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure',
       'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
       'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
       'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod',
       'MonthlyCharges', 'TotalCharges', 'Churn', 'tenure_yr', 'tenure_year'],
      dtype='object')
```

In [170]:
```
data1=df[['Churn','InternetService','tenure','Partner','Dependents','PhoneService',\
          'MultipleLines','OnlineSecurity','OnlineBackup', 'DeviceProtection', \
          'TechSupport', 'StreamingTV','StreamingMovies', 'Contract', 'PaperlessBilling',\
          'PaymentMethod','MonthlyCharges','tenure_year']]
```

In [171]:
```
data1.head(1)
```

Out[171]:

	Churn	InternetService	tenure	Partner	Dependents	PhoneService	MultipleLines	OnlineSecurity	OnlineBackup	DeviceP
0	No	DSL	1	Yes	No	No	No	No	Yes	No

In [172]:
```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 18 columns):
Churn               7043 non-null object
InternetService     7043 non-null object
tenure              7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
PhoneService        7043 non-null object
MultipleLines       7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null object
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
tenure_year         7043 non-null float64
dtypes: float64(2), int64(1), object(15)
memory usage: 990.5+ KB
```

Preprocessing: 2) Data Transformation

In [173]:
```
data1.Churn.replace(['No','Yes'],[0,1],inplace=True)
data1.InternetService.replace(['No','Fiber optic','DSL'],[0,1,2],inplace=True)
data1.Partner.replace(['No','Yes'],[0,1],inplace=True)
data1.Dependents.replace(['No','Yes'],[0,1],inplace=True)
data1.PhoneService.replace(['No','Yes'],[0,1],inplace=True)
data1.MultipleLines.replace(['No','Yes'],[0,1],inplace=True)
data1.OnlineSecurity.replace(['No','Yes'],[0,1],inplace=True)
data1.OnlineBackup.replace(['No','Yes'],[0,1],inplace=True)
data1.DeviceProtection.replace(['No','Yes'],[0,1],inplace=True)
data1.TechSupport.replace(['No','Yes'],[0,1],inplace=True)
data1.StreamingTV.replace(['No','Yes'],[0,1],inplace=True)
data1.StreamingMovies.replace(['No','Yes'],[0,1],inplace=True)
data1.Contract.replace(['Month-to-month','One year','Two year'],[0,1,2],inplace=True)
data1.PaperlessBilling.replace(['No','Yes'],[0,1],inplace=True)
data1.PaymentMethod.replace(['Electronic check','Mailed check','Bank transfer (automatic)','Credit card (auto
matic)'],[1,2,3,4],inplace=True)
```

```
C:\Users\Sachin\Anaconda3\lib\site-packages\pandas\core\generic.py:5886: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view
-versus-copy
  self._update_inplace(new_data)
```

In [174]:
```
data1.head()
```

Out[174]:

	Churn	InternetService	tenure	Partner	Dependents	PhoneService	MultipleLines	OnlineSecurity	OnlineBackup	DeviceP
0	0	2	1	1	0	0	0	0	1	0
1	0	2	34	0	0	1	0	1	0	1
2	1	2	2	0	0	1	0	1	1	0
3	0	2	45	0	0	0	0	1	0	1
4	1	1	2	0	0	1	0	0	0	0

Apply Modeling Technique

Define Dependent Attributes (Y) or Independent Attributes(X) in Feature Selecting Dataset.

In [175]:
```
X=data1[['tenure','InternetService','PhoneService','MultipleLines','Contract', 'PaperlessBilling','MonthlyCha
rges']]
Y=data1['Churn']
```

SPLITTING DATA INTO TRAIN & TEST

In [176]:
```
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.25)# 75% training and 25% test.
```

```
C:\Users\Sachin\Anaconda3\lib\site-packages\sklearn\cross_validation.py:41: DeprecationWarning: This module w
as deprecated in version 0.18 in favor of the model_selection module into which all the refactored classes an
d functions are moved. Also note that the interface of the new CV iterators are different from that of this m
odule. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
```

In [177]:
```
X_train.shape
```

Out[177]: (5282, 7)

In [178]:
```
X_test.shape
```

Out[178]: (1761, 7)

Apply Logistic Regression Model

In [179]:
```
# import the class
from sklearn.linear_model import LogisticRegression

# instantiate the model (using the default parameters)
logreg = LogisticRegression()

# fit the model with data
logreg.fit(X_train,y_train)
```

Out[179]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

Pridiction Test

In [180]:
```
y_pred=logreg.predict(X_test)        # Prediction Function
```

Check Accuracy

In [182]:
```
from sklearn import metrics
print(metrics.accuracy_score(y_test, y_pred))
```

```
0.7904599659284497
```

Apply Logistic Regression Model Technique has accuracy is apporx 80 %.

Conclusion

Features such as tenure, Contract, PaperlessBilling, MonthlyCharges and InternetService appear to play a role in customer churn.

There does not seem to be a relationship between gender and churn.

Customers in a month-to-month contract, with PaperlessBilling and are within 12 months tenure, are more likely to churn; On the other hand, customers with one or two year contract, with longer than 12 months tenure, that are not using PaperlessBilling, are less likely to churn.