In [1]:

```
import pandas as pd
```

In [2]:

```
df = pd.read_csv('Screen Time Data.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	index	Date	Week Day	Total Screen Time	Social Networking	Reading and Reference	Other	Productivity	Health and Fitness	Ente
0	0	04/17/19	Wednesday	187	89	17	41	22	0	
1	1	04/18/19	Thursday	123	78	17	8	9	0	
2	2	04/19/19	Friday	112	52	40	8	4	0	
3	3	04/20/19	Saturday	101	69	9	38	2	0	
4	4	04/21/19	Sunday	56	35	2	43	3	0	

In [4]:

```
df.tail()
```

Out[4]:

	index	Date	Week Day	Total Screen Time	Social Networking	Reading and Reference	Other	Productivity	Health and Fitness	Entert
23	23	05-10-2019	Friday	161	93	13	17	16	1	
24	24	05-11-2019	Saturday	58	49	1	2	2	0	
25	25	05-12-2019	Sunday	52	28	1	1	6	0	
26	26	05/13/19	Monday	61	37	1	0	4	0	
27	27	05/14/19	Tuesday	88	41	2	7	15	0	

In [5]:

```
df = df.drop('index', axis = 1)
```

In [6]:

```
df.shape
```

Out[6]:

```
(28, 11)
```

In [7]:

```
df.columns
```

Out[7]:

```
Index(['Date', 'Week Day', 'Total Screen Time ', 'Social Networking',
       'Reading and Reference', 'Other', 'Productivity', 'Health and Fitness',
       'Entertainment', 'Creativity', 'Yoga'],
      dtype='object')
```

In [8]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28 entries, 0 to 27
Data columns (total 11 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   Date                   28 non-null     object
 1   Week Day               28 non-null     object
 2   Total Screen Time      28 non-null     int64 
 3   Social Networking      28 non-null     int64 
 4   Reading and Reference  28 non-null     int64 
 5   Other                  28 non-null     int64 
 6   Productivity           28 non-null     int64 
 7   Health and Fitness     28 non-null     int64 
 8   Entertainment          28 non-null     int64 
 9   Creativity             28 non-null     int64 
 10  Yoga                   28 non-null     int64 
dtypes: int64(9), object(2)
memory usage: 2.5+ KB
```

In [9]:

```
df.describe()
```

Out[9]:

	Total Screen Time	Social Networking	Reading and Reference	Other	Productivity	Health and Fitness	Entertainment	Cre
count	28.000000	28.000000	28.000000	28.000000	28.000000	28.000000	28.000000	28.0
mean	113.250000	60.714286	8.714286	14.821429	9.071429	1.285714	1.821429	0.2
std	43.562322	22.853571	9.340700	13.754605	6.097532	3.952094	6.188721	0.5
min	52.000000	25.000000	0.000000	0.000000	2.000000	0.000000	0.000000	0.0
25%	83.500000	41.750000	2.000000	4.750000	3.000000	0.000000	0.000000	0.0
50%	111.000000	58.000000	5.000000	9.000000	8.000000	0.000000	0.000000	0.0
75%	137.250000	76.500000	13.500000	21.500000	15.000000	0.000000	1.000000	0.0
max	198.000000	109.000000	40.000000	43.000000	22.000000	15.000000	32.000000	2.0

In [10]:

```
df.nunique()
```

Out[10]:

```
Date                     28
Week Day                  7
Total Screen Time        26
Social Networking        27
Reading and Reference    17
Other                    22
Productivity             13
Health and Fitness        4
Entertainment             5
Creativity                3
Yoga                      2
dtype: int64
```

In [11]:

```
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [12]:

```
df['Week Day'].unique()
```

Out[12]:

```
array(['Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday', 'Monday',
       'Tuesday'], dtype=object)
```

In [13]:

```
df['Week Day'].value_counts()
```

Out[13]:

```
Wednesday    4
Thursday     4
Friday       4
Saturday     4
Sunday       4
Monday       4
Tuesday      4
Name: Week Day, dtype: int64
```

In [14]:

```
df['Health and Fitness'].unique()
```

Out[14]:

```
array([ 0,  4, 15,  1], dtype=int64)
```

In [15]:

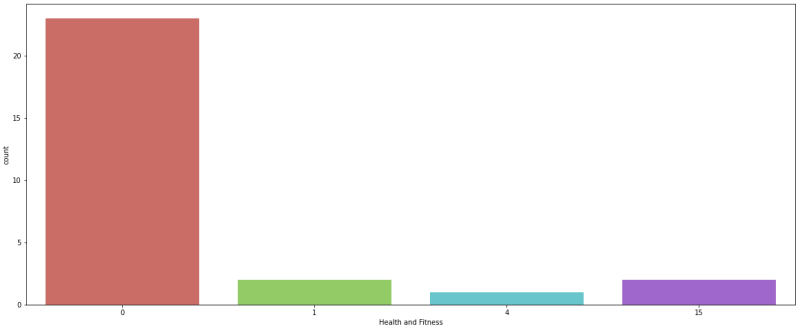
```
df['Health and Fitness'].value_counts()
```

Out[15]:

```
0     23
15     2
1      2
4      1
Name: Health and Fitness, dtype: int64
```

In [16]:

```
plt.figure(figsize=(20,8))
sns.countplot('Health and Fitness', data = df, palette = 'hls')
plt.show()
```



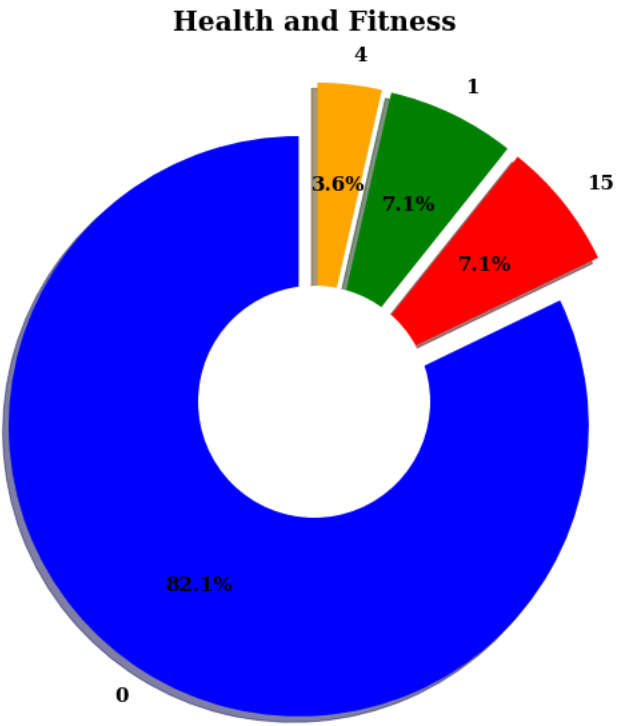
In [17]:

```
label_data = df['Health and Fitness'].value_counts()

explode = (0.1, 0.1, 0.1, 0.1)
plt.figure(figsize=(14, 10))
patches, texts, pcts = plt.pie(label_data,
                               labels = label_data.index,
                               colors = ['blue', 'red', 'green', 'orange'],
                               pctdistance = 0.65,
                               shadow = True,
                               startangle = 90,
                               explode = explode,
                               autopct = '%1.1f%%',
                               textprops={ 'fontsize': 15,
                                           'color': 'black',
                                           'weight': 'bold',
                                           'family': 'serif' })
plt.setp(pcts, color='black')

hfont = {'fontname':'serif', 'weight': 'bold'}
plt.title('Health and Fitness', size=20, **hfont)

centre_circle = plt.Circle((0,0),0.40,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.show()
```



In [18]:

```
df['Entertainment'].unique()
```

Out[18]:

```
array([ 0,  3,  1, 32,  9], dtype=int64)
```

In [19]:

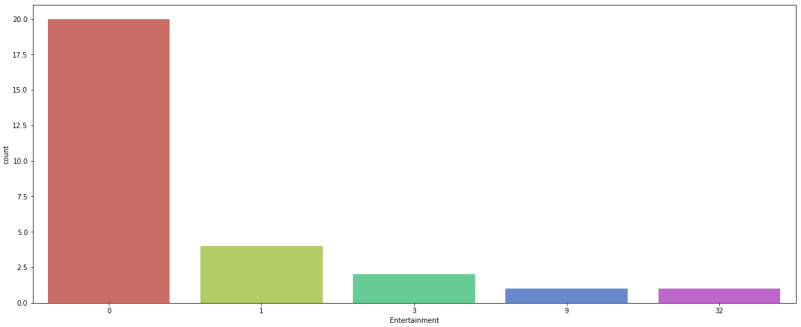
```
df['Entertainment'].value_counts()
```

Out[19]:

```
0     20
1      4
3      2
32     1
9      1
Name: Entertainment, dtype: int64
```

In [20]:

```
plt.figure(figsize=(20,8))
sns.countplot('Entertainment', data = df, palette = 'hls')
plt.show()
```



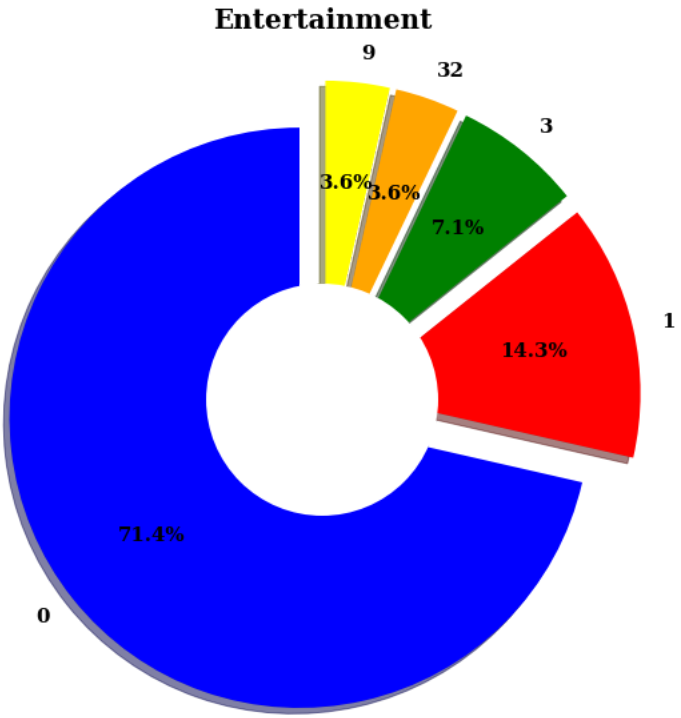
In [21]:

```
label_data = df['Entertainment'].value_counts()

explode = (0.1, 0.1, 0.1, 0.1, 0.1)
plt.figure(figsize=(14, 10))
patches, texts, pcts = plt.pie(label_data,
                               labels = label_data.index,
                               colors = ['blue', 'red', 'green', 'orange',
                                         'yellow'],
                               pctdistance = 0.65,
                               shadow = True,
                               startangle = 90,
                               explode = explode,
                               autopct = '%1.1f%%',
                               textprops={ 'fontsize': 15,
                                           'color': 'black',
                                           'weight': 'bold',
                                           'family': 'serif' })
plt.setp(pcts, color='black')

hfont = {'fontname':'serif', 'weight': 'bold'}
plt.title('Entertainment', size=20, **hfont)

centre_circle = plt.Circle((0,0),0.40,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.show()
```



In [22]:

```
df['Creativity'].unique()
```

Out[22]:

```
array([0, 1, 2], dtype=int64)
```

In [23]:

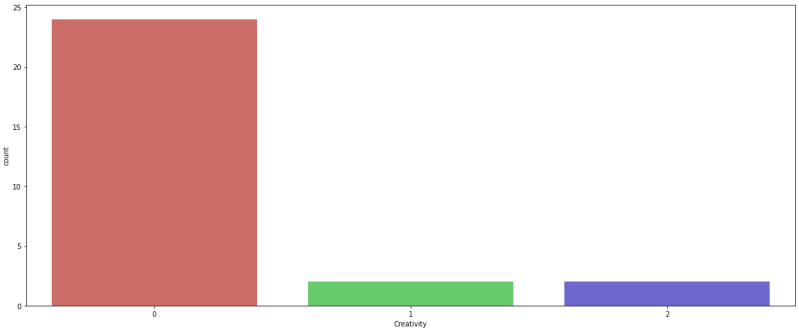
```
df['Creativity'].value_counts()
```

Out[23]:

```
0    24
1     2
2     2
Name: Creativity, dtype: int64
```

In [24]:

```
plt.figure(figsize=(20,8))
sns.countplot('Creativity', data = df, palette = 'hls')
plt.show()
```



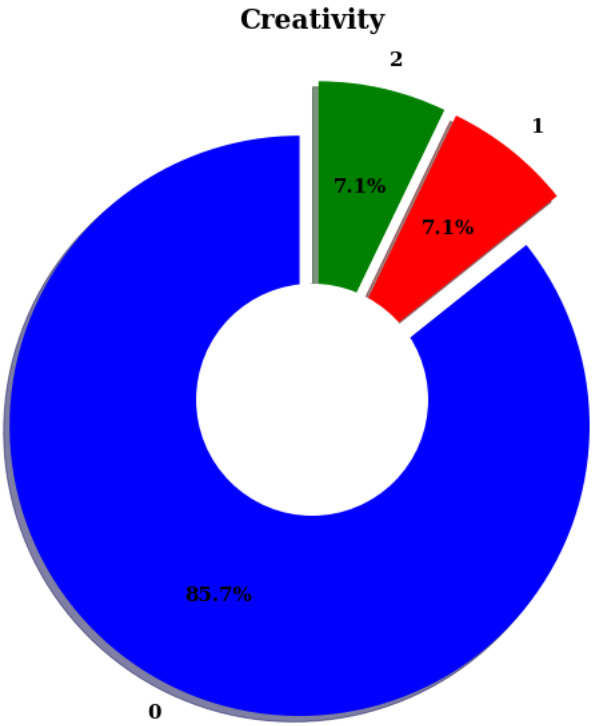
In [25]:

```
label_data = df['Creativity'].value_counts()

explode = (0.1, 0.1, 0.1)
plt.figure(figsize=(14, 10))
patches, texts, pcts = plt.pie(label_data,
                               labels = label_data.index,
                               colors = ['blue', 'red', 'green'],
                               pctdistance = 0.65,
                               shadow = True,
                               startangle = 90,
                               explode = explode,
                               autopct = '%1.1f%%',
                               textprops={ 'fontsize': 15,
                                           'color': 'black',
                                           'weight': 'bold',
                                           'family': 'serif' })
plt.setp(pcts, color='black')

hfont = {'fontname':'serif', 'weight': 'bold'}
plt.title('Creativity', size=20, **hfont)

centre_circle = plt.Circle((0,0),0.40,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.show()
```



In [26]:

```
df['Yoga'].unique()
```

Out[26]:

```
array([0, 1], dtype=int64)
```

In [27]:

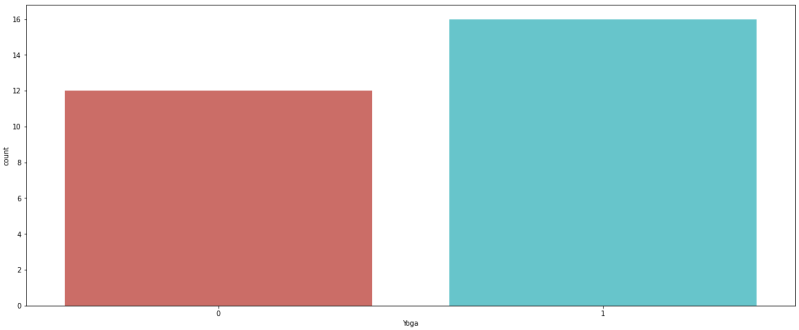
```
df['Yoga'].value_counts()
```

Out[27]:

```
1    16
0    12
Name: Yoga, dtype: int64
```

In [28]:

```
plt.figure(figsize=(20,8))
sns.countplot('Yoga', data = df, palette = 'hls')
plt.show()
```



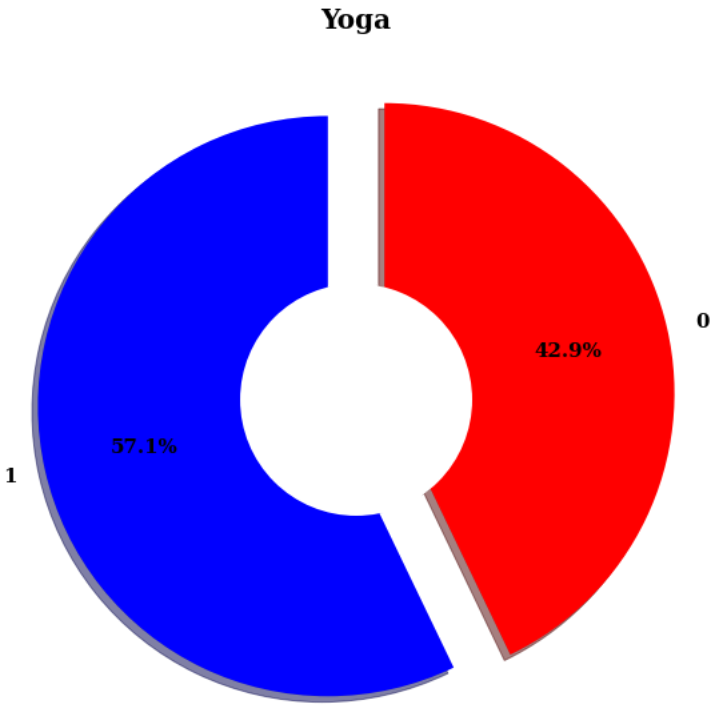
In [29]:

```
label_data = df['Yoga'].value_counts()

explode = (0.1, 0.1)
plt.figure(figsize=(14, 10))
patches, texts, pcts = plt.pie(label_data,
                               labels = label_data.index,
                               colors = ['blue', 'red'],
                               pctdistance = 0.65,
                               shadow = True,
                               startangle = 90,
                               explode = explode,
                               autopct = '%1.1f%%',
                               textprops={ 'fontsize': 15,
                                           'color': 'black',
                                           'weight': 'bold',
                                           'family': 'serif' })
plt.setp(pcts, color='black')

hfont = {'fontname':'serif', 'weight': 'bold'}
plt.title('Yoga', size=20, **hfont)

centre_circle = plt.Circle((0,0),0.40,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.show()
```



In [30]:

```
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
```

In [31]:

```
df['day'] = (df['Date']).dt.day
df['month'] = (df['Date']).dt.month
df['year'] = (df['Date']).dt.year
```

In [32]:

```
df
```

Out[32]:

	Date	Week Day	Total Screen Time	Social Networking	Reading and Reference	Other	Productivity	Health and Fitness	Entertainment
0	2019-04-17	Wednesday	187	89	17	41	22	0	0
1	2019-04-18	Thursday	123	78	17	8	9	0	0
2	2019-04-19	Friday	112	52	40	8	4	0	3
3	2019-04-20	Saturday	101	69	9	38	2	0	3
4	2019-04-21	Sunday	56	35	2	43	3	0	1
5	2019-04-22	Monday	189	68	0	9	3	4	0
6	2019-04-23	Tuesday	158	56	18	41	12	15	0
7	2019-04-24	Wednesday	135	98	3	33	16	0	0
8	2019-04-25	Thursday	52	25	7	3	16	0	0
9	2019-04-26	Friday	198	76	8	29	15	0	32
10	2019-04-27	Saturday	116	75	10	20	5	0	0
11	2019-04-28	Sunday	85	42	22	4	2	0	0
12	2019-04-29	Monday	109	46	8	13	9	15	1
13	2019-04-30	Tuesday	79	40	2	9	12	0	0
14	2019-05-01	Wednesday	127	90	0	10	7	0	0
15	2019-05-02	Thursday	170	60	3	2	11	0	0
16	2019-05-03	Friday	91	64	2	18	5	1	1
17	2019-05-04	Saturday	58	34	4	5	3	0	1
18	2019-05-05	Sunday	133	109	5	1	3	0	0
19	2019-05-06	Monday	144	81	4	5	3	0	0
20	2019-05-07	Tuesday	110	70	5	6	15	0	9
21	2019-05-08	Wednesday	122	53	25	26	15	0	0
22	2019-05-09	Thursday	96	42	15	16	19	0	0
23	2019-05-10	Friday	161	93	13	17	16	1	0

	Date	Week Day	Total Screen Time	Social Networking	Reading and Reference	Other	Productivity	Health and Fitness	Entertainment
24	2019-05-11	Saturday	58	49	1	2	2	0	0
25	2019-05-12	Sunday	52	28	1	1	6	0	0
26	2019-05-13	Monday	61	37	1	0	4	0	0
27	2019-05-14	Tuesday	88	41	2	7	15	0	0

In [33]:

```
df = df.rename(columns = {'Total Screen Time ':'Total_Screen_Time'})
```

In [34]:

```
df
```

Out[34]:

	Date	Week Day	Total_Screen_Time	Social Networking	Reading and Reference	Other	Productivity	Health and Fitness	E
0	2019-04-17	Wednesday	187	89	17	41	22	0	
1	2019-04-18	Thursday	123	78	17	8	9	0	
2	2019-04-19	Friday	112	52	40	8	4	0	
3	2019-04-20	Saturday	101	69	9	38	2	0	
4	2019-04-21	Sunday	56	35	2	43	3	0	
5	2019-04-22	Monday	189	68	0	9	3	4	
6	2019-04-23	Tuesday	158	56	18	41	12	15	
7	2019-04-24	Wednesday	135	98	3	33	16	0	
8	2019-04-25	Thursday	52	25	7	3	16	0	
9	2019-04-26	Friday	198	76	8	29	15	0	
10	2019-04-27	Saturday	116	75	10	20	5	0	
11	2019-04-28	Sunday	85	42	22	4	2	0	
12	2019-04-29	Monday	109	46	8	13	9	15	
13	2019-04-30	Tuesday	79	40	2	9	12	0	
14	2019-05-01	Wednesday	127	90	0	10	7	0	
15	2019-05-02	Thursday	170	60	3	2	11	0	
16	2019-05-03	Friday	91	64	2	18	5	1	
17	2019-05-04	Saturday	58	34	4	5	3	0	
18	2019-05-05	Sunday	133	109	5	1	3	0	
19	2019-05-06	Monday	144	81	4	5	3	0	
20	2019-05-07	Tuesday	110	70	5	6	15	0	
21	2019-05-08	Wednesday	122	53	25	26	15	0	
22	2019-05-09	Thursday	96	42	15	16	19	0	
23	2019-05-10	Friday	161	93	13	17	16	1	

	Date	Week Day	Total_Screen_Time	Social Networking	Reading and Reference	Other	Productivity	Health and Fitness	E
24	2019-05-11	Saturday	58	49	1	2	2	0	
25	2019-05-12	Sunday	52	28	1	1	6	0	
26	2019-05-13	Monday	61	37	1	0	4	0	
27	2019-05-14	Tuesday	88	41	2	7	15	0	

In [35]:

```
df.groupby('month')['Total_Screen_Time'].sum()
```

Qut[35]:

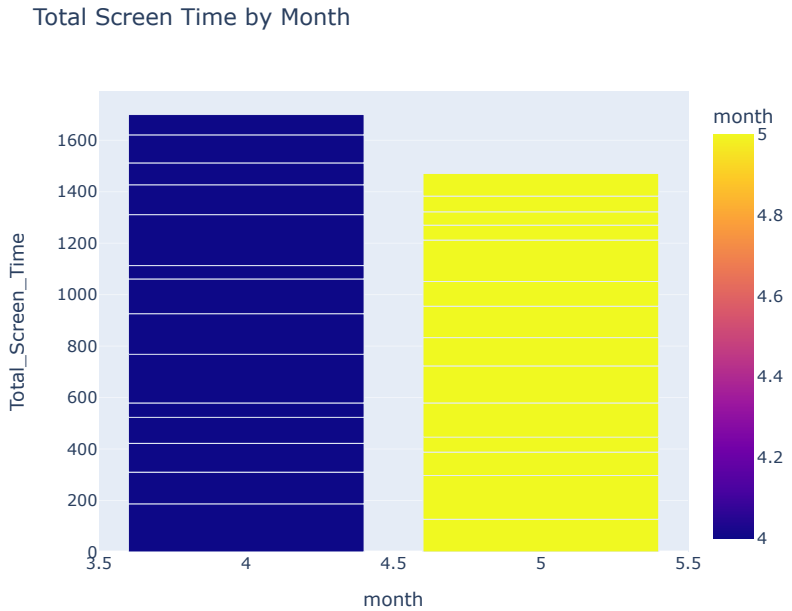
```
month
4    1700
5    1471
Name: Total_Screen_Time, dtype: int64
```

In [36]:

```
import plotly.express as px
```

In [37]:

```
fig = px.bar(df, x="month", y="Total_Screen_Time",
             color="month", title="Total Screen Time by Month")
fig.show()
```



In [38]:

```
df = df.rename(columns = {'Week Day':'Week_Day'})
```

In [39]:

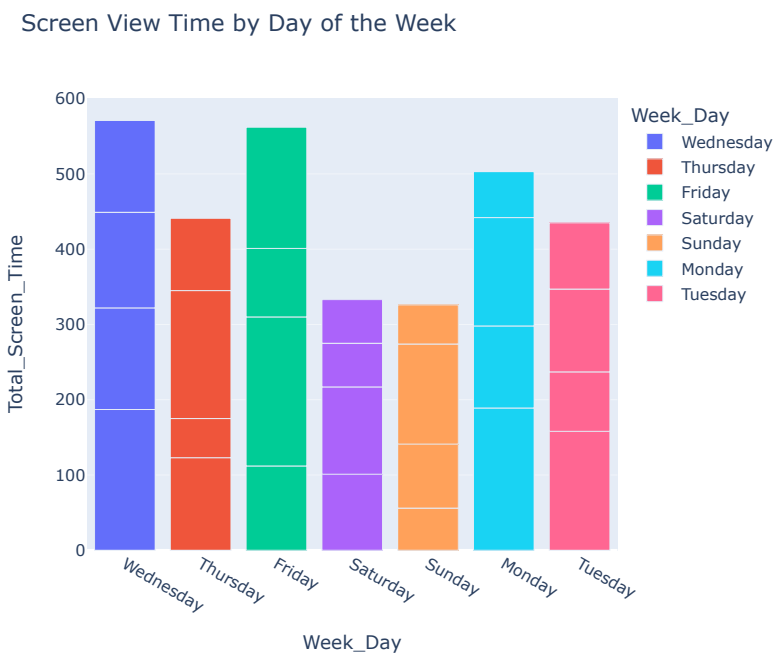
```
df.groupby('Week_Day')['Total_Screen_Time'].sum()
```

Out[39]:

```
Week_Day
Friday       562
Monday       503
Saturday     333
Sunday       326
Thursday     441
Tuesday      435
Wednesday    571
Name: Total_Screen_Time, dtype: int64
```

In [40]:

```
fig = px.bar(df, x="Week_Day", y="Total_Screen_Time", color="Week_Day",
             title="Screen View Time by Day of the Week")
fig.show()
```



In [41]:

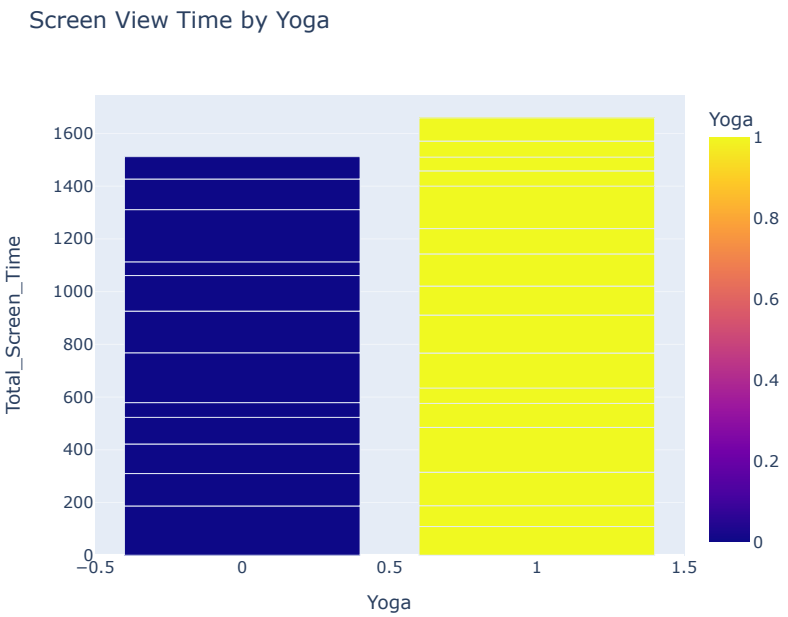
```
df.groupby('Yoga')['Total_Screen_Time'].sum()
```

Out[41]:

```
Yoga
0    1512
1    1659
Name: Total_Screen_Time, dtype: int64
```

In [42]:

```
fig = px.bar(df, x="Yoga", y="Total_Screen_Time", color="Yoga",
             title="Screen View Time by Yoga")
fig.show()
```



In [43]:

```
df.groupby('day')['Total_Screen_Time'].sum()
```

Out[43]:

```
day
1     127
2     170
3      91
4      58
5     133
6     144
7     110
8     122
9      96
10    161
11     58
12     52
13     61
14     88
17    187
18    123
19    112
20    101
21     56
22    189
23    158
24    135
25     52
26    198
27    116
28     85
29    109
30     79
Name: Total_Screen_Time, dtype: int64
```

In [44]:

```
df = df.rename(columns = {'Health and Fitness':'Health_and_Fitness'})
```

In [45]:

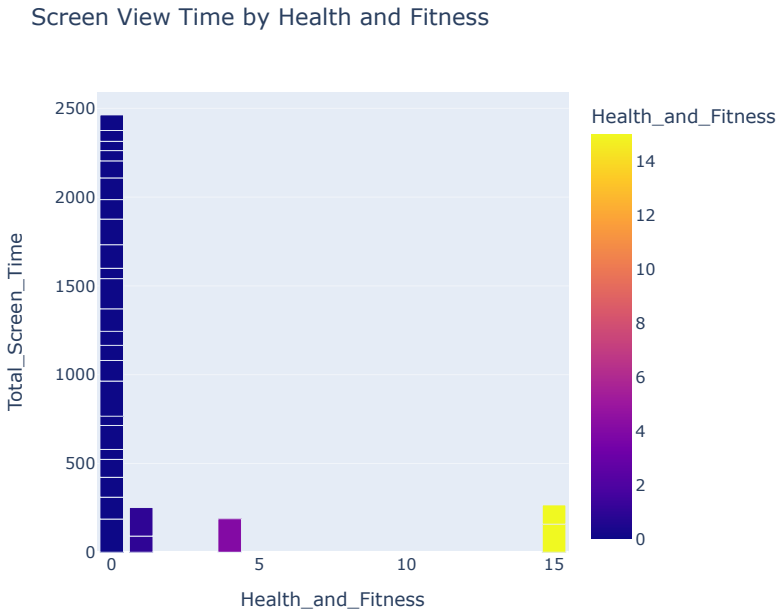
```
df.groupby('Health_and_Fitness')['Total_Screen_Time'].sum()
```

Out[45]:

```
Health_and_Fitness
0     2463
1      252
4      189
15     267
Name: Total_Screen_Time, dtype: int64
```

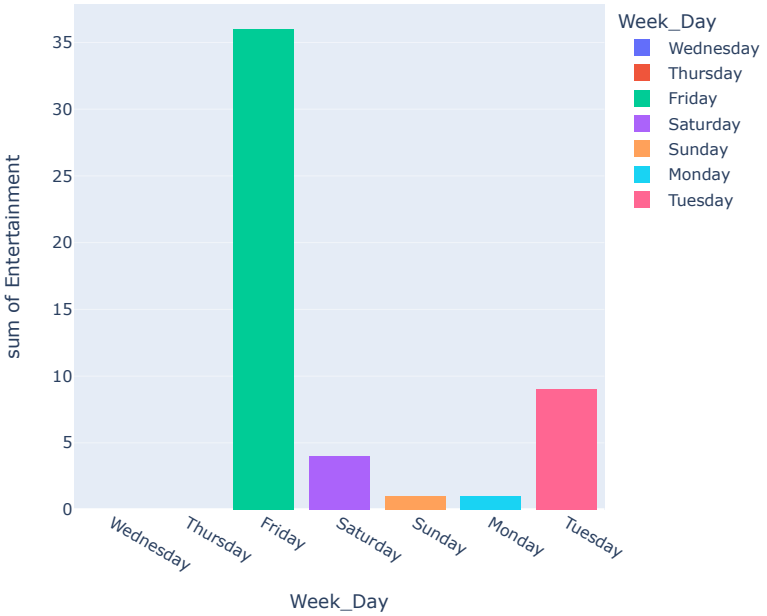
In [46]:

```
fig = px.bar(df, x="Health_and_Fitness", y="Total_Screen_Time", color="Health_and_Fitness",
             title="Screen View Time by Health and Fitness")
fig.show()
```



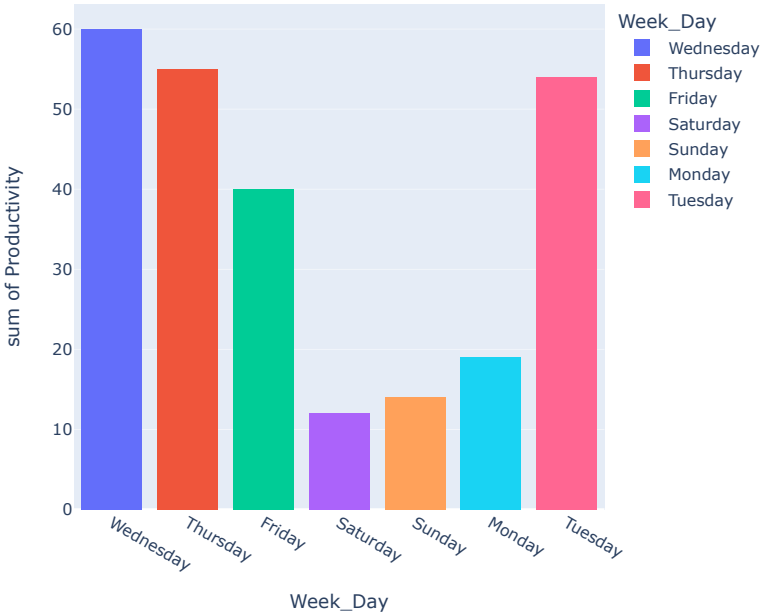
In [47]:

```
fig = px.histogram(df,x="Week_Day",y="Entertainment",color="Week_Day")
fig.show()
```



In [48]:

```
fig = px.histogram(df,x="Week_Day",y="Productivity",color="Week_Day")
fig.show()
```

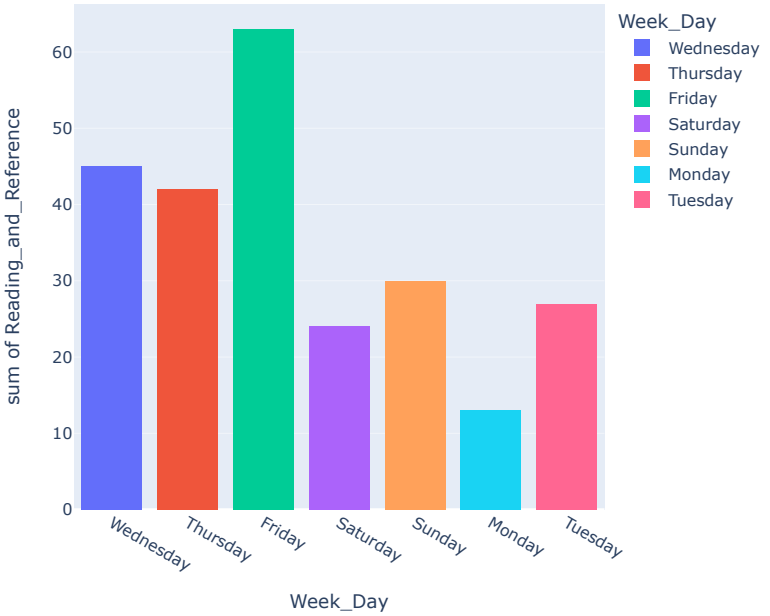


In [49]:

```
df = df.rename(columns = {'Reading and Reference':'Reading_and_Reference'})
```

In [50]:

```
fig = px.histogram(df,x="Week_Day",y="Reading_and_Reference",color="Week_Day")
fig.show()
```

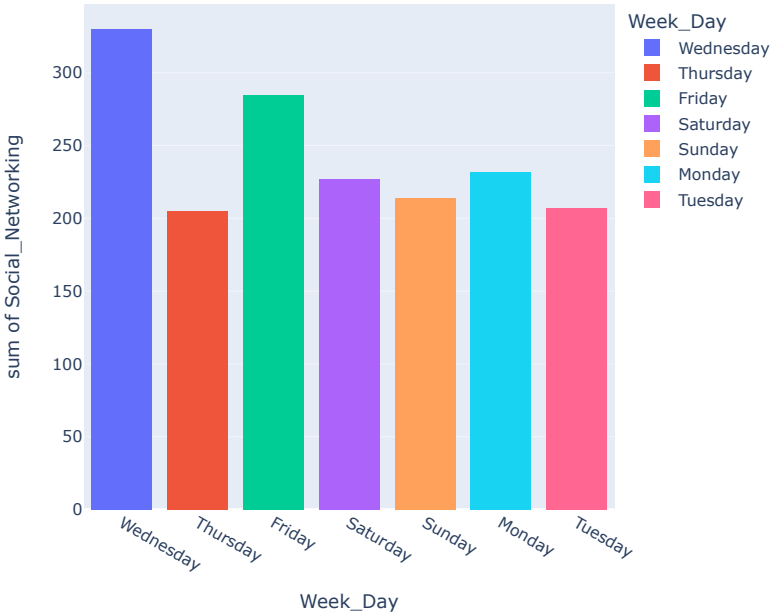


In [51]:

```
df = df.rename(columns = {'Social Networking':'Social_Networking'})
```

In [52]:

```
fig = px.histogram(df,x="Week_Day",y="Social_Networking",color="Week_Day")
fig.show()
```



In [53]:

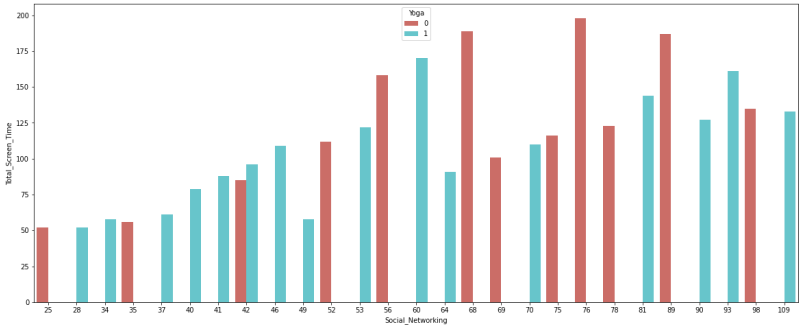
```
df.columns
```

Out[53]:

```
Index(['Date', 'Week_Day', 'Total_Screen_Time', 'Social_Networking',
       'Reading_and_Reference', 'Other', 'Productivity', 'Health_and_Fitness',
       'Entertainment', 'Creativity', 'Yoga', 'day', 'month', 'year'],
      dtype='object')
```

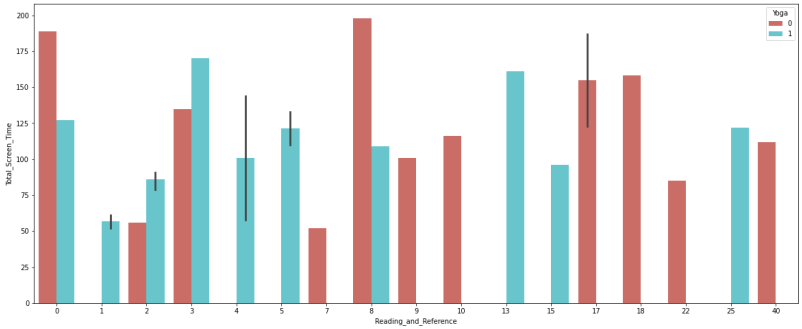
In [54]:

```
plt.figure(figsize=(20,8))
sns.barplot(x = 'Social_Networking', y = 'Total_Screen_Time',
            hue = 'Yoga', data = df, palette = 'hls')
plt.show()
```



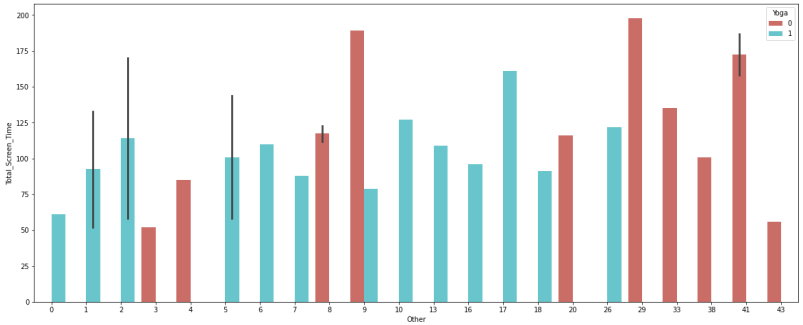
In [55]:

```
plt.figure(figsize=(20,8))
sns.barplot(x = 'Reading_and_Reference', y = 'Total_Screen_Time',
            hue = 'Yoga', data = df, palette = 'hls')
plt.show()
```



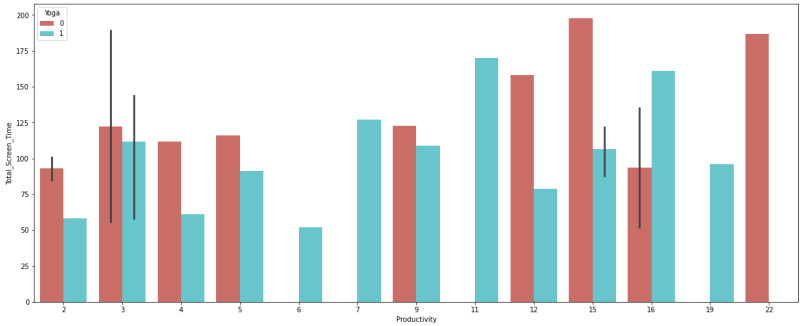
In [56]:

```
plt.figure(figsize=(20,8))
sns.barplot(x = 'Other', y = 'Total_Screen_Time',
            hue = 'Yoga', data = df, palette = 'hls')
plt.show()
```



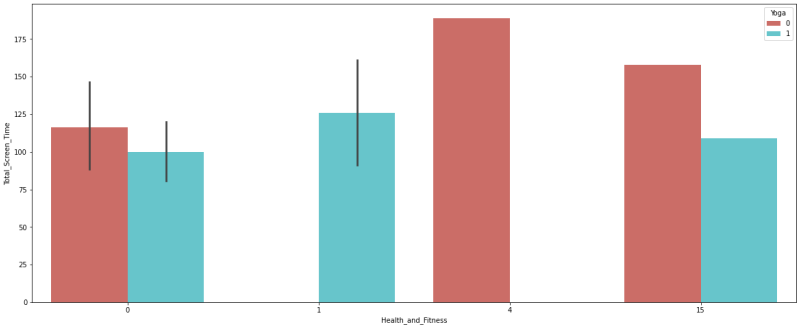
In [57]:

```
plt.figure(figsize=(20,8))
sns.barplot(x = 'Productivity', y = 'Total_Screen_Time',
            hue = 'Yoga', data = df, palette = 'hls')
plt.show()
```



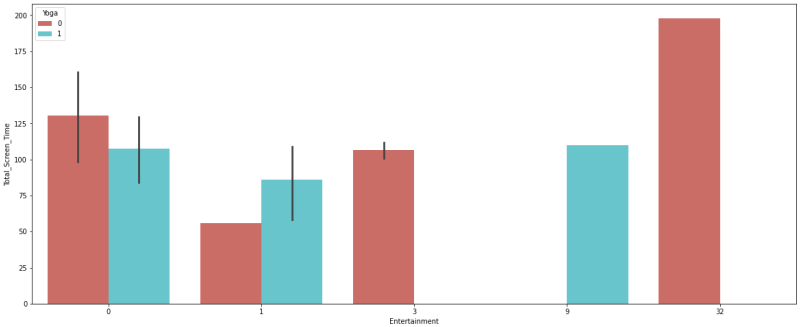
In [58]:

```
plt.figure(figsize=(20,8))
sns.barplot(x = 'Health_and_Fitness', y = 'Total_Screen_Time',
            hue = 'Yoga', data = df, palette = 'hls')
plt.show()
```



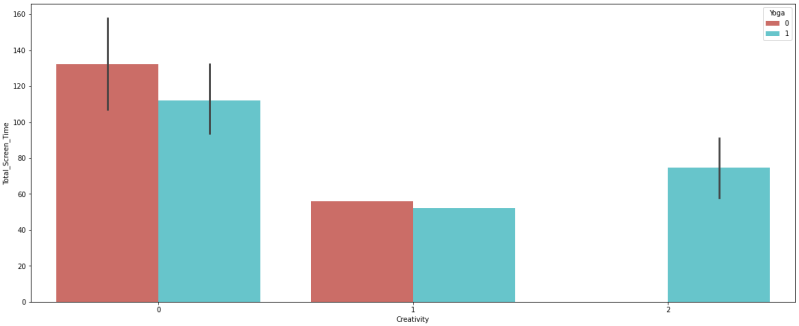
In [59]:

```
plt.figure(figsize=(20,8))
sns.barplot(x = 'Entertainment', y = 'Total_Screen_Time',
            hue = 'Yoga', data = df, palette = 'hls')
plt.show()
```



In [60]:

```
plt.figure(figsize=(20,8))
sns.barplot(x = 'Creativity', y = 'Total_Screen_Time',
            hue = 'Yoga', data = df, palette = 'hls')
plt.show()
```



In [61]:

```
df = df.set_index('Date')
```

In [62]:

```
df
```

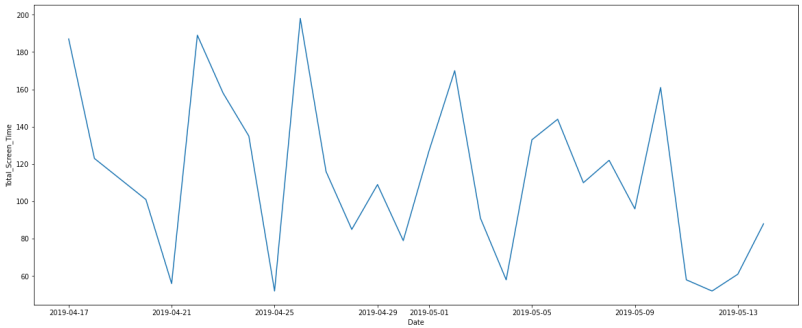
Out[62]:

Date	Week_Day	Total_Screen_Time	Social_Networking	Reading_and_Reference	Other	Product
2019-04-17	Wednesday	187	89	17	41	
2019-04-18	Thursday	123	78	17	8	
2019-04-19	Friday	112	52	40	8	
2019-04-20	Saturday	101	69	9	38	
2019-04-21	Sunday	56	35	2	43	
2019-04-22	Monday	189	68	0	9	
2019-04-23	Tuesday	158	56	18	41	
2019-04-24	Wednesday	135	98	3	33	
2019-04-25	Thursday	52	25	7	3	
2019-04-26	Friday	198	76	8	29	
2019-04-27	Saturday	116	75	10	20	
2019-04-28	Sunday	85	42	22	4	
2019-04-29	Monday	109	46	8	13	
2019-04-30	Tuesday	79	40	2	9	
2019-05-01	Wednesday	127	90	0	10	
2019-05-02	Thursday	170	60	3	2	
2019-05-03	Friday	91	64	2	18	
2019-05-04	Saturday	58	34	4	5	
2019-05-05	Sunday	133	109	5	1	
2019-05-06	Monday	144	81	4	5	
2019-05-07	Tuesday	110	70	5	6	
2019-05-08	Wednesday	122	53	25	26	
2019-05-09	Thursday	96	42	15	16	
2019-05-10	Friday	161	93	13	17	

Date	Week_Day	Total_Screen_Time	Social_Networking	Reading_and_Reference	Other	Product
2019-05-11	Saturday	58	49	1	2	
2019-05-12	Sunday	52	28	1	1	
2019-05-13	Monday	61	37	1	0	
2019-05-14	Tuesday	88	41	2	7	

In [63]:

```
plt.figure(figsize=(20,8))
sns.lineplot(x = df.index, y = 'Total_Screen_Time', data = df, palette = 'hls')
plt.show()
```

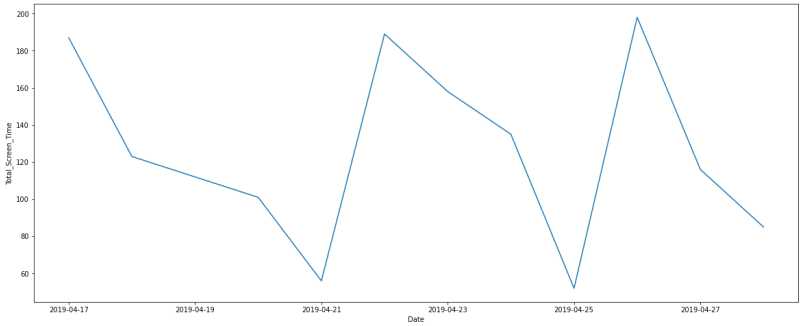


In [64]:

```
df_0 = df[df['Yoga'] == 0]
```

In [65]:

```
plt.figure(figsize=(20,8))
sns.lineplot(x = df_0.index, y = 'Total_Screen_Time', data = df_0,
             palette = 'hls')
plt.show()
```

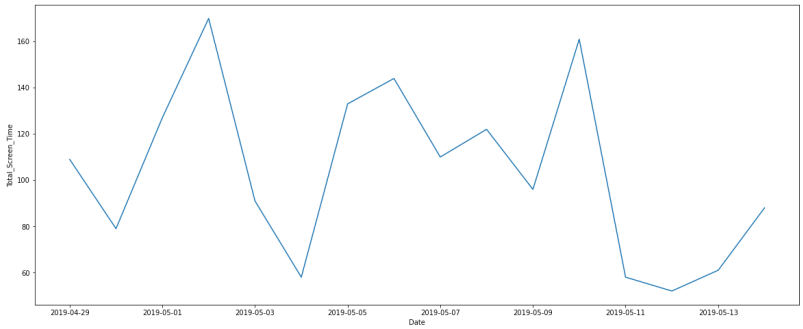


In [66]:

```
df_1 = df[df['Yoga'] == 1]
```

In [67]:

```
plt.figure(figsize=(20,8))
sns.lineplot(x = df_1.index, y = 'Total_Screen_Time', data = df_1,
             palette = 'hls')
plt.show()
```



In [68]:

```
plt.figure(figsize=(15,8))
sns.set(style="darkgrid")
corr = df.corr()
sns.heatmap(corr,annot=True,cmap="crest")
plt.show()
```