```
 1
 2 import pandas as pd
 3 import plotly.express as px
 4 import plotly.graph_objects as go
 5 import numpy as np
 6 import dash
 7 from dash import dcc
 8 from dash import html
 9 from dash.dependencies import Input, Output #
   Wichtig für Callbacks
10
11 external_stylesheets = ['https://codepen.io/
   chriddyp/pen/bWLwgP.css']
12
13 app = dash.Dash(__name__,
   external_stylesheets=external_stylesheets)
14
15 df = pd.read_csv(r'.\Summer-Olympic-medals-
   1976-to-2008.csv', encoding="utf-8")
16 df1 = pd.read_csv(r'.\summary.csv', encoding=
   "utf-8") #Zusammenfassung des grossen
   Datensatzes
17 #df.info()
18 #df1.info()
19
20 colors = {
21     'background': '#F0F8FF',
22     'text': '#00008B'
23 }
24
25 # #Funktionnierender Plot für Tests
26 # fig = px.bar(df1, x=df1['Year'], y=[df1['
   Bronze'],df1['Silver'],df1['Gold']],
   color_discrete_map={'Bronze': 'orange', '
   Silver': 'silver', 'Gold':'gold'}, title="By
```

```
26 Country")
27
28 app.layout = html.Div([
29     html.H1('Oympics', style = {'text-align':
   'center'}),
30
31     html.Div(
32         html.Img(src='https://upload.
   wikimedia.org/wikipedia/commons/thumb/5/5c/
   Olympic_rings_without_rims.svg/1920px-
   Olympic_rings_without_rims.svg.png',width=120
   , alt="Olympic Rings"), style = {'text-align'
   :'center'}
33     ),
34
35     html.Div(
36         "Dashboard by Brunold & Rusconi"
37     ),
38
39     html.Br(),
40
41     #Dropdown
42     dcc.Dropdown(id='dropdownCountry',
43                  options = [{'label': i, '
   value': i} for i in df1['Country'].unique()],
44                  multi = False,
45                  value = 'Switzerland',
46                  style = {"width": "40%"}
47                  ),
48
49     dcc.Graph(id='countryplot', figure
    = {}), #Plot abbilden. in {} kommt der
   Return von der Callback-Funktion
50
51 #  Plot ohne Dropdown und Callback zum
```

```
51 Schauen ob Plots sauber abgebildet werden
52     # dcc.Graph(
53     #     id='example-graph',
54     #     figure=fig
55     # )
56 ])
57 
58 #Callback (Raffi)
59 @app.callback(
60     [Output(component_id='countryplot',
   component_property='figure')],  #Es kommt das
    angepasste Diagramm raus
61     [Input(component_id='dropdownCountry',
   component_property='value')] #Es kommt die
   Auswahl vom Dropdown rein
62 )
63 def update_graph(option_slctd):
64     dff = df1.copy()
65     dff = dff[dff["Country"] == option_slctd]
66 #Plotly Express (Diagramm definieren)
67     fig = px.bar(dff, x='Year', y=['Bronze',
   'Silver', 'Gold'],
68                  color_discrete_map={'Bronze'
   : 'orange', 'Silver': 'silver', 'Gold': 'gold
   '},
69                  labels=dict(value="Number of
    medals won", year="Year", variable="Medal"),
70                  title="Medals won by
   selected country")
71     fig.update_xaxes(
72         dtick=4)
73     return fig,
74 
75 
76 if __name__ == '__main__':
```

```
77      app.run_server(debug=True)
```