



dash.plotly.com

Wat is Dash plotly?

Een “framework” om met weinig code snel “data-apps” te kunnen maken.

Het is geschreven bovenop:

- Flask - webserver functionaliteit
- Plotly.js - grafieken
- React - interactie

Nodig:

```
pip install dash
```

Voor deze presentatie extra:

```
pip install pandas
```

```
pip install openai python-dotenv
```

Waarvoor Dash plotly?

Erg handig/makkelijk voor het snel maken van een specifiek dashboard voor het weergeven / analyseren van data.

Zeker handig als je veel weet van data maar weinig van web development

Wel heel specifiek.

Voor een meer generieke (data weergave / analyse) website: Flask, of PHP, of
mbv bv matplotlib of pandas plot of ...

De werking

Enkele belangrijke aspecten:

1. Maak (“**construct**”) het dash-app object
2. Hoe ziet de data-app er uit? -> app.**layout**
3. Interactie? -> app.**callback**

1 - Maak het dash-app object

```
from dash import Dash
```

```
app = Dash(__name__)
```

2 - app.layout

```
from dash import Dash, html

app = Dash(__name__)

app.layout = html.Div([
    html.Div(children='Hello World')
])

app.run()
```

2 - app.layout

```
from dash import Dash, html, dash_table
import pandas as pd

df = pd.read_csv('Air Quality.csv')

app = Dash(__name__)
app.layout = html.Div([
    html.H1(children='Luchtkwaliteit New York'),
    dash_table.DataTable (data=df.to_dict('records'), page_size=10)
])
app.run(debug=True)
```

2 - app.layout

```
from dash import Dash, html, dash_table, dcc

import pandas as pd

import plotly.express as px

df = pd.read_csv('Air_Quality.csv')

app = Dash(__name__)

app.layout = html.Div([

    html.H1(children=Luchtkwaliteit New York),

    dash_table.DataTable(data=df.to_dict('records'), page_size=10),

    dcc.Graph(figure=px.histogram(df, x='Geo Place Name', y='Data Value', histfunc='avg'))

])

app.run(debug=True)
```


3 - Interactie? -> app.callback

Tabellen en grafieken tot nu toe: statisch

Voor interactie: gebruik (en begrijp) het input/output “callback” mechanisme.

Definitie: een dash callback functie wordt gebruikt wanneer een “Input property” veranderd met als resultaat het veranderen van een andere property (de “Output property”).

Syntax:

```
@callback(  
    Output(id='output'),  
    Input(id='input')  
)  
def update_output_div(inputvalue):  
    return f'Output: {inputvalue}'
```

Straks meer..

Python decorator?

Zie een decorator als een “wrapper”: voegt nieuwe functionaliteit toe aan een object (bv. een functie) zonder de structuur van het object te wijzigen.

In andere woorden: de decorator returned een *gewijzigde* versie van de functie.

Iets meer over functies in Python

- functie kun je toekennen aan een variabele
- functie kun je definiëren binnen een andere functie
- functie kan argument zijn voor een andere functie
- **functie kan een andere functie retourneren**
- scope: geneste functie heeft toegang tot argumenten aanroepende functie

Syntax decorator (callback) aanroep

```
def hallo():  
    return 'Hallo!'  
  
hallo = uppercase_decorator(hallo)
```

Gebruikelijk:

```
@uppercase_decorator # <=> hallo = uppercase_decorator(hallo)  
def hallo():  
    return 'Hallo!'
```

Terug naar de dash callback function

```
@uppercase_decorator # <=> hallo = uppercase_decorator(hallo)

def hallo():

    return 'Hallo!'
```

Vergelijk:

```
@callback(

    Output(component_id='my-output', component_property='children'),

    Input(component_id='my-input', component_property='value')

)

def update_output_div(input_value):

    return f'Output: {input_value}'
```

Een dash callback functie wordt gebruikt wanneer een **Input** property” veranderd met als resultaat het veranderen van een andere property (de **Output** property”).

Een voorbeeld

Gebruik een select list om een grafiek te wijzigen.

```
app.layout = html.Div([

    dcc.Dropdown(df.country.unique(), 'Canada',
id='dropdown-selection' ),

    dcc.Graph(id='graph-content', config={'displayModeBar': False})

])

@callback(

    Input('dropdown-selection', 'value')

    Output('graph-content', 'figure'),

)

def update_graph(value):

    df2 = df[df.country==value]
```

Voorbeeld “eigen” ChatGPT

Referentie

- <https://dash.plotly.com/>
- Ideeën opdoen: <https://plotly.com/examples/>

Alweer bier?