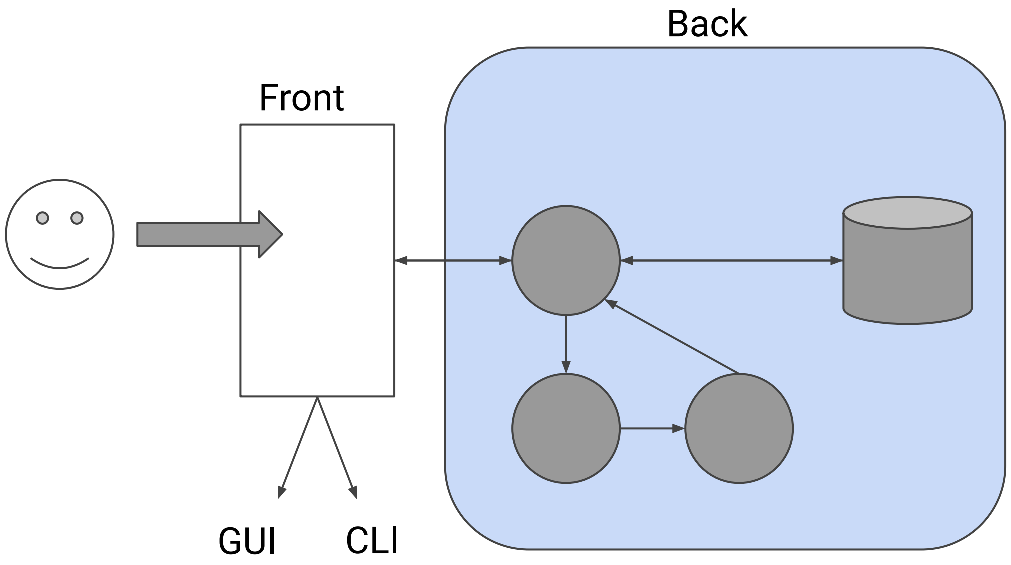
Python Dash

Petr Svarny, 2020

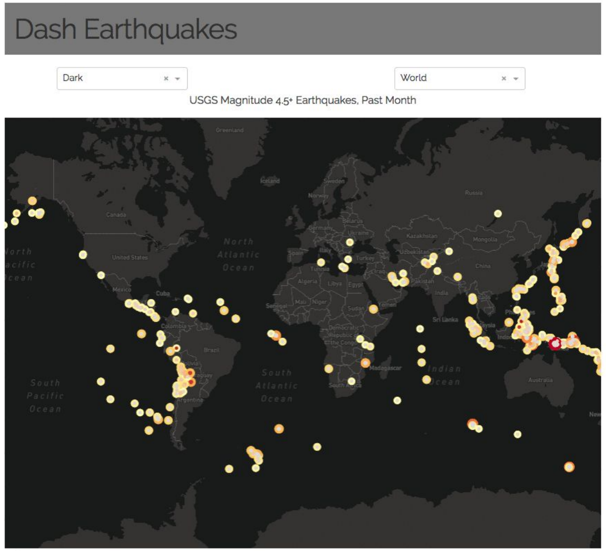
Who wants to make a GUI/“frontend”?

Programming:
BACKEND vs FRONTEND
Metro Goldwyn Mayer
ARS GRATIA ARTIS
TRADE
MARK

Frontend vs backend vs GUI



Cool projects in Dash - earthquakes


Dash Earthquakes

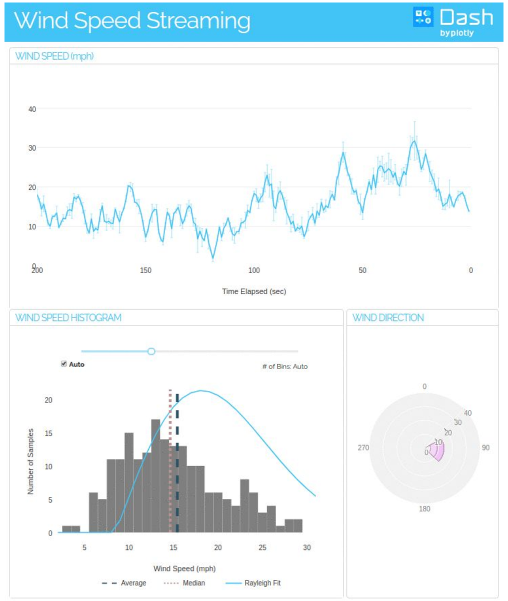
Dark

World

USGS Magnitude 4.5+ Earthquakes, Past Month

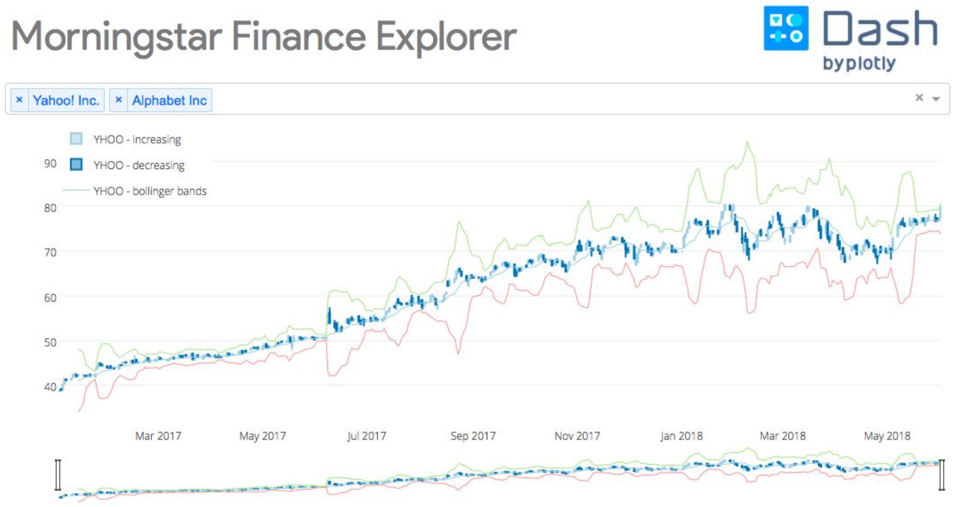
Dash app Github repo

Cool projects in Dash - live wind streaming



Dash app Github repo

Cool projects in Dash



[Dash app Github repo]

What is Dash?

Dash

- Framework for building interactive visualizations
- No previous knowledge of HTML, CSS and JavaScript is needed to build the UI
    - (but it does help to know a little bit)
- Suitable for multiple-users
- You can use real time streaming data
- Tutorials

When do you want to use Dash?

- You do not know enough HTML/CSS/JavaScript
- You want to share (graphical) results of your analyses
- You have little time
- You don't want to involve DEV

Structure of Dash application

```
app = dash.Dash()

app.layout = html.Div([
      dcc.Input(id='input-text', value='initial value', type="text"),
      html.Div(id='display-text')
])

@app.callback(
      Output(component_id='display-text', component_property='children'),
      [Input(component_id='input-text', component_property='value')]
)

def update_output_div(input_value):
      return 'You wrote"{}"'.format(input_value)

if __name__ == '__main__':
      app.run_server(debug=True)
```

Initialize app

Create UI

Define when update

Write update function

Run app

Import Dash and plotly libraries

```
import dash

import dash_core_components as dcc

import dash_html_components as html

import plotly.graph_objs as go
```

Components

- All components in `app.layout` should be part of the html.Div object (you can then put each object into specific Div)
- `dash_html_components`
  - Block - `Div`
  - Header - `H1`
  - Paragraph - `P`
  - `Label`
  - `Button`
- `dash_core_components`
  - `Dropdown`
  - `Graph`
  - `Markdown`

HTML component example - block and label

```
html.Div([
    html.Label('Hello, what do you like to do in your free time?')],
    style={
        'display': 'inline-block', 'vertical-align': 'middle',
        'textAlign': 'center', 'font-size': '1.6em', 'width':
    '40%'
    })
```

Core component example - dropdown menu

```
dcc.Dropdown(
id='example-dropdown',
  options=[
  {'label': 'Read books', 'value': 'read'},
  {'label': 'Bake cakes', 'value': 'bake'},
 ],
 value=''
)
```

Core component example - plot

```
dcc.Graph(
        id='example-plot',
        figure={
            'data': [
               go.Bar(x=[1], y=[628], name='Paperback'),
               go.Bar(x=[1], y=[796], name='Hard book')
            ],
            'layout': {
                'title': 'Book weight in grams'
                }
            }
    )
```

Run application locally

- In your terminal type
  - python name_of_the_app.py
  - python3 name_of_the_app.py
- You will see similar output

```
Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
```

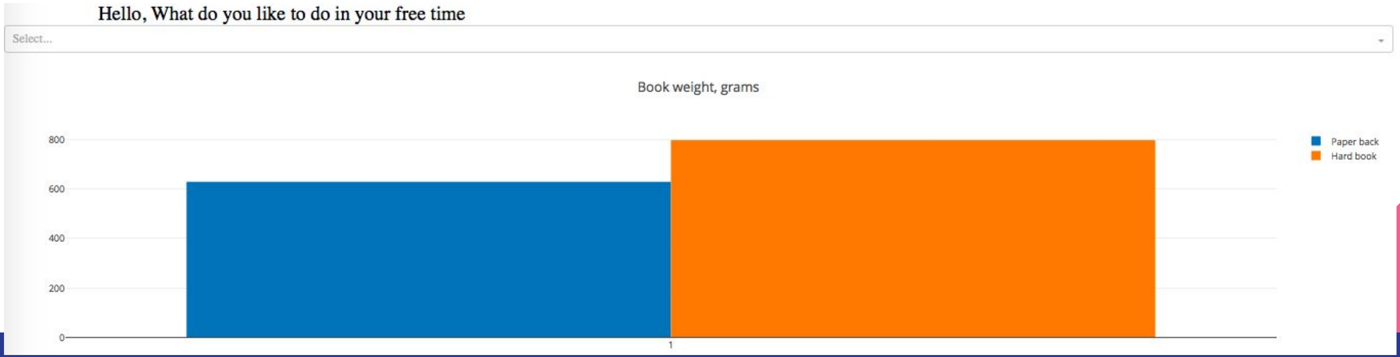
- Type http address in your browser (or Ctrl+click/ Cmd+click on address)

Run application locally

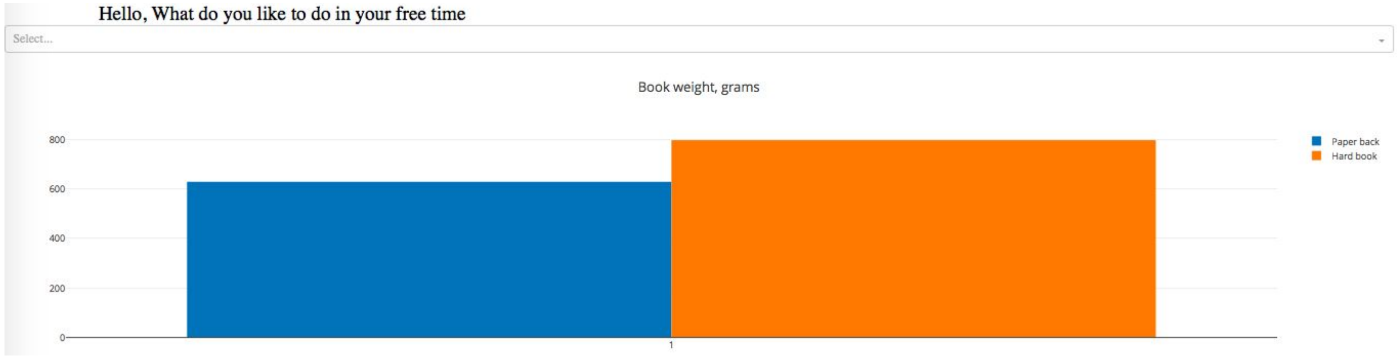
- Use the following code to automatically apply saved changes in your application
  - `app.run_server(debug=True)`
- In other case:
  - Stop app by Ctrl+C
  - Run application again
- **Do not forget to save changes!**

Exercise

- From the blocks in the slides above make an example dash application (dash_app_book.py)
    - Import statements
    - HTML component - Div . in Div put:
        - Dash core component - Dropdown
        - Dash core component - Plot
- Run the application by typing in terminal: python (or python3) dash_app_book.py
- You must see the following image in your browser


Hello, What do you like to do in your free time

When we put all examples together...


Hello, What do you like to do in your free time

We will now add interactivity

app.callback

- Input
  - When to run the underlying function
  - Can be multiple
- Output
  - What element of UI to change, once `@app.callback` is activated
  - Can be only one - create another callback for additional output
- More examples

```
@app.callback(
    Output(component_id='example-plot', component_property='figure'),
    [Input(component_id='example-dropdown', component_property='value')]
)
def update_plot(choice):
    if choice = ... :
        data =...
    else:
        data = ...
    return data
```

Component property

- Is used to define the attribute of the element needed to be changed or is activating change
- `'value'`
- `'children'`
- `'figure'`
- Can be also an event
  - `'clickData'`

Adding external CSS file

- Use following syntax

```
app.css.append_css({
    'external_url': my_css_url
})
```

- Tutorial

Callback with state

- When you don't want to fire callback immediately
- Use `dash.dependencies.State`
- Callback function will be activated only when `dash.dependencies.Input` will change
- More information

Callback with state - example

Callback is triggered by listening to the `n_clicks` property of the `html.Button` component.

```
app.layout = html.Div([
    dcc.Input(id='day', type="text", value='Day'),
    dcc.Input(id='time', type="text", value='Time'),
    html.Button(id='submit-button', n_clicks=0, children='Show weather
forecast'),
    html.Div(id='show-weather')
])

@app.callback(Output('show-weather', 'children'),
              [Input('submit-button', 'n_clicks')],
              [State('day', 'value'),
               State('time', 'value')])
```

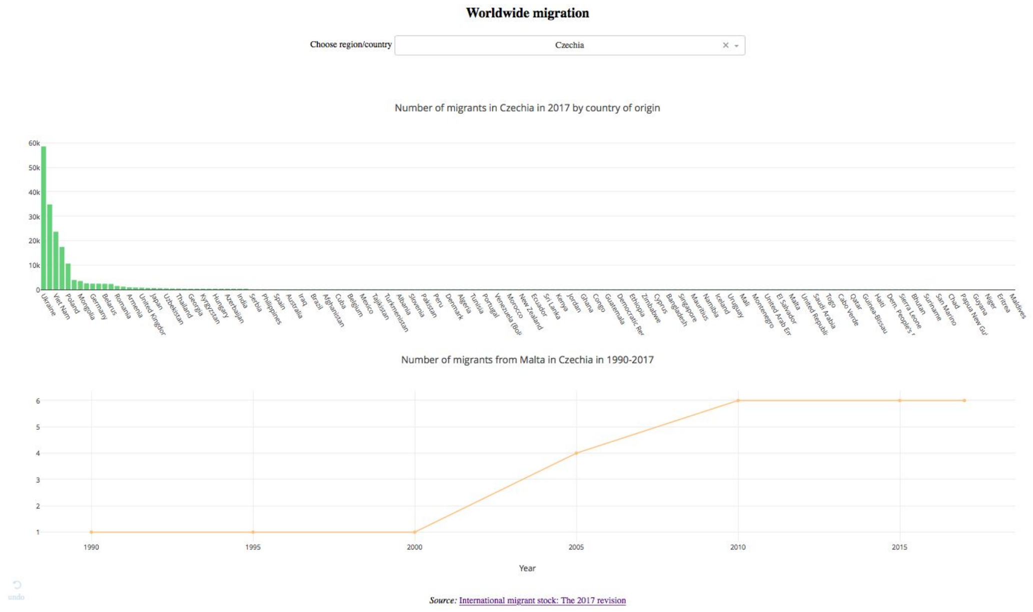
Callback with state - example

```
app.layout = html.Div([
    dcc.Input(id='day', type="text", value='Day'),
    dcc.Input(id='time', type="text", value='Time'),
    html.Button(id='submit-button', n_clicks=0, children='Show weather forecast'),
    html.Div(id='show-weather')
])

@app.callback(Output('show-weather', 'children'),
              [Input('submit-button', 'n_clicks')],
              [State('day', 'value'),
               State('time', 'value')])

def update_function(n_clicks, day, time):
    ...
```

Exercise

Use the slides above to add some additional interactivity of your choice to your dashboard.

Dash - example project



Worldwide migration

Choose region/country: Czechia

Number of migrants in Czechia in 2017 by country of origin

Number of migrants from Malta in Czechia in 1990-2017

Source: International migrant stock: The 2017 revision

http://dash-migration.herokuapp.com

Used data

- Migration data from UN report
  - International migrant stock: The 2017 revision
- Data from 232 countries in 1990, 1995, 2000, 2005, 2010, 2015 and 2017
- Person is considered to be migrant if his place of birth/citizenship is foreign country

Advanced Dash

- Streaming data
- Sharing state between callbacks
- Deploying Dash app