Advanced Programming with Python

Dash

Pepe García jgarciah@faculty.ie.edu

Plan for today

- Questions about assignment?
- Learn about dash

Dash

https://dash.plot.ly

Dash is a library for creating data visualizations. A big difference with other libraries is that we'll be able to do everything in Python, we won't need any JS to make it work.

Installing dash

Dash is **not included in Anaconda**, we need to install it ourselves. We'll use a virtual environment for that.

(remember, $ indicates a new command in the terminal, don't write it)

```
$ python3 -m venv venv
$ source venv/bin/activate
$ pip install -r requirements.txt
```

After all this, modify the interpreter in VSCode too.

Dash. Layout

There are a couple of things we'll need to import from the main dash module:

- `from dash import html`
- `from dash import dcc`
- `from dash import Dash`

`html` is used to create HTML programatically

`dcc` (stands for dash core components) is used to create more interesting visual components, such as graphs, or selectors.

`Dash` is the entry point for our application.

Dash. Layout

We can create HTML layouts directly in Python with Dash

```
from dash import dcc # stands for dash core components
from dash import html

app = Dash(__name__)
app.layout = html.Div(children = [
    html.H1("title"),
    dcc.Dropdown(
        id="Are you over 18?",
        options={"yes": True, "no": False},
        multi=False,
        value=[]
    )
])
```

Practice

see **example1.py**

Dash. Callbacks

callbacks make our Dash applications interactive. They're functions that whenever an **input** component changes, will change an **output** component

```python
@app.callback(
    Output(component_id="accidents-graph",component_property="figure"),
    [Input(component_id="district", component_property="value")]
)
def update(districts):
    pass
```

Practice

let's see a simple example of callbacks.

example2.py

Dash. Graphs

We can introduce plotly graphs in our HTML layouts with the **dcc.Graph** class.

```
dcc.Graph(
    id='first-graph',
    figure={
        'data': [
            {'x': [1, 2, 3], 'y': [4, 1, 2], 'type': 'bar', 'name': 'bar1
            {'x': [1, 2, 3], 'y': [2, 4, 5], 'type': 'bar', 'name': 'bar2
        ],
        'layout': {
            'title': 'Dash Data Visualization'
        }
    }
)
```

Practice

But, callbacks can do much more than that, they can modify graphs whenever some component value is changed by the user.

example3.py

Working with real data

For the following example we'll use the dataset that Madrid government provides about bike accidents.

You can find more interesting datasets here:

https://datos.madrid.es/portal/site/egob/

Working with real data

In the following example we'll visualize how the amount of bike accidents change by district.

example4.py

Working with real data

Exercise

Modify **example4.py** so that it filters by accident type (**TIPO ACCIDENTE**) too.

Practice

Example 5

It's also possible to update more than one graph at the same time, but we'll need to do that with different callbacks.
See **example5.py**

Materials

- https://dash.plotly.com
- https://plotly.com/python/plotly-express/