

# Advanced Programming with Python

## Dash

Pepe García [jgarciah@faculty.ie.edu](mailto:jgarciah@faculty.ie.edu)

# Plan for today

- Learn about dash
- Questions regarding assignment

<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 other language.

There are a couple of modules we'll need to import:

- `import dash_html_components as html`
- `import dash_core_components as dcc`

`html` is used to create all HTML tags we're already used to

`dcc` is used to create more interesting visual components, such as graphs, or selectors.

# Dash. Layout

We can create HTML layouts directly in Python with Dash

```
import dash_core_components as dcc
import dash_html_components as html
```

```
html.Div(children = [
    html.H1("title"),
    dcc.Dropdown(
        id="district",
        options=districts,
        multi=True,
        value=["SALAMANCA", "ARGANZUELA"]
    )
])
```

It's important to add unique `id` attributes to all elements that will be used interactively.

# Dash. Graphs

```
import dash_core_components as dcc
import dash_html_components as html

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



see **example1.py**

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

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



let's see

**example2.py**

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/>

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

**example4.py**

## Exercise

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

## 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**

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

