

Проектирование и разработка дашбордов в dash

Импорт библиотек для дашбордов

```
In import dash # библиотека dash
import dash_core_components as dcc # элементы управления dash
import dash_html_components as html # элементы дашборда dash
import plotly.graph_objs as go # графики plotly
```

Задание лейаута дашборда

```
In external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']
app = dash.Dash(__name__, external_stylesheets=external_stylesheets)
app.layout = html.Div(children=[
    # ваш код для графиков и элементов управления
])
```

Заголовок дашборда

```
In html.H1(children = 'Заголовок'),
```

График дашборда

```
In dcc.Graph(
    figure = {
        'data': [
            # набор графических элементов plotly, отображаемых на графике
        ],
        'layout': # набор визуальных свойств графика
    },
    # параметр опускается, если формируется динамически
    id = 'graph_id' # идентификатор графика
),
```

Отображения графика-таблицы

```
In go.Table(header = {'values': ['<b>Страна</b>',
                                '<b>Год</b>',
                                '<b>% городского населения</b>'],
                  'fill_color': 'lightgrey', # цвет заливки фона ячеек
                  'align': 'center'}, # выравнивание текста
            cells = {'values': table.T.values})
```

Элемент управления - выбор временного периода

```
In dcc.DatePickerRange(
    start_date = urbanization['Year'].dt.date.min(), # дата начала
                                                    # со значением по умолчанию
    end_date = urbanization['Year'].dt.date.max(), # дата окончания
                                                    # со значением по умолчанию
    display_format = 'YYYY', # формат отображения даты и времени
    id = 'dt_selector', # уникальный идентификатор элемента управления
),
```

Добавление пустой строки между элементами дашборда

```
In html.Br()
```

Элемент управления - набор флажков

```
In dcc.Checklist(options = [{'label': 'Африка', 'value': 'afr'},
                             {'label': 'Евразия', 'value': 'eur'},
                             {'label': 'Австралия', 'value': 'au'},
                             {'label': 'Америка', 'value': 'am'}], # массив опций выбора

# label – значение, которое видит пользователь
# value – техническое значение для обработки внутри программы

value = ['afr', 'eur', 'au', 'am'], # набор value, выбранных по умолчанию
id = 'continent_selector' )         # уникальный идентификатор элемента управления
```

Элемент управления - выпадающий список

```
In dcc.Dropdown(options = [{'label': 'Африка', 'value': 'afr'},
                             {'label': 'Евразия', 'value': 'eur'},
                             {'label': 'Австралия', 'value': 'au'},
                             {'label': 'Америка', 'value': 'am'}], # массив опций выбора

# label – значение, которое видит пользователь
# value – техническое значение для обработки внутри программы

value = ['afr', 'eur', 'au', 'am'], # набор value, выбранных по умолчанию
multi = True,                       # возможностью выбрать несколько элементов из списка
id = 'continent_selector')          # уникальный идентификатор элемента управления
```

Элемент управления - радиокнопка

```
In dcc.RadioItems(options = [{'label': 'Чай', 'value': 'tea'},
                              {'label': 'Кофе', 'value': 'coffee'},
                              {'label': 'Потанцуем', 'value': 'lets_dance'}], # массив опций выбора

# label – значение, которое видит пользователь
# value – техническое значение для обработки внутри программы

value = 'tea',                      # набор value, выбранных по умолчанию
id = 'drink_selector')              # уникальный идентификатор элемента управления
```

Описываем логику дашборда, зависящего от элементов управления

```
In @app.callback(
    [Output('trig_func', 'figure')],
    [Input('mode_selector', 'value')],
    [])
def update_figures(selected_mode):
    # код обновления графика
```

Элемент div

```
In html.Div([
    html.Div([
        html.Label('Я шириной 9 колонок из 12-ти возможных:'),
    ], className = 'nine columns'),
    html.Div([
        html.Label('Я шириной 3 колонки из 12-ти возможных:'),
    ], className = 'three columns'),
], className = 'row'),
```

Случай множественных входов

```
In @app.callback(
    [Output('plot_1', 'figure'),
     Output('plot_2', 'figure')],
    [Input('control_1', 'value'),
     Input('control_2', 'value'),
     Input('control_3', 'value')]
)
def update_figures(control_1_value, control_2_value, control_3_value):
    # формируем динамические графики
    return (
        { # это график для plot_1
          'data': plot_1_data,
          'layout': go.Layout(...)
        },
        { # это график для plot_2
          'data': plot_2_data,
          'layout': go.Layout(...)
        },
    )
)
```

Отображение графика высотой 25% от ширины страницы

```
In dcc.Graph(
    style = {'height': '25vw'},
    id = 'sales_by_platform'
)
```