# layout.py modul működése

## Összefoglaló

A layout.py létrehoz egy adott fülhöz tartozó összes komponenst, létrehozz az alkalmazás teljes elrendezését a fülek generálásával együtt. Az oszlopválasztók és a fül tartalmak az adatkészlet alapján automatikusan generáljuk le.

## Részletes működés

1. create\_tab függvény létrehoz egy dinamikus fület az alkalmazáshoz.

Minden fülhöz tartozik:

* + Oszlopválasztó
  + Jövedelem típus választó
  + Diagram típus választó
  + Egy grafikon az adatok megjelenítésére

Paraméterei:

* tab\_id: a fül azonosítója
* label: a fül elnevezése
* column\_options: oszlopválasztó lehetőségei
* income\_type\_id: azonosító a jövedelem típus választóhoz
* chart\_type\_id: azonosító a diagram típus választóhoz
* graph\_id: azonosító a grafikon komponenshez

def create\_tab(tab\_id, label, column\_options, income\_type\_id, chart\_type\_id, graph\_id):

1. fül definiálása, létrehozz a fület a megadott címkével és tartalommal

return dcc.Tab(

        label=label,

        children=[

            html.Div(

                children=[

1. Oszlopválasztó, amely megjeleníti az oszlopok listáját. A tab\_id az oszlopok alapján egyedi, hogy a kiválasztott oszlopok ne jelenjenek meg a többi fülön.

html.Div(

                children=[

                      html.Div(

                        children=[

                    html.Label("Válassz oszlopokat:"),

                    dcc.Checklist(

                        id=f"{tab\_id}-columns-selector",

                        options=[

                            {'label': col, 'value': val} for col, val in column\_options

                        ],

                        value=[],

                        style={'margin': '-4px','margin-top':'10px'}

                    ),],

                        className="text-box",

                        ),

1. létrehoz egy jövedelem típus választót, aminek az alapértelmezett értéke a Bruttó jövedelem.

html.Div(

                children=[

                    html.Label("Válassz jövedelemtípust:"),

                    dcc.Checklist(

                        id=income\_type\_id,

                        options=[

                            {'label': 'Bruttó', 'value': 'Bruttó jövedelem'},

                            {'label': 'Nettó', 'value': 'Nettó jövedelem'}

                        ],

                        value=['Bruttó jövedelem'],

                        style={'margin': '-4px','margin-top':'10px'}

                    ),],

                        className="text-box",),

1. létrehoz egy diagram típus választót, alapértelmezettként a vonal van beállítva

html.Div(

                children=[

                    html.Label("Válassz diagramtípust:"),

                    dcc.Checklist(

                        id=chart\_type\_id,

                        options=[

                            {'label': 'Vonal', 'value': 'Vonal'},

                            {'label': 'Pont', 'value': 'Pont'},

                            {'label': 'Lineáris regresszió', 'value': 'Lineáris regresszió'}

                        ],

                        value=['Vonal'],

                        style={'margin': '-4px','margin-top':'10px'}

                    ),],

                        className="text-box",),

                ],

                className="check-box",

                    ),

1. hozzáad egy gafikont, amely az adatokat ábrázolja

                    dcc.Graph(id=graph\_id),

1. létrehozz az egész alkalmazás elrendezését, amely tartalmazza az összes fület és komponenseit. Paramétere a teljes adatkészlet.

def create\_layout(data):

1. Oszlopok egyszerűsítése, amennyiben nem „Év” vagy „Jövedelem típus” áll az oszlop elnevezésében, akkor keressünk „:”. Amennyiben találtunk, töröljük az előtte szereplő szöveget a „:”-tal együtt.

simplified\_columns = [

        (col.split(":")[-1].strip(), col) for col in data.columns if col not in ['Jövedelem\_típus', 'Év']

    ]

1. négy különálló fül létrehozása create\_tab függvénnyel:
   * Országos: csak az Ország összesen oszlopot jelenítse meg.
   * Régiók szerint: csak a Régiók szerint oszlopot jelenítse meg
   * Települések típusa szerint: csak a Települések típusa szerint oszlopot jelenítse meg.
   * Összesített: az összes oszlopot jelenítse meg.

tabs = [

        create\_tab(

            tab\_id="orszagos",

            label="Országos",

            column\_options=[('Ország összesen', 'Ország összesen')],

            income\_type\_id="orszagos-income-type-selector",

            chart\_type\_id="orszagos-chart-type-selector",

            graph\_id="orszagos-chart"

        ),

        create\_tab(

            tab\_id="regions",

            label="Régiók szerint",

            column\_options=[

                (col.split(":")[-1].strip(), col) for col in data.columns if "Régiók szerint" in col

            ],

            income\_type\_id="regions-income-type-selector",

            chart\_type\_id="regions-chart-type-selector",

            graph\_id="regions-chart"

        ),

        create\_tab(

            tab\_id="settlements",

            label="Települések típusa szerint",

            column\_options=[

                (col.split(":")[-1].strip(), col) for col in data.columns if "Települések típusa szerint" in col

            ],

            income\_type\_id="settlements-income-type-selector",

            chart\_type\_id="settlements-chart-type-selector",

            graph\_id="settlements-chart"

        ),

        create\_tab(

            tab\_id="all",

            label="Összesített",

            column\_options=simplified\_columns,

            income\_type\_id="all-income-type-selector",

            chart\_type\_id="all-chart-type-selector",

            graph\_id="all-chart"

        )

    ]

1. a füleket egy dcc.Tabs komponensbe rendezi. Visszaadja az egész elrendezést.

 return html.Div([

        dcc.Tabs(tabs)

    ])