

Adatbányászat a Gyakorlatban

2. Gyakorlat: Dash alapok

Kuknyó Dániel
Budapesti Gazdasági Egyetem

2024/25
1.félév

- 1 Bevezetés
- 2 Dash komponensek
- 3 Elrendezés

1 Bevezetés

2 Dash komponensek

3 Elrendezés

Órai környezet telepítése

- 1 Új Anaconda környezet létrehozása dash néven:

```
$ conda create --name dash python=3.12
```

- 2 Környezet aktiválása:

```
$ conda activate dash
```

- 3 Órai tárhely klónozása:

```
$ git clone https://github.com/basictask/Adatbanyaszat.git
```

- 4 Függőségek telepítése:

```
$ cd Adatbanyaszat  
$ pip install -r requirements.txt
```

A Dash keretrendszer

A Dash keretrendszer segítségével lehetséges interaktív, dinamikus adatalapú műszerfalakat és alkalmazásokat készíteni szintisztán Python nyelvben.

A Dash a **Flask** mikrokeretrendszert használja backend szerverként, **Plotly** segítségével jelenítui meg a diagramokat és **React** komponenseket használ a felhasználói interakció kezelésére.

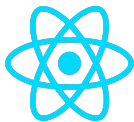


Flask

+



+



=



plotly | Dash

A Dash könyvtár komponensei

Dash

Ez a fő csomag, amely bármely alkalmazás gerincét biztosítja a `dash.Dash` objektumon keresztül.

Emellett néhány más eszközt is biztosít az interaktivitás és kivételek kezeléséhez, amelyekről később fogunk beszélni, amikor építjük az alkalmazásunkat.

A Dash könyvtár komponensei

Dash Core Components

Egy csomag, amely interaktív komponensek készletét biztosítja, amelyeket a felhasználók manipulálhatnak.

Legördülő menük, dátumválasztók, csúszkák és sok más komponens is megtalálható ebben a csomagban.

A Dash könyvtár komponensei

Dash HTML Components

Ez a csomag az összes elérhető HTML címkét Python osztályként biztosítja. Egyszerűen átalakítja a Python HTML-re.

Például, Pythonban a `dash_html_components.H1('Hello, World')` kód átalakul `<h1>Hello, World</h1>` HTML kóddá.

A Dash könyvtár komponensei

Dash Bootstrap Components

Ez egy harmadik féltől származó csomag, amely Bootstrap funkcionalitást ad a Dash-hez. Ez a csomag és annak komponensei számos elrendezéssel és vizuális jelekkel kapcsolatos lehetőséget kezelnek.

Az elemek egymás mellé vagy egymás fölé helyezése, méretük meghatározása a böngésző képernyőmérete alapján, valamint kódolt színek készletének biztosítása a jobb kommunikáció érdekében a felhasználókkal.

Egy Dash alkalmazás struktúrája

- Importálások:

```
1 import dash
2 import dash_html_components as
   html
3 import dash_core_components as dcc
```

- Elrendezés:

```
1 app.layout = html.Div([
2     dcc.Dropdown()
3     dcc.Graph()
4     ...
5 ])
```

- Visszahívási függvények:

```
1 @app.callback()
2 ...
3 @app.callback()
4 ...
```

- Alkalmazás példányosítása:

```
1 app = dash.Dash(__name__)
```

- Alkalmazás futtatása:

```
1 if __name__ == '__main__':
2     app.run_server()
```

- 1 Bevezetés
- 2 Dash komponensek
- 3 Elrendezés

Egy kezdeti alkalmazás

- ① Egy új, `app.py` fájlban a következő csomagok importálásával:

```
1 import dash
2 import dash_core_components as dcc
```

- ② Alkalmazás példányosítása:

```
1 app = dash.Dash(__name__)
```

- ③ Alkalmazás elrendezésének létrehozása:

```
1 app.layout = html.Div([
2     html.H1('Hello, World!')
3 ])
```

- ④ Futtatás:

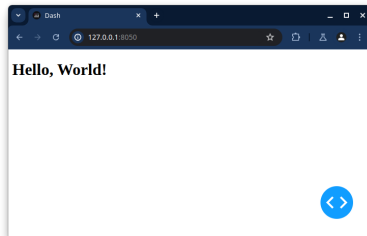
```
1 if __name__ == '__main__':
2     app.run_server(debug=True)
```

Az alkalmazás futtatása

A Python értelmező segítségével a megfelelő könyvtárban állva az `app_v1_1.py` fájl futtatásával az eredmény a következő:

```
1 (dash) daniel@neptune:~/Documents/BGE/  
   Adatbanyaszat/2_dash/code$ python  
   app_v1_1.py  
2 Dash is running on http  
   ://127.0.0.1:8050/  
3  
4 * Serving Flask app 'app_v1_1'  
5 * Debug mode: on
```

A böngészőben a `127.0.0.1:8050` címre navigálva a következő eredmény látható:



Komponensek hozzáadása az alkalmazáshoz

Komponensek hozzáadása az alkalmazás elrendezésének szerkesztésével érhető el (`app.layout`). Ez meglehetősen egyszerű, csak a legfelső szintű `html.Div` komponens `children` attribútumához kell hozzáfűzni a megfelelő elemeket.

```
1 html.Div(children=[component_1, component_2, component_3, ...])
```

Div

A `Div` a HTML-ben egy blokk szintű elem, amely képes egy dokumentum különböző komponenseit csoportosítani. A `Div` nem rendelkezik semmilyen alapértelmezett stílussal vagy viselkedéssel.

HTML komponensek hozzáadása

children

Ez az első, és a fő konténere a komponenseknek. Paraméterül kaphatja elemek listáját vagy egyetlen elemet is.

className

Ez megegyezik a HTML class attribútumával.

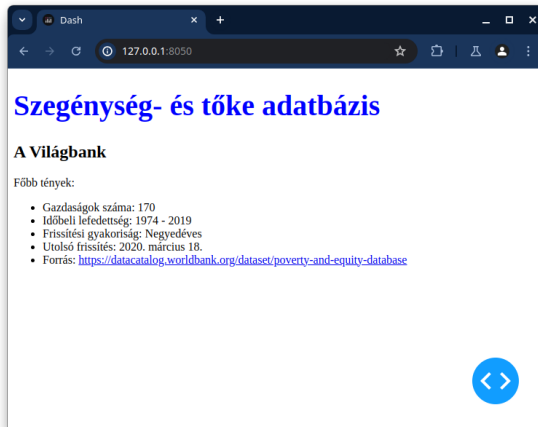
id

A komponens azonosítója. Az interaktivitás megvalósításában van kulcsfontosságú szerepe

style

Ez megfelel az azonos nevű HTML attribútumnak azzal a különbséggel, hogy camelCase stílust használ a változók elnevezésére.

A Dash alkalmazás HTML komponensekkel



1 Bevezetés

2 Dash komponensek

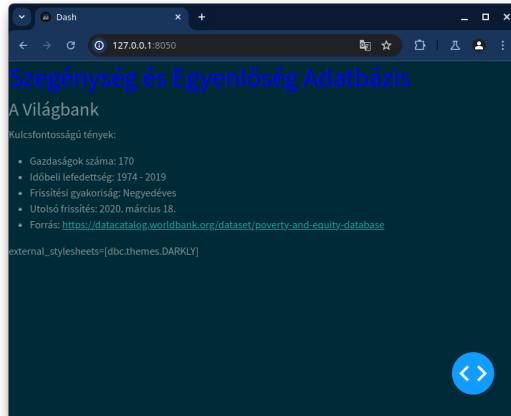
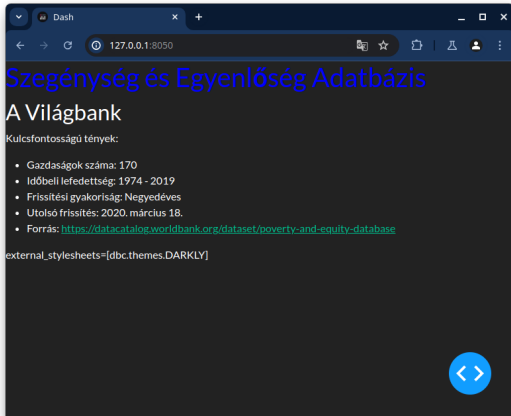
3 Elrendezés

Témák

Egy Dash alkalmazás témájának megváltoztatása rendkívül egyszerű: a Dash objektum létrehozásakor kell egy új téma argumentumot bevinni a konstruktor függvénybe.

```
1 import dash_bootstrap_components as dbc
2 ...
3 app = dash.Dash(__name__, external_stylesheets=[dbc.themes.BOOTSTRAP])
4 ...
```

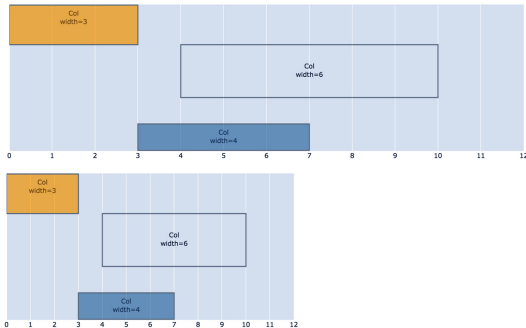
Témák az előző alkalmazásban



A rács rendszer

A Bootstrap segítségével lehetséges oszlopokat definiálni, ami egy független képernyőként viselkedik, egymás fölött megjelenítve az elemeket.

A rács rendszer 12 oszlopra bontja a képernyőt, és egy komponens szélessége oszlopok számában adható meg.

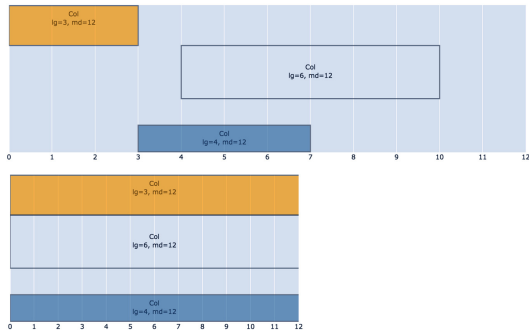


```
1 import dash_bootstrap_components as dbc
2 dbc.Col(children=[child1, child2, ...])
```

Rácsok dinamikus képernyő méreten

Vannak olyan esetek, amikor nem kívánatos az elemek méretezése a képernyővel együtt. Amikor a képernyő kisebb lesz, némelyik komponenseknek jó, ha kiterjednek méretükben.

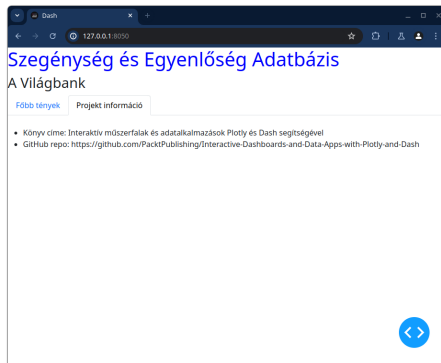
Öt különböző méretet lehet definiálni: xs (extra-small), md (medium), lg (large), xl (extra-large).



```
1 import dash_bootstrap_components as dbc
2 dbc.Col(children=[child1, child2, ...], lg=6, md=12)
```

Bootstrap komponensek hozzáadása az alkalmazáshoz

Az alkalmazás következő verziójában két új komponens kerül hozzáadásra, a Tabs és Tab. Ezek szorosan kapcsolódnak egymáshoz. A Tabs a Tab konténere. Ennek eredménye egy informatívabb és jobban elrendezett alkalmazás.



asdasd