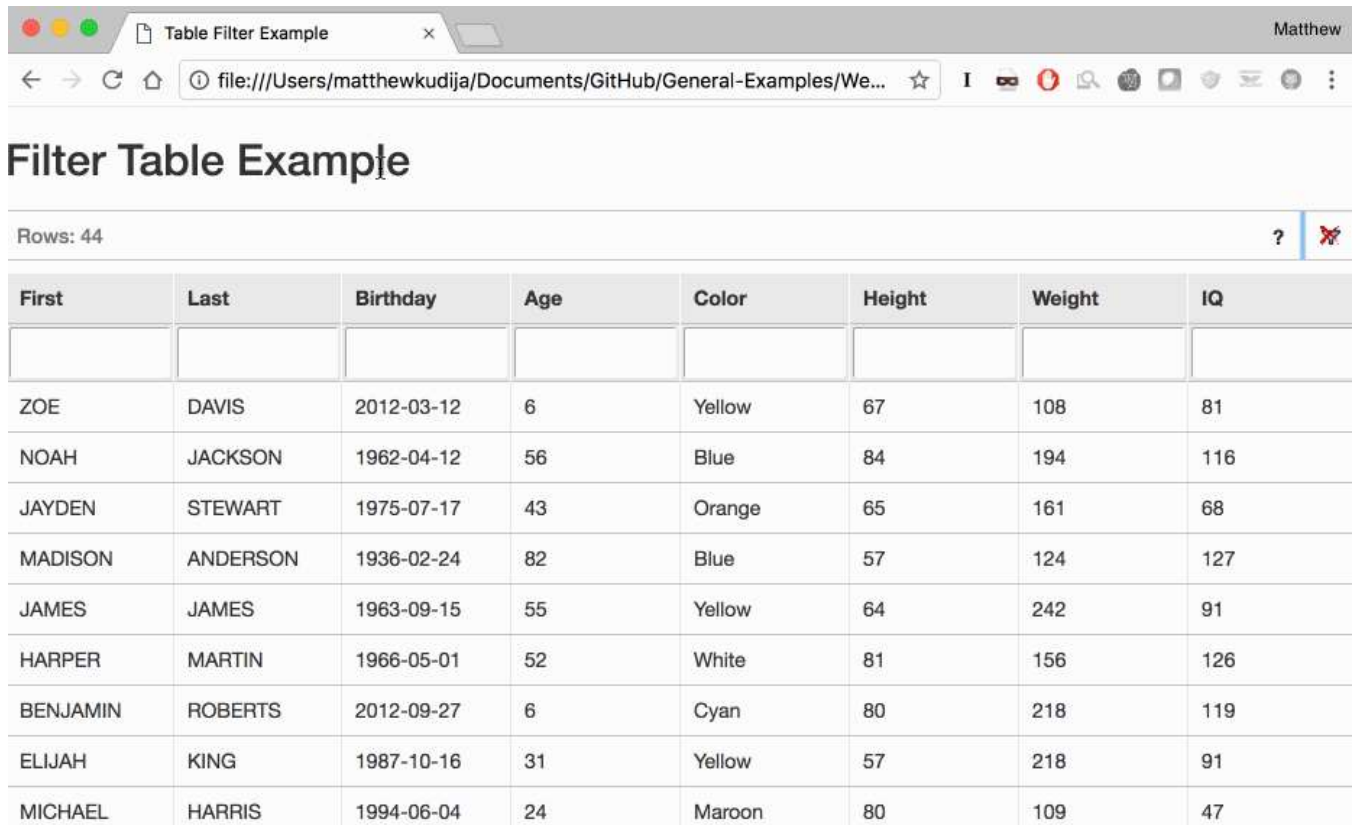


Filtrage de table en HTML et JavaScript

mar 03 avril 2018



First	Last	Birthday	Age	Color	Height	Weight	IQ
ZOE	DAVIS	2012-03-12	6	Yellow	67	108	81
NOAH	JACKSON	1962-04-12	56	Blue	84	194	116
JAYDEN	STEWART	1975-07-17	43	Orange	65	161	68
MADISON	ANDERSON	1936-02-24	82	Blue	57	124	127
JAMES	JAMES	1963-09-15	55	Yellow	64	242	91
HARPER	MARTIN	1966-05-01	52	White	81	156	126
BENJAMIN	ROBERTS	2012-09-27	6	Cyan	80	218	119
ELIJAH	KING	1987-10-16	31	Yellow	57	218	91
MICHAEL	HARRIS	1994-06-04	24	Maroon	80	109	47

Les pages GitHub facilitent le partage d'informations : envoyez simplement l'URL au lieu d'envoyer une pièce jointe par e-mail. Lorsque je voulais partager le contenu d'un fichier Excel, c'était une plate-forme naturelle, mais avec plus de 1 000 lignes de données, j'avais besoin d'un moyen de filtrer le tableau sur la page Web.

Avec mes connaissances à l'époque limitées à du HTML de base et très peu de JavaScript, il s'agit d'un exercice de piratage d'une solution. Continuez à lire pour voir comment j'ai implémenté une solution à l'aide de la bibliothèque JavaScript TableFilter .

(<http://koalyptus.github.io/TableFilter/>)

Exigences et recherche

Avec nos données à partager dans une feuille de calcul locale, l'exigence de base est la possibilité de filtrer un tableau sur un site Web sur plusieurs colonnes comme vous le pouvez dans une feuille de calcul :

	A	B	C	D	E	F	G	H	I	J
1	First	Last	Birthday	Age	Color	Height	Weight	IQ		
2	ZOE	DAVIS	3/12/12	6	Yellow	67	108	89		
3	NOAH	JACKSON	4/12/62	56	Blue	84	194	152		
4	JAYDEN	STEWART	7/17/75	43	Orange	65	161	108		
5	MADISON	ANDERSON	2/24/36	82	Blue	57	124	132		
6	JAMES	JAMES	9/15/63	55	Yellow	64	242	135		
7	HARPER	MARTIN	5/1/66	52	White	81	156	63		
8	BENJAMIN	ROBERTS	9/27/12	6	Cyan	80	218	71		
9	ELIJAH	KING	10/16/87	31	Yellow	57	218	75		
10	MICHAEL	HARRIS	6/4/94	24	Maroon	80	109	73		
11	ARIA	WILSON	10/18/06	12	Pink	74	197	52		
12	MIA	WILLIAMS	7/15/93	25	Red	84	136	75		
13	AIDEN	LEWIS	10/19/13	5	White	79	214	96		
14	SOPHIA	SMITH	12/9/13	5	Blue	80	196	105		
15	JACOB	JOHN	11/8/06	12	Purple	67	170	107		
16	LAYLA	HALL	7/10/08	10	Green	58	140	93		
17	ABIGAIL	WOOD	8/27/50	68	Pink	70	199	128		
18	LOGAN	LEE	10/12/09	9	Blue	61	112	91		
19	OLIVER	EVANS	8/22/32	86	Pink	84	80	106		

Étant à l'aise en Python, mon premier réflexe est de filtrer

(<https://github.com/mkudija/General-Examples/blob/master/Pandas/filter.md>) un Pandas DataFrame. Mais nous hébergeons ceci sur les pages GitHub en tant que site Web statique et nous ne pouvons pas exécuter le filtrage côté serveur. Par conséquent, nous utiliserons JavaScript pour filtrer la table côté client dans le navigateur.

J'ai pu trouver un certain nombre

(https://www.w3schools.com/howto/howto_js_filter_table.asp) d'exemples

(<https://stackoverflow.com/questions/9127498/how-to-perform-a-real-time-search-and-filter-on-a-html-table>) qui fournissent une capacité de filtrage de base avec JavaScript, mais ceux-ci sont limités au filtrage sur une colonne. Finalement, je suis tombé sur la bibliothèque JavaScript TableFilter, qui contient une excellente

(<http://koalyptus.github.io/TableFilter/>) documentation

(<http://koalyptus.github.io/TableFilter/docs/>) et de nombreux exemples

(<http://koalyptus.github.io/TableFilter/examples.html>) .

Utilisation de TableFilter

Pour commencer, j'ai cloné TableFilter dans mon référentiel :

```
git clone https://github.com/koalyptus/TableFilter.git (https://github.com/koalyptus/TableFilter.git)
```

Nous allons éditer `index.html`, le fichier HTML qui contient notre table et l'utilise `tablefilter.js` pour effectuer le filtrage.

```
├─ assets
├─ index.html
├─ tablefilter
  └─ tablefilter.js
```

La stratégie que j'ai utilisée pour obtenir un produit fonctionnel est simple et applicable à des défis techniques plus généraux :

1. Commencez par un exemple de travail qui fait approximativement ce dont vous avez besoin ou plus
2. Supprimer ou modifier des fonctionnalités pour mieux s'aligner sur votre cas d'utilisation
3. Répétez l'étape 2 jusqu'à ce que vous ayez ce que vous voulez

La clé ici est de maintenir un exemple de travail tout au long et de prendre du recul si vous cassez quelque chose. En itérant plusieurs fois sur le `index.html` j'ai emprunté à l'un des exemples de base, il me restait ceci:

```
<!DOCTYPE HTML>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Table Filter Example</title>

  <!-- Bootstrap core CSS -->
  <link rel="stylesheet" href="assets/css/bootstrap.min.css">

<body>

  <!-- Header -->
  <section id="header">
    <header>
      <h2>Filter Table Example</h2>
    </header>

    <!-- TABLE CODE START -->
    <table id="demo" border="1" class="dataframe">
      <thead>
        <tr style="text-align: right;">
          <th>First</th>
          <th>Last</th>
          <th>Birthday</th>
          <th>Age</th>
          <th>Color</th>
          <th>Height</th>
          <th>Weight</th>
          <th>IQ</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>ZOE</td>
          <td>DAVIS</td>
          <td>2012-03-12</td>
          <td>6</td>
          <td>Yellow</td>
          <td>67</td>
          <td>108</td>
          <td>81</td>
        </tr>

        ...

      </tbody>
    </table>
    <!-- TABLE CODE END -->

  </section>
</body>
</head>

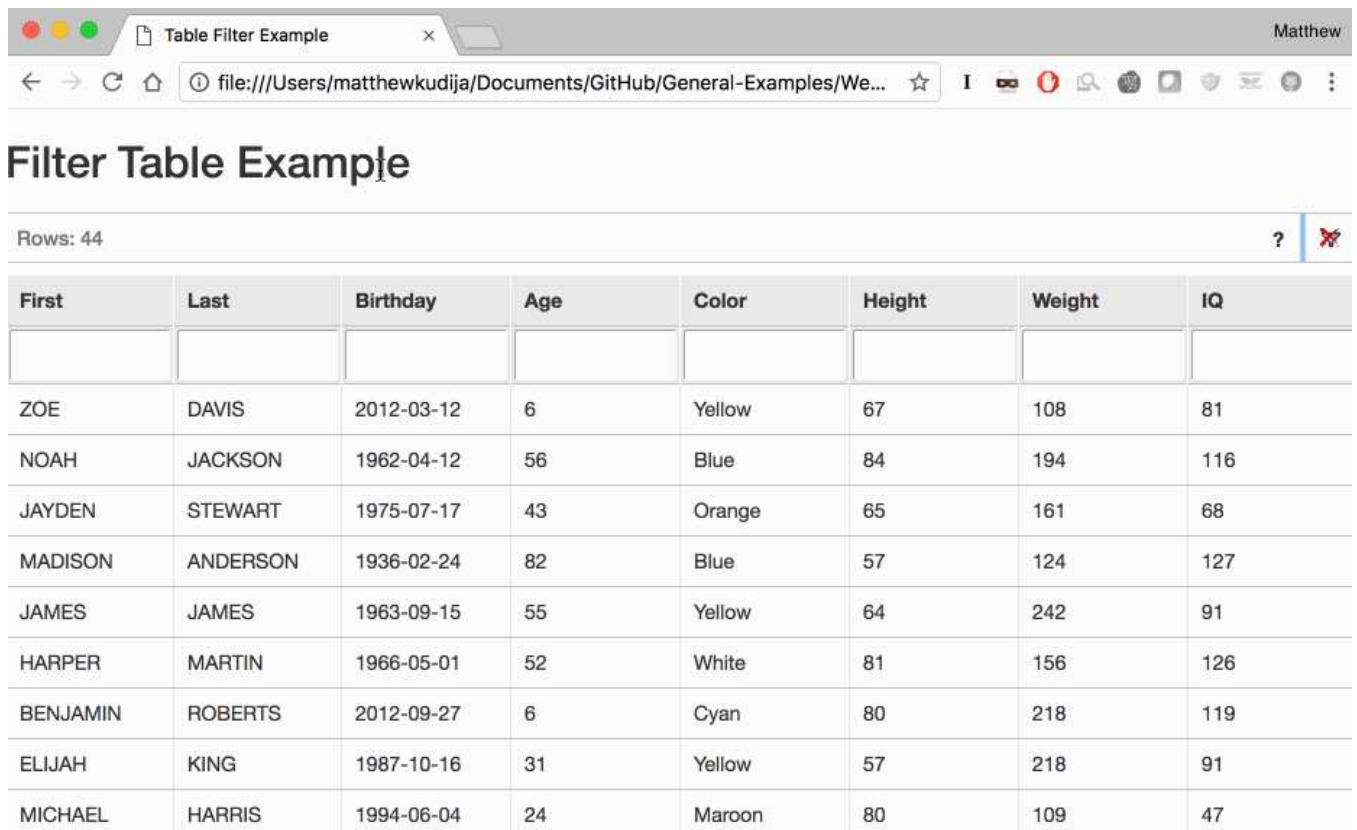
<script src="tablefilter/tablefilter.js"></script>
```

```
<script data-config="">
var filtersConfig = {
  base_path: 'tablefilter/',
  auto_filter: {
    delay: 110 //milliseconds
  },
  filters_row_index: 1,
  state: true,
  alternate_rows: true,
  rows_counter: true,
  btn_reset: true,
  status_bar: true,
  msg_filter: 'Filtering...'
};
var tf = new TableFilter('demo', filtersConfig);
tf.init();
</script>
```

Vous pouvez afficher le code (https://github.com/mkudija/General-Examples/tree/master/Web/HTML_Table_Filter) original ou un exemple en direct (http://matthewkudija.com/General-Examples/Web/HTML_Table_Filter/index.html) du produit final.

Personnalisez cela en incluant le code HTML définissant le tableau entre `<!-- TABLE CODE START -->` et `<!-- TABLE CODE END -->` (facile à automatiser avec Python si vous devez souvent mettre à jour les données). Juste en dessous de la définition de table, nous appelons le `tablefilter.js` script et définissons les éléments de configuration.

Cela nous laisse avec un tableau facile à filtrer :



First	Last	Birthday	Age	Color	Height	Weight	IQ
ZOE	DAVIS	2012-03-12	6	Yellow	67	108	81
NOAH	JACKSON	1962-04-12	56	Blue	84	194	116
JAYDEN	STEWART	1975-07-17	43	Orange	65	161	68
MADISON	ANDERSON	1936-02-24	82	Blue	57	124	127
JAMES	JAMES	1963-09-15	55	Yellow	64	242	91
HARPER	MARTIN	1966-05-01	52	White	81	156	126
BENJAMIN	ROBERTS	2012-09-27	6	Cyan	80	218	119
ELIJAH	KING	1987-10-16	31	Yellow	57	218	91
MICHAEL	HARRIS	1994-06-04	24	Maroon	80	109	47

Réflexions finales

Est-ce la meilleure façon de partager des données ? Certainement pas. Mais c'était une solution rapide à un problème que j'avais dans une situation (de travail) où la rapidité et la facilité sont valorisées. De plus, cette solution utilise l'infrastructure de pages GitHub existante et une excellente bibliothèque open source .

(<https://github.com/koalyptus/TableFilter/blob/master/LICENSE>)

J'espère que cet exemple vous fournira une solution rapide à un problème que j'ai déjà rencontré ou vous montrera comment vous pourrez peut-être pirater ensemble une solution en itérant sur un exemple de travail similaire que vous pouvez trouver en ligne.

- *Tous les noms de cet ensemble de données sont faux. Toute ressemblance avec des personnes réelles, vivantes ou décédées, est purement fortuite... J'ai inventé certaines données puisque les données motivant cet exemple sont propriétaires.*
- *Vous pouvez afficher le code (https://github.com/mkudija/General-Examples/tree/master/Web/HTML_Table_Filter) original ou un exemple en direct (http://matthewkudija.com/General-Examples/Web/HTML_Table_Filter/index.html) du produit final.*
- *En écrivant ceci, j'ai découvert Brython (<https://github.com/brython-dev/brython>) , une implémentation de Python 3 dans le navigateur. Cela vaut la peine d'être vérifié à l'avenir.*

html (../../../tag/html.html)

javascript (../../../tag/javascript.html)

la toile (../../../tag/web.html)

TableauFiltre (../../../tag/tablefilter.html)

adresse e-mail

S'abonner

ALSO ON MATTHEW KUDIJA

Visualizing An Infant's Schedule

4 years ago • 1 comment

A newborn may only eat, sleep, and poop, but they sure do it a lot! Here is a ...

Diff Two Excel Files with Python

4 years ago • 36 comments

It's handy to be able to quickly find the differences between two Excel files. ...

xlwings: Use Python and Excel to ...

5 years ago • 3 comments

The Python xlwings library provides easy interaction between Python and a ...

Diff Two Excel Files with Python

4 years ago • 36 comments

It's handy to be able to quickly find the differences between two Excel files. ...

3 Comments

 Login ▾

Join the discussion...

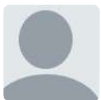
LOG IN WITH

OR SIGN UP WITH DISQUS 

Sort by Best ▾

**Unnikrishnan R** • 2 years ago

great post. This really helped me

 |  • Reply • Share ›**vijay sai** • 4 years ago

Great post Matthew. This really helped. I've a question though, I've some html content inside my html table that I return from Pandas DataFrame. With this approach, the html content is also displayed as simple text. Did you happen to come across any setting in the library that will help enable the html ?

 |  • Reply • Share ›**Matthew Kudija** Mod ➔ vijay sai • 3 years ago

I don't know how to do this right off, but let me know if you figure it out!

 |  • Reply • Share › [Subscribe](#)  [Privacy](#)  [Do Not Sell My Data](#)© 2005 -2022 Matthieu Kudija (<https://matthewkudija.com>) | La source (<https://github.com/mkudija/blog>)