

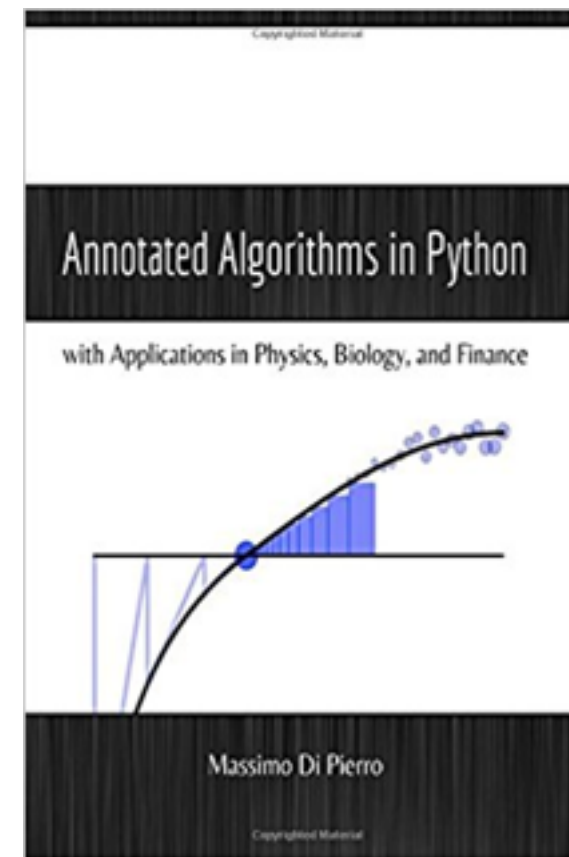
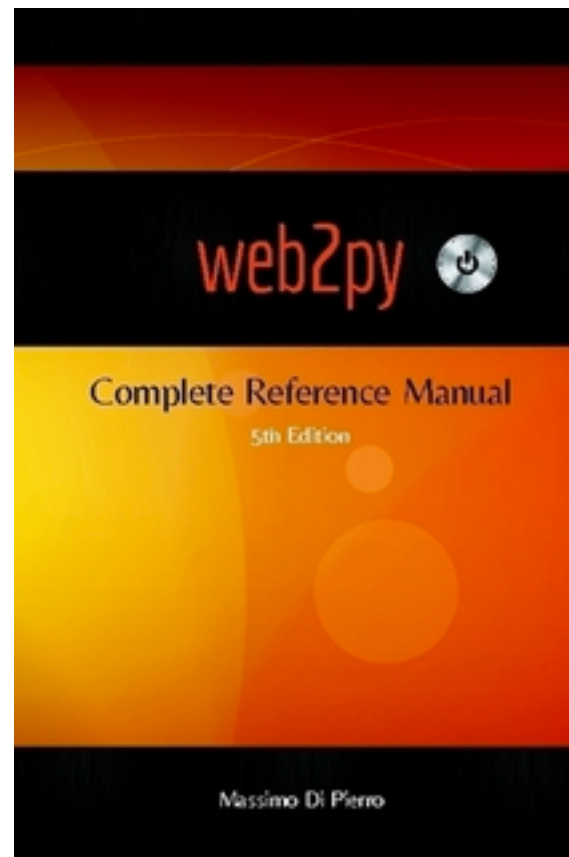


# **Building a Scientific Platform with Python**

Massimo Di Pierro

(DePaul University, Chicago)

# about me



# Python for Science

Numpy (efficient arrays)

```
numpy.random.random((2,3)).reshape(6).sum()
```

Pandas (tabular data)

```
df[df['colA']>0]['colB'].sum()
```

Scipy (algorithms)

```
linalg.inv(numpy.array([[1,2],[3,4]]))
```

ScikitLearn (Machine Learning fit/predict/cluster)

```
sklearn.svc.SVC().fit(data, target).predict(new_data)
```

Keras / Tensorflow (Naural Network fit/predict)

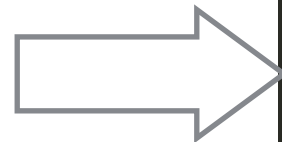
```
model = Sequential()  
model.add(RNN(128, return_sequences=True))  
...  
model.compile(...)  
model.fit(data, target)  
model.predict(new_data)
```

# How to share?

## Python Modules



input

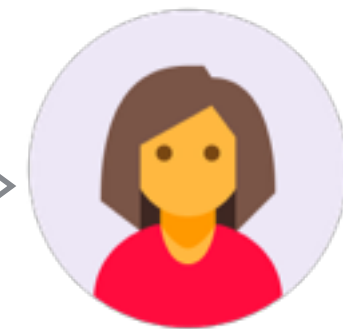
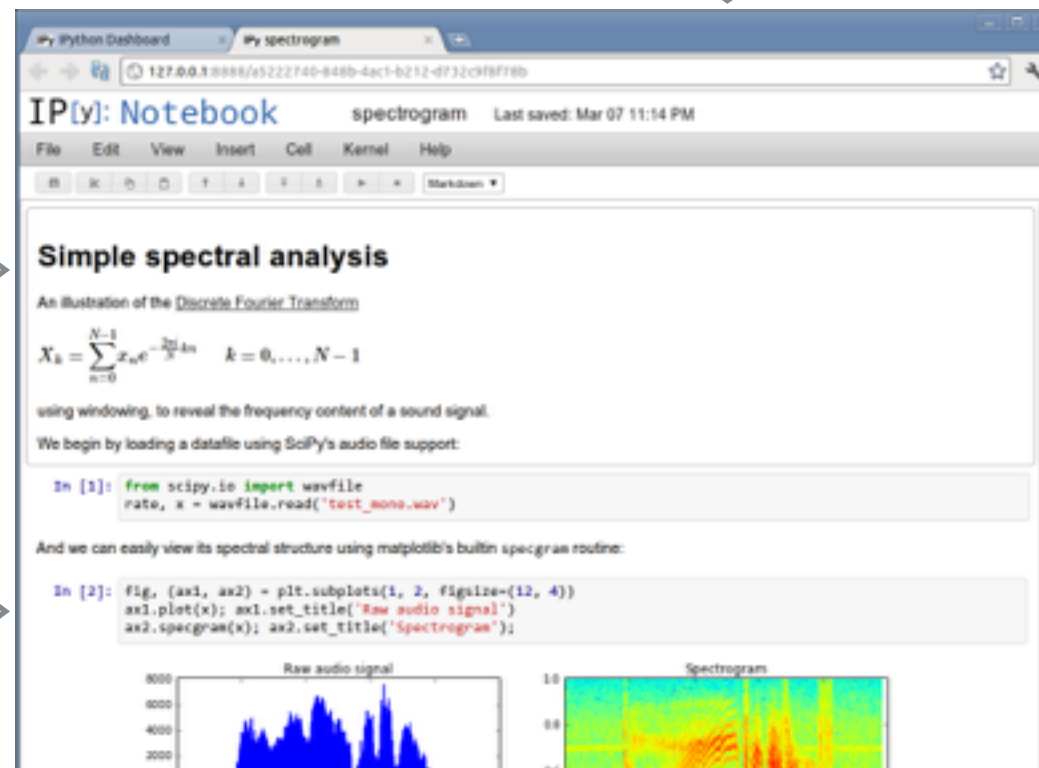
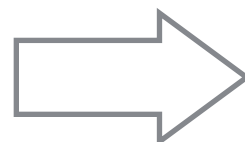
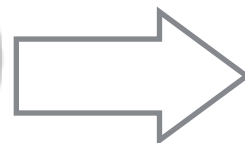
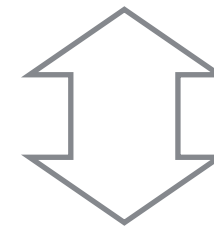


```
for i in people.data.users:
    response = client.api.statuses.user_timeline.get(screen_name=i.screen_name)
    print 'Got', len(response.data), 'tweets from', i.screen_name
    if len(response.data) != 0:
        ldate = response.data[0]['created_at']
        ldate2 = datetime.strptime(ldate, '%a %b %d %H:%M:%S +0000 %Y')
        today = datetime.now()
        howlong = (today - ldate2).days
        if howlong < daywindow:
            print i.screen_name, 'has tweeted in the past', daywindow,
            totaltweets += len(response.data)
            for j in response.data:
                if j.entities.urls:
                    for k in j.entities.urls:
                        newurl = k['expanded_url']
                        urlset.add((newurl, j.user.screen_name))
            else:
                print i.screen_name, 'has not tweeted in the past', daywindow
```

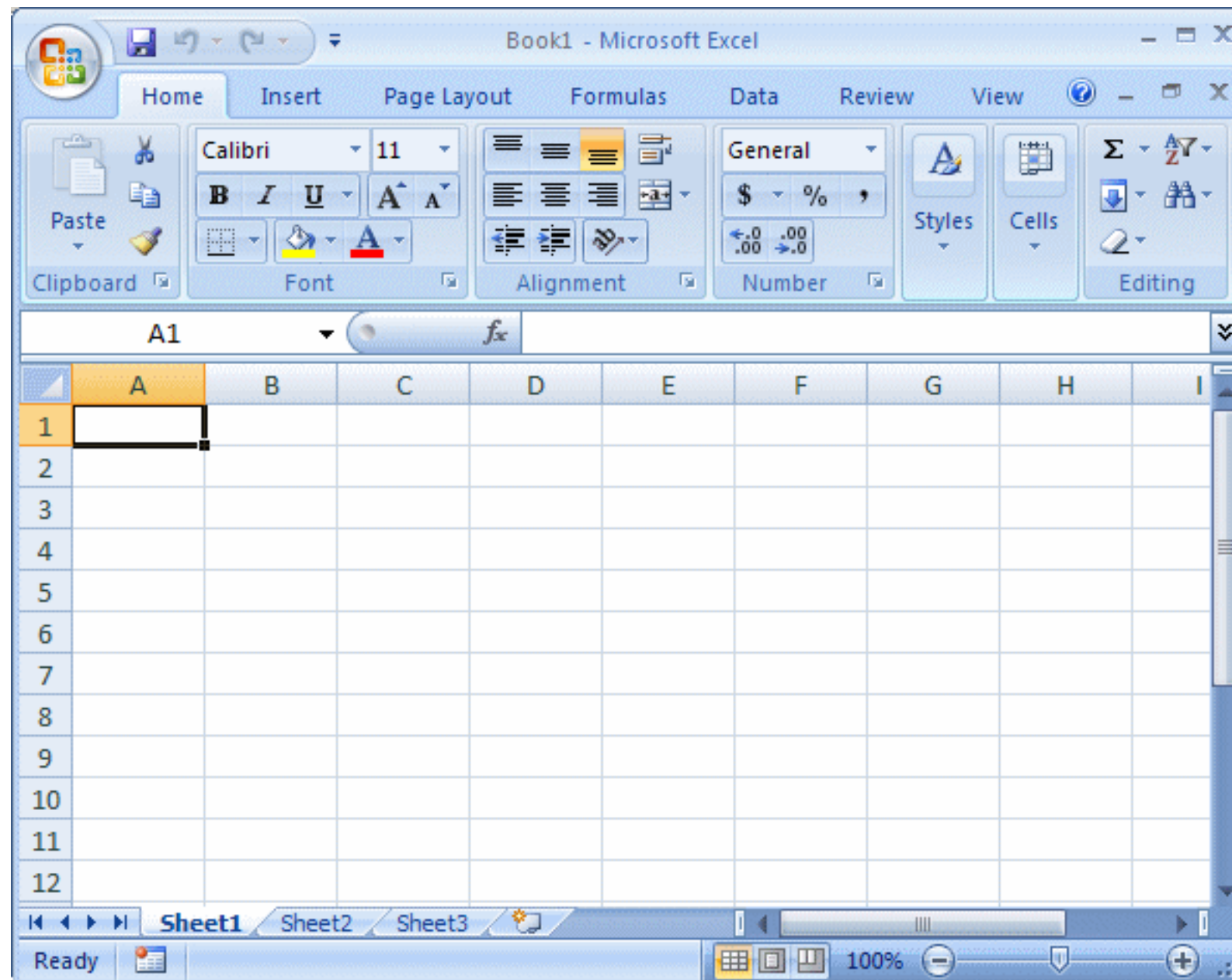
output



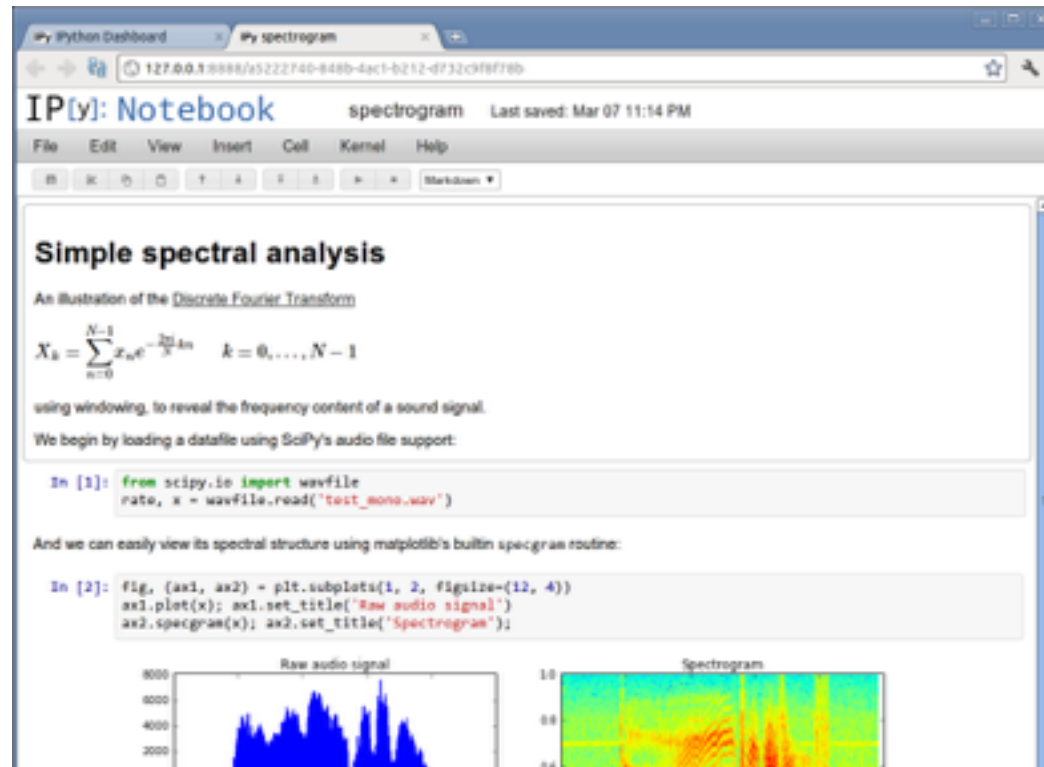
## iPython/Jupyter Notebook



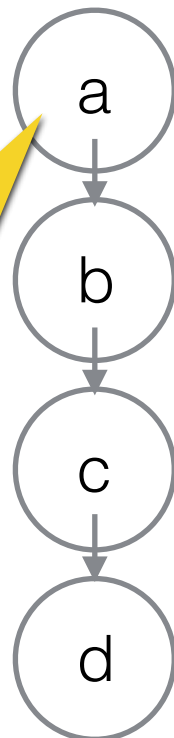
# Excel (really?)



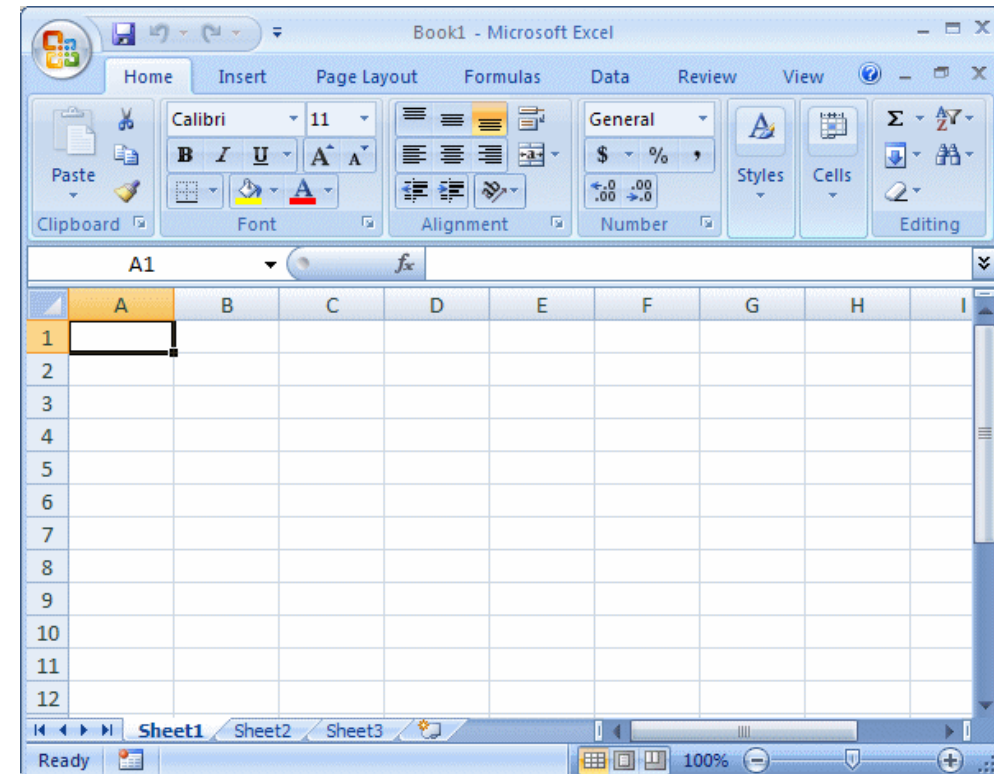
# ipython



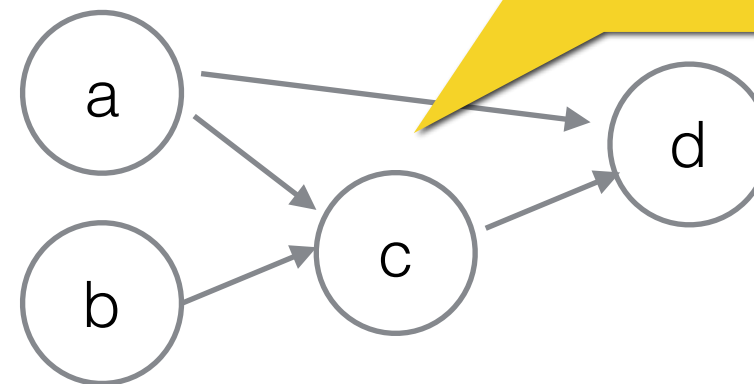
linear  
manual  
dependencies



# excel



graph  
automatic  
dependency



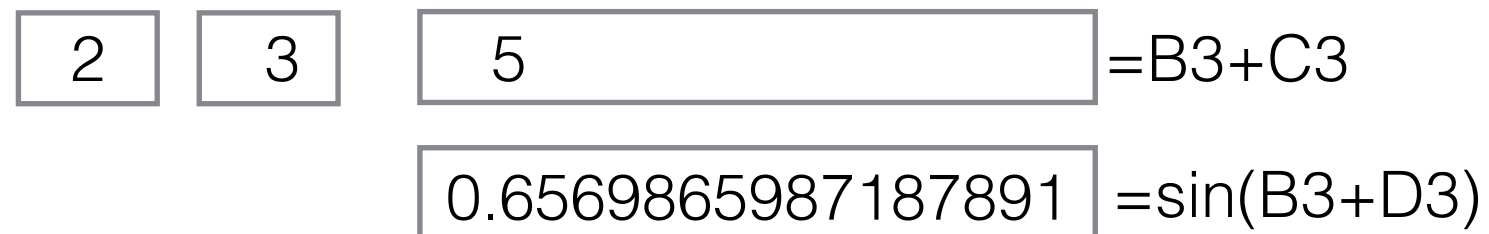
# Excel (really?)

	A	B	C	D	E	F	G	H	I
1									
2									
3		2	3	=B3+C3					
4									
5				=sin(B3+D3)					
6									
7									
8									
9									
10									
11									
12									

values = {'B3': 2, 'C3': 3}

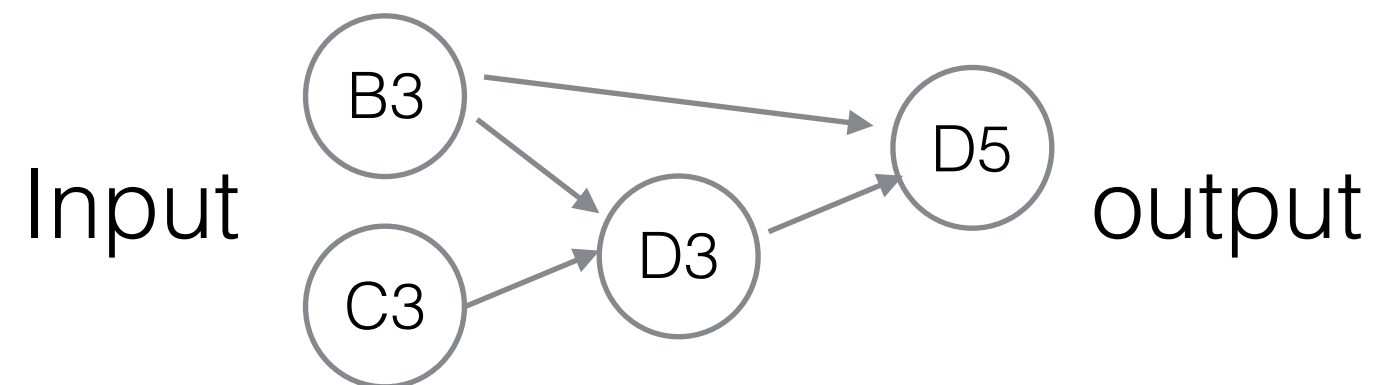
formulas = {'D3': '=B3+C3', 'D5': '=sin(B3+D3)'}

# Excel (really?)



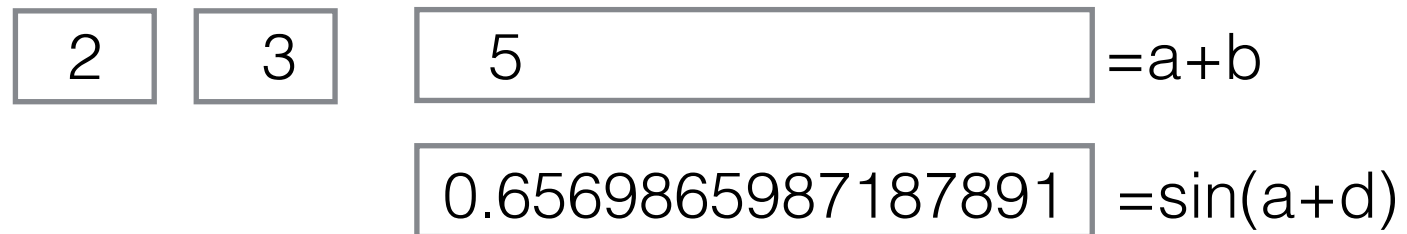
values = {'B3': 2, 'C3': 3}

formulas = {'D3': 'B3+C3', 'D5': 'sin(B3+D3)'}



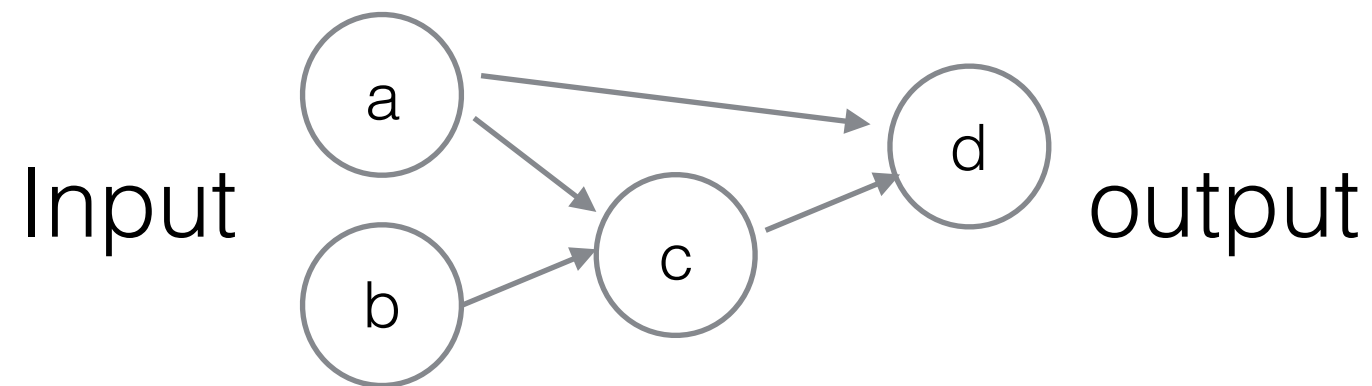


# Excel (really?)



values = {'a': 2, 'b': 3}

formulas = {'c': 'a+b', 'd': 'sin(a+d)'}



# Idea (1/2)



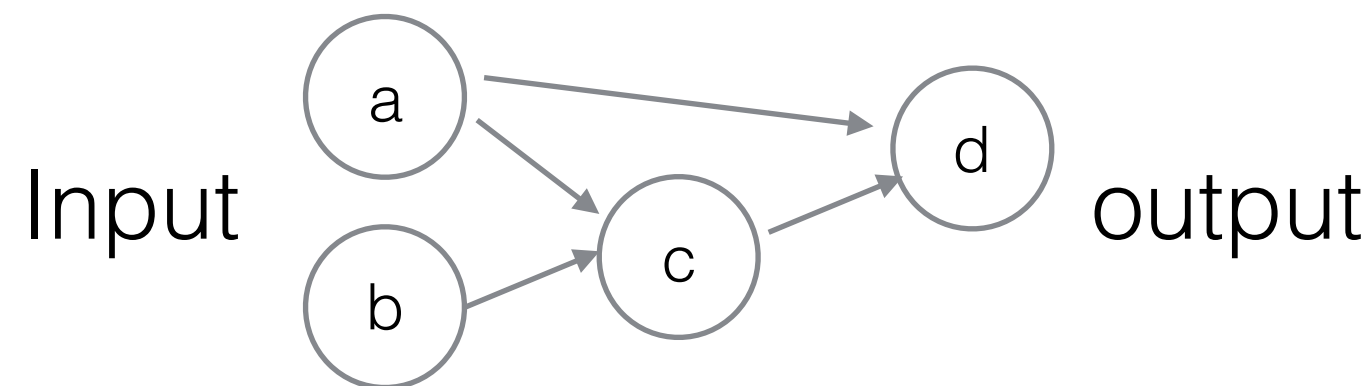
output fields  
(computed)

input fields  
(validated)

values = {'a': 2, 'b': 3}

formulas = {'c': 'a+b', 'd': 'sin(a+d)'}

any python code



# Idea (2/2) - CMS



use wiki syntax to describe pages



use wiki syntax to describe forms

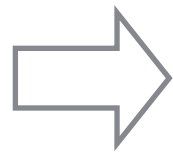


embed code in pages

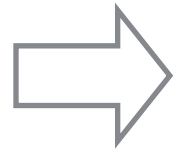


generate interactive pages

# Example



create new page



edit

```
# Title  
...  
[[a]] [[b]] [[c=a+b]] [[d=sin(a+d)]]
```

```
from math import sin
```

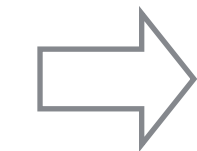
author can  
change formulas  
and code



publish

<http://>

```
Title  
...  
[2] [3] 5 0.6569865987187891
```



visitors

<http://>

```
Title  
...  
[1] [7] 8 0.4121184852417566
```

visitors can only  
change input

input fields  
(validated)

output fields  
(computed)



# Example

client

server

http://

**Title**

...  
[2] [3] 5 0.6569865987187891

http://

**Othe Page**

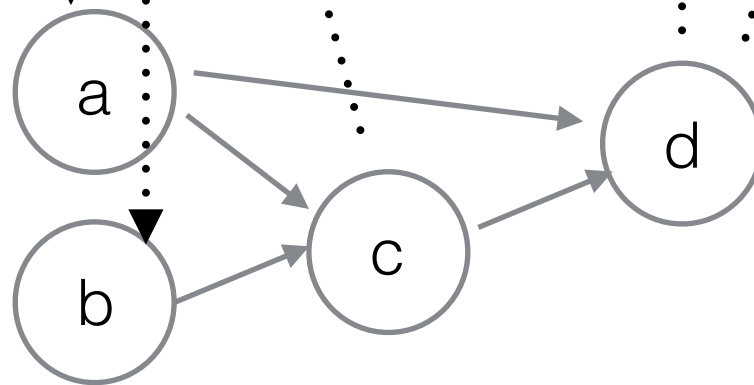
...  
0.6569865987187891

web sockets  
async

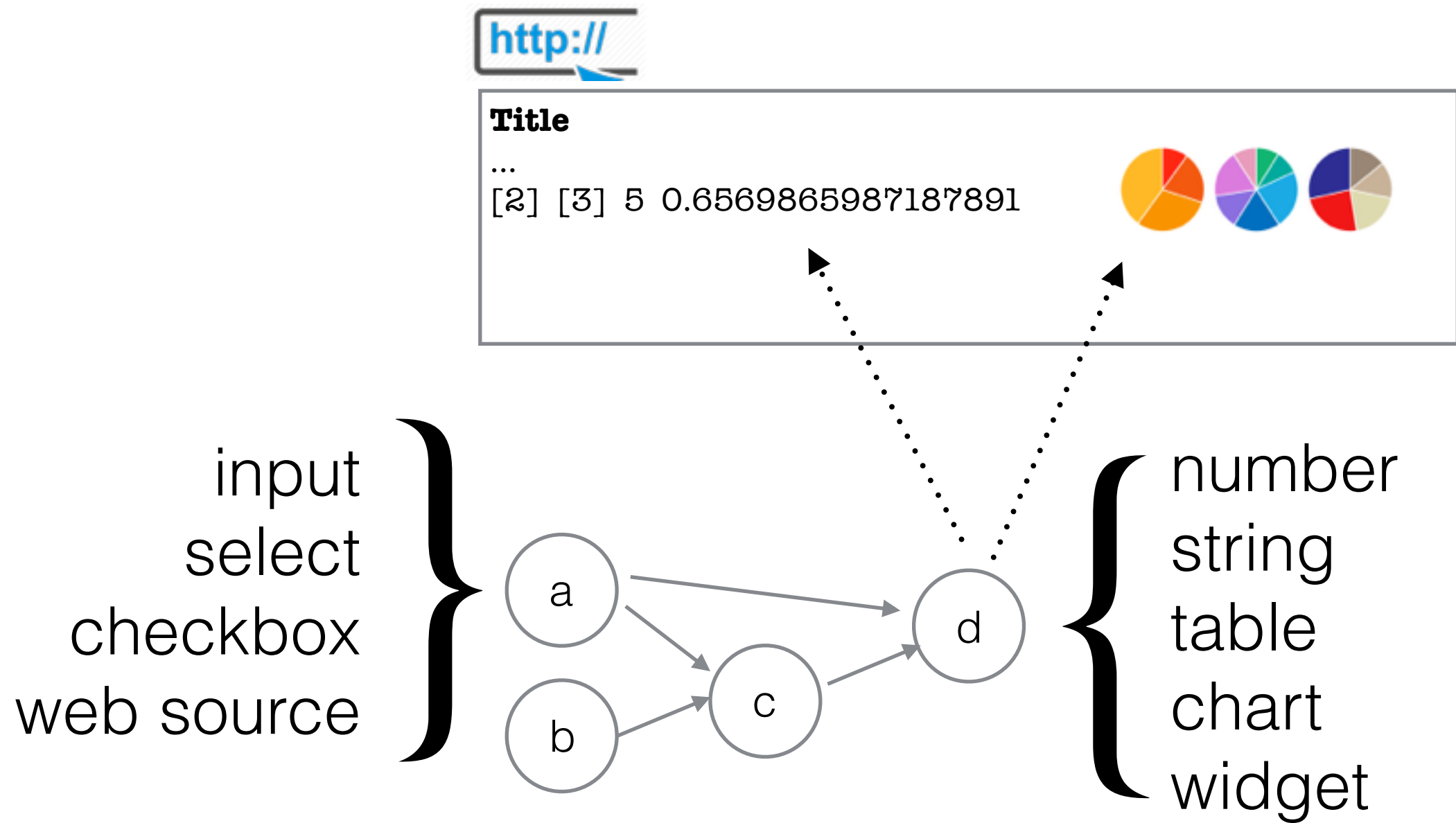
storage

load

save



# Example



# Choice of Technologies

- server: bottle.py
- server: gevent websockets
- client: jquery
- client: vue.js
- client: markdown (modified)