

Jupyter Notebook Driven Development with pynt

pynt: Software Engineering with Jupyter Notebooks

Who am I?

- Edward Banner
- Data Platform Software Engineer at Stitch Fix
- San Francisco, CA



Talk outline

- Introduction and motivation
- Introduction to Emacs IPython Notebook
- Feature tour
- Remarks and observations
- The future

What is pynt? 🍺

- “Python Interactive”
- It gets your code into a jupyter notebook.

The image shows two screenshots illustrating the pynt process. The top screenshot is a terminal window titled 'copy_paste.py' containing the following Python code:

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```

The bottom screenshot shows a Jupyter Notebook interface titled 'jupyter Image Size'. It has a toolbar with File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and a dashboard view. A red arrow points from the terminal window to the Jupyter interface. Two large red question marks are overlaid on the Jupyter interface area.

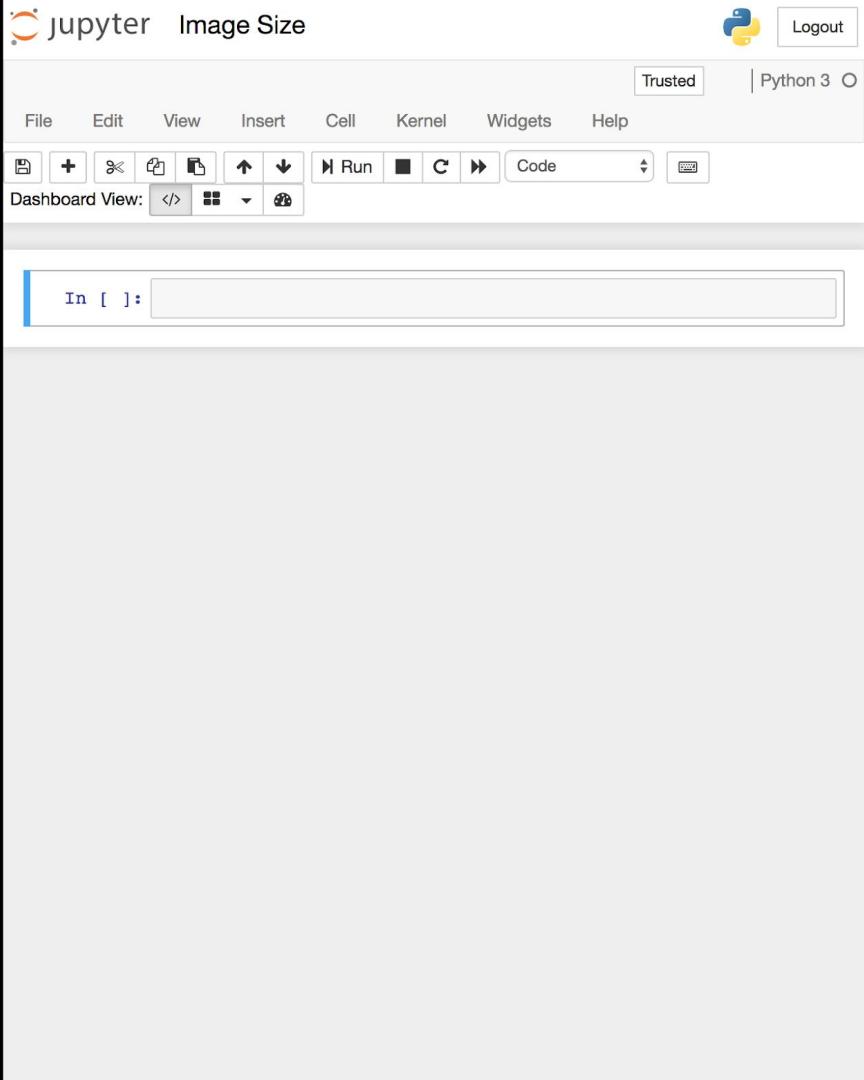


Why pynt?

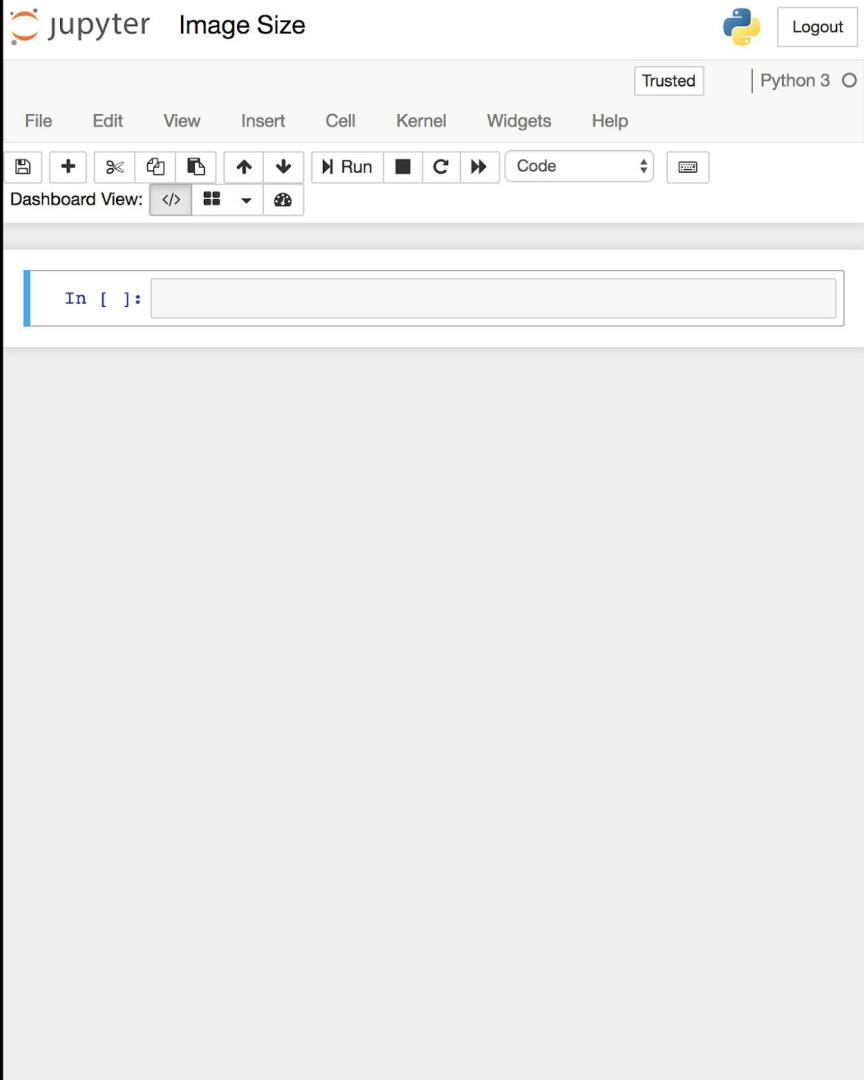
- I ❤️ jupyter notebooks.
- Decrease notebook FOMO. *
- **Already using notebooks to develop software.**



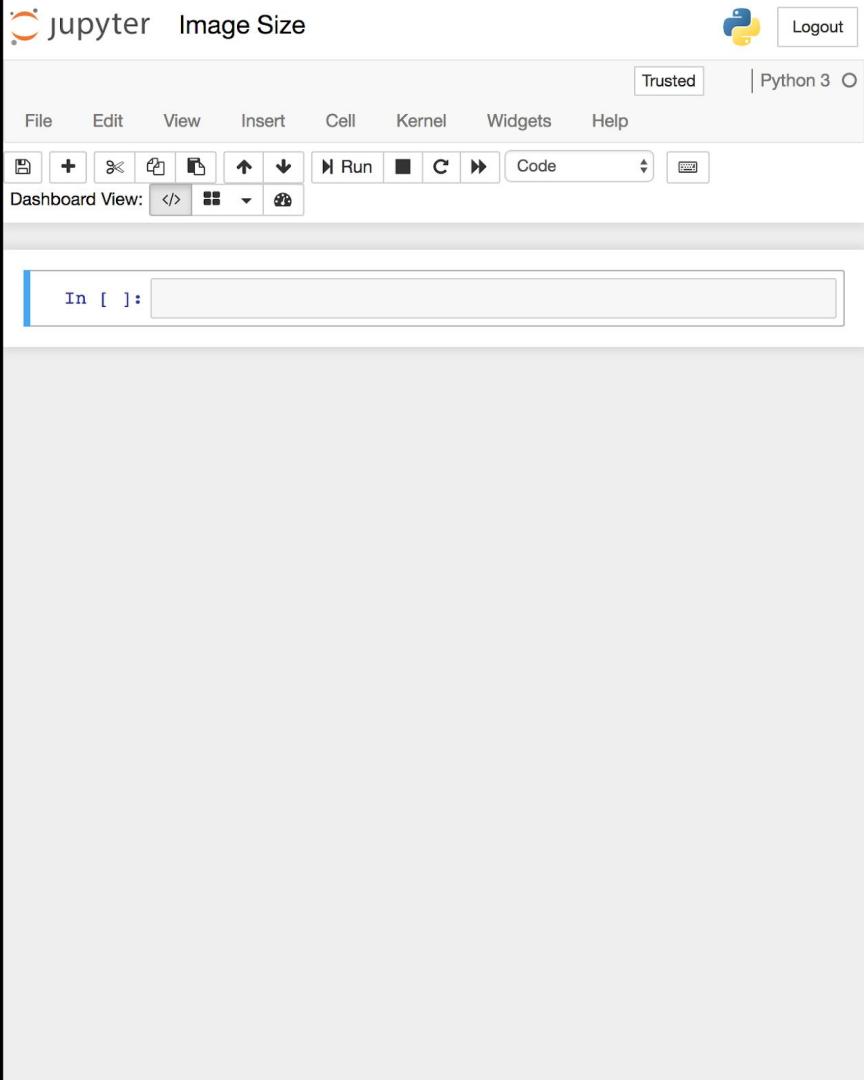
```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```



```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    resized_img = img.resize(size=[128, 128])
11    return resized_img.size
```

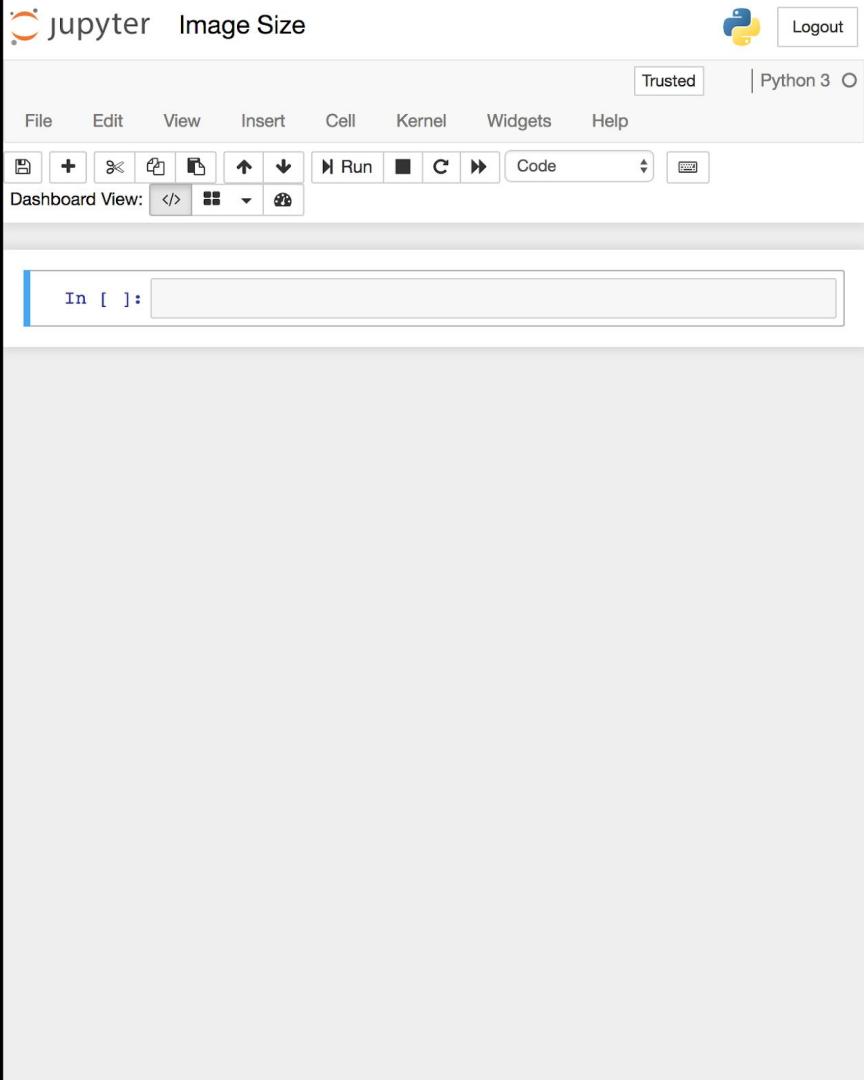


```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```



```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~



1 - 186 copy_paste.py unix | 10: 4 All
Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```

jupyter Image Size

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Dashboard View: </>

```
In [ ]: data = requests.get(url, stream=True).raw
        img = PIL.Image.open(data)
        return img.size
```

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```

jupyter Image Size

Logout Trusted | Python 3 ○

File Edit View Insert Cell Kernel Widgets Help

Make the Request

```
In [ ]: data = requests.get(url, stream=True).raw  
data
```

Convert Bytes to Image

```
In [ ]: img = PIL.Image.open(data)
        img
```

Get Image Size

```
In [ ]: img.size
```

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```

jupyter Image Size

Logout Trusted | Python 3 ○

File Edit View Insert Cell Kernel Widgets Help

Make the Request

```
In [ ]: import requests
```

```
data = requests.get(url, stream=True).raw  
data
```

Convert Bytes to Image

In []: `import PIL`

```
img = PIL.Image.open(data)  
img
```

Get Image Size

In []: img.size

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```

jupyter Image Size

Logout Trusted Python 3

File Edit View Insert Cell Kernel Widgets Help

Example URL

```
In [ ]: url = 'https://bit.ly/2zp7YxL'
```

Make the Request

```
In [ ]: import requests

        data = requests.get(url, stream=True).raw
        data
```

Convert Bytes to Image

```
In [ ]: import PIL  
  
        img = PIL.Image.open(data)  
        img
```

Get Image Size

```
In [1]: img.size
```

1 - 186 **copy_paste.py** unix | 1: 0 All
Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```

① – 186 copy_paste.py

unix | 1: 0 All

 Jupyter Image Size



[Logout](#)

Trusted

Python 3 ○

File Edit View Insert Cell Kernel Widgets Help



Dashboard View: </>   

Example URL

```
In [4]: url = 'https://bit.ly/2zp7YxL'
```

Make the Request

```
In [5]: import requests

        data = requests.get(url, stream=True).raw
        data
```



```
Out[5]: <urllib3.response.HTTPResponse at 0x109c92f28>
```

Convert Bytes to Image

```
In [6]: import PIL  
        img = PIL.Image.open(data)  
        img
```

Out[6]:



```
1 import PIL  
2 import requests  
3  
4  
5 def image_size(url):  
6     """Compute image size at `url`"""  
7  
8     data = requests.get(url, stream=True).raw  
9     img = PIL.Image.open(data)  
10    return img.size
```

jupyter Image Size

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Run C Code

Dashboard View: </>

Convert Bytes to Image

In [6]:

```
import PIL  
  
img = PIL.Image.open(data)  
img
```

Out[6]:



Resize Image

In []:

```
img.resize(size=[128, 128])
```

Get Image Size

In [7]:

```
img.size
```

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```

jupyter Image Size

Logout Trusted Python 3

File Edit View Insert Cell Kernel Widgets Help



The screenshot shows a software interface with a toolbar at the top. The toolbar includes icons for file operations (New, Open, Save, Print, Find, Copy, Paste), navigation (Up, Down, Left, Right), a Run button, a Code dropdown menu, and a dashboard view icon.

```
In [11]: import PIL  
  
        img = PIL.Image.open(data)  
        img
```



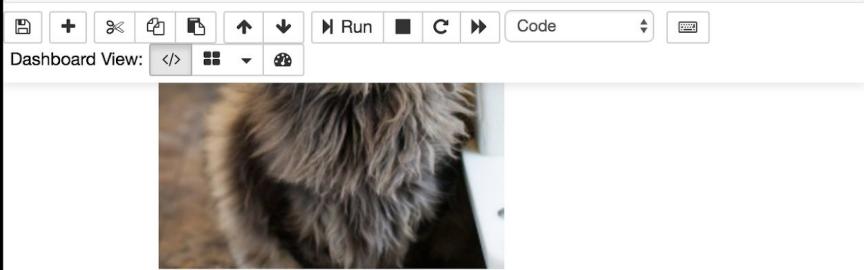
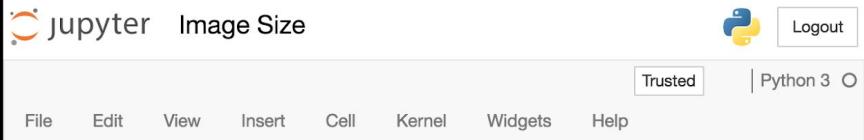
Resize Image

```
In [12]: resized_img = img.resize(size=[128, 128])
resized_img
```

Out[12]:



```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```



Resize Image

```
In [13]: resized_img = img.resize(size=[128, 128])  
resized_img
```

Out[13]:

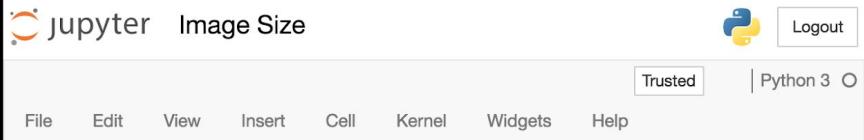


Get Image Size

```
In [14]: resized_img.size
```

Out[14]: (128, 128)

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```



Dashboard View: </>



Resize Image

```
In [13]: resized_img = img.resize(size=[128, 128])  
resized_img
```

Out[13]:



Get Image Size

```
In [14]: resized_img.size
```

Out[14]: (128, 128)

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```

The figure shows a Jupyter Notebook interface with the following components:

- Header:** "jupyter Image Size" on the left, "Trusted" and "Logout" buttons on the right.
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help buttons.
- Dashboard View:** Buttons for file operations like Open, Save, and Run, along with a "Code" dropdown and a terminal icon.
- In [11]:** Code cell containing:

```
import PIL

img = PIL.Image.open(data)
img
```
- Out[11]:** Output cell showing a large, fluffy cat with yellow eyes.
- In []:** Code cell containing:

```
# Resize Image

resized_img = img.resize(size=[128, 128])
resized_img

# Get Image Size

resized_img.size
```

① * 186 **copy_paste.py** unix | 1: 0 All
Auto-saving...done

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```

jupyter Image Size

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Dashboard View: </>

Convert Bytes to Image

In [11]:

```
import PIL
img = PIL.Image.open(data)
img
```

Out[11]:



In [15]:

```
resized_img = img.resize(size=[128, 128])
resized_img.size
```

Out[15]:

```
(128, 128)
```

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```

jupyter Image Size

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Run C Code

Dashboard View: </>

Convert Bytes to Image

In [11]:

```
import PIL
img = PIL.Image.open(data)
img
```

Out[11]:



In [15]:

```
resized_img = img.resize(size=[128, 128])
resized_img.size
```

Out[15]:

```
(128, 128)
```

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```

jupyter Image Size

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Run C Code

Dashboard View: </>

Convert Bytes to Image

In [11]:

```
import PIL
img = PIL.Image.open(data)
img
```

Out[11]:



In [15]:

```
resized_img = img.resize(size=[128, 128])
resized_img.size
```

Out[15]:

```
(128, 128)
```

1 * 186 copy_paste.py unix | 10: 0 All
-- VISUAL LINE --

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    resized_img = img.resize(size=[128, 128])
11    return resized_img.size
```

jupyter Image Size

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Run C ► Code

Dashboard View: </> ■ □

Convert Bytes to Image

In [11]:

```
import PIL
img = PIL.Image.open(data)
img
```

Out[11]:



In [15]:

```
resized_img = img.resize(size=[128, 128])
resized_img.size
```

Out[15]:

```
(128, 128)
```

What did we just do?

1. Select a region of code.
 2. Paste it into the notebook.
 3. Split up each line into its own code cell.
 4. ???
 5. Paste back notebook code.

The screenshot shows a Jupyter Notebook interface with a Python script in the code cell. The code imports PIL and requests, defines a function to calculate image size from a URL, and then prints a large number of underscores (~) followed by a copy/paste message.

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at 'url'"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
11
12 ~
13 ~
14 ~
15 ~
16 ~
17 ~
18 ~
19 ~
20 ~
21 ~
22 ~
23 ~
24 ~
25 ~
26 ~
27 ~
28 ~
29 ~
30 ~
31 ~
32 ~
33 ~
34 ~
35 ~
36 ~
37 ~
38 ~
39 ~
40 ~
41 ~
42 ~
43 ~
44 ~
45 ~
46 ~
47 ~
48 ~
49 ~
50 ~
51 ~
52 ~
53 ~
54 ~
55 ~
56 ~
57 ~
58 ~
59 ~
60 ~
61 ~
62 ~
63 ~
64 ~
65 ~
66 ~
67 ~
68 ~
69 ~
70 ~
71 ~
72 ~
73 ~
74 ~
75 ~
76 ~
77 ~
78 ~
79 ~
80 ~
81 ~
82 ~
83 ~
84 ~
85 ~
86 ~
87 ~
88 ~
89 ~
90 ~
91 ~
92 ~
93 ~
94 ~
95 ~
96 ~
97 ~
98 ~
99 ~
100 ~
101 ~
102 ~
103 ~
104 ~
105 ~
106 ~
107 ~
108 ~
109 ~
110 ~
111 ~
112 ~
113 ~
114 ~
115 ~
116 ~
117 ~
118 ~
119 ~
120 ~
121 ~
122 ~
123 ~
124 ~
125 ~
126 ~
127 ~
128 ~
129 ~
130 ~
131 ~
132 ~
133 ~
134 ~
135 ~
136 ~
137 ~
138 ~
139 ~
140 ~
141 ~
142 ~
143 ~
144 ~
145 ~
146 ~
147 ~
148 ~
149 ~
150 ~
151 ~
152 ~
153 ~
154 ~
155 ~
156 ~
157 ~
158 ~
159 ~
160 ~
161 ~
162 ~
163 ~
164 ~
165 ~
166 ~
167 ~
168 ~
169 ~
170 ~
171 ~
172 ~
173 ~
174 ~
175 ~
176 ~
177 ~
178 ~
179 ~
180 ~
181 ~
182 ~
183 ~
184 ~
185 ~
186 copy_paste.py
```

The screenshot shows a Jupyter Notebook interface with two main panes. The left pane contains Python code for calculating image dimensions from URLs using the PIL library. The right pane shows the results of running this code, including the URL used, the request headers, the raw response data, and the resulting image size.

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at 'url'"""
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10    return img.size
```

Example URL

```
In [1]: url = "https://bit.ly/2zg7YXG"
```

Make the Request

```
In [2]: import requests
data = requests.get(url, stream=True).raw
data
```

Convert Bytes to Image

```
In [3]: import PIL
img = PIL.Image.open(data)
img
```

Get Image Size

```
In [4]: img.size
```

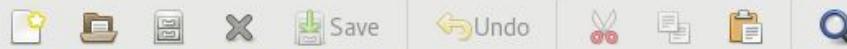
0 - 186 copy_paste.py unix | 1: 0 All
Server error: TypeError('__init__() got an unexpected keyword argument 'environment''')

Emacs IPython (Jupyter) Notebook

Emacs + Jupyter Notebook



File Edit Options Buffers Tools Emacs-Lisp Help



```
(defsubst hash-table-empty-p (hash-table)
  "Check whether HASH-TABLE is empty (has 0 elements)."
  (zerop (hash-table-count hash-table)))

(defsubst hash-table-keys (hash-table)
  "Return a list of keys in HASH-TABLE."
  (let ((keys '()))
    (maphash (lambda (k _v) (push k keys)) hash-table)
    keys))

(defsubst hash-table-values (hash-table)
  "Return a list of values in HASH-TABLE."
  (let ((values '()))
    -:--- subr-x.el.gz 36% L148 (Emacs-Lisp)
Next: Distrib, Up: \(dir\)
\(emacs\)Top
```

The Emacs Editor

Emacs is the extensible, customizable, self-documenting real-time display editor. This manual describes how to edit with Emacs and some of the ways to customize it; it corresponds to GNU Emacs version 26.0.50.

If you are reading this in Emacs, type 'h' to read a basic introduction to the Info documentation system.

U:%%- *info* (emacs) [Top](#) Top L9 (Info Narrow)

Emacs IPython (Jupyter) Notebook (EIN)

IP[3]: /1\ /2\ [+]\ |python3

Define the URL

In [1]
url = 'https://bit.ly/2zp7YxL'

Download the Image

In [2]
import PIL
import requests

data = requests.get(url, stream=True).raw
img = PIL.Image.open(data)

img

Out [2]:



Get the Size

In [3]
img.size
Out [3]:
(256, 256)



Trusted

Python 3

File Edit View Insert Cell Kernel Widgets Help



Dashboard View: </>

Define the URL

In [5]: `url = 'https://bit.ly/2zp7YxL'`

Download the Image

In [6]: `import PIL
import requests`

```
data = requests.get(url, stream=True).raw  
img = PIL.Image.open(data)
```

`img`

Out[6]:



Get the Size

IP[4]: /1\ /2\ [+ python3]

In []

Define the URL

In [2]: `url = 'https://bit.ly/2zp7YxL'`

Download the Image

In [3]: `import PIL
import requests`

```
data = requests.get(url, stream=True).raw  
img = PIL.Image.open(data)
```

`img`

Out [3]:



Get the Size

Pynt Feature Tour

jupyter notebook management

```
1 # Define the URL  
2  
3 url = 'https://bit.ly/2zp7YxL'  
4  
5 # Download the Image  
6  
7 import PIL  
8 import requests  
9  
10 data = requests.get(url, stream=True).raw  
11 img = PIL.Image.open(data)  
12  
13 # Get the Size  
14  
15 img.size
```

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~

```
1 # Define the URL
```

```
2  
3 url = 'https://bit.ly/2zp7YxL'  
4  
5 # Download the Image  
6  
7 import PIL  
8 import requests  
9  
10 data = requests.get(url, stream=True).raw  
11 img = PIL.Image.open(data)  
12  
13 # Get the Size  
14  
15 img.size
```

```
IP[1]: /1\ /2\ [+]|python3|
```

```
In [1]
```

```
pwd
```

```
Out [1]:
```

```
'/Users/ebanner/Documents/demo'
```

code dumping

1 # Define the URL

IP[2]: /1\ /2\ [+] |python3|

```
1 # Define the URL
2
3 url = 'https://bit.ly/2zp7YxL'
4
5 # Download the Image
6
7 import PIL
8 import requests
9
10 data = requests.get(url, stream=True).raw
11 img = PIL.Image.open(data)
12
13 # Get the Size
14
15 img.size
```

In []

① - 195 demo050.py

unix | 1: 0 All ②

Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)

All

```
1 # Define the URL
2
3 url = 'https://bit.ly/2zp7YxL'
4
5 # Download the Image
6
7 import PIL
8 import requests
9
10 data = requests.get(url, stream=True).raw
11 img = PIL.Image.open(data)
12
13 # Get the Size
14
15 img.size
~
```

IP[2]: /1\ /2\ [+] |python3|

~

h1:

demo050

In []:

```
url = 'https://bit.ly/2zp7YxL'  
url
```

In []:

```
import PIL
```

In []:

```
import requests
```

In []:

```
data = requests.get(url, stream=True).raw  
data
```

In []:

```
img = PIL.Image.open(data)  
img
```

In []:

```
img.size
```

~

~

```
1 # Define the URL
2
3 url = 'https://bit.ly/2zp7YxL'
4
5 # Download the Image
6
7 import PIL
8 import requests
9
10 data = requests.get(url, stream=True).raw
11 img = PIL.Image.open(data)
12
13 # Get the Size
14
15 img.size
~
```

IP[2]: /1\ /2\ [+]|python3|

h1:
`demo050`

In [*]\$
url = 'https://bit.ly/2zp7YxL'
url

In [*]\$
import PIL

In [*]\$
import requests

In [*]\$
data = requests.get(url, stream=True).raw
data

In [*]\$
img = PIL.Image.open(data)
img

In [*]\$
img.size

```
1 # Define the URL
2
3 url = 'https://bit.ly/2zp7YxL'
4
5 # Download the Image
6
7 import PIL
8 import requests
9
10 data = requests.get(url, stream=True).raw
11 img = PIL.Image.open(data)
12
13 # Get the Size
14
15 img.size
~
```

IP[8]: /1\ /2\ [+] |python3|

h1:
`demo050`

In [3]⚡
url = 'https://bit.ly/2zp7YxL'
url
Out [3]:
'https://bit.ly/2zp7YxL'

In [4]⚡
import PIL

In [5]⚡
import requests

In [6]⚡
data = requests.get(url, stream=True).raw
data
Out [6]:
<requests.packages.urllib3.response.HTTPResponse at >

In [7]⚡
img = PIL.Image.open(data)
img
Out [7]:



```
1 # Define the URL
2
3 url = 'https://bit.ly/2zp7YxL'
4
5 # Download the Image
6
7 import PIL
8 import requests
9
10 data = requests.get(url, stream=True).raw
11 img = PIL.Image.open(data)
12
13 # Get the Size
14
15 img.size
```

IP[8]: /1\ /2\ [+] |python3|
import requests

In [6] ↗
data = requests.get(url, stream=True).raw
data

Out [6]:
<requests.packages.urllib3.response.HTTPResponse at >

In [7] ↗
img = PIL.Image.open(data)
img

Out [7]:



In [8] ↗
img.size
Out [8]:
(256, 256)

code regions


```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     # Define the URL
7
8     url = 'https://bit.ly/2zp7YxL'
9
10    # Download the Image
11
12    data = requests.get(url, stream=True).raw
13    img = PIL.Image.open(data)
14
15    # Get the Size
16
17    img.size
```

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~

1

- 245 code_regions.py

Python

@pynt®®h®

unix | 1: 0 | All

Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     # Define the URL
7
8     url = 'https://bit.ly/2zp7YxL'
9
10    # Download the Image
11
12    data = requests.get(url, stream=True).raw
13    img = PIL.Image.open(data)
14
15    # Get the Size
16
17    img.size
```

1 – 245 code_regions.py

unix | 1: 0 All

2

ein: [info] Opened notebook list at <http://localhost:8889> with path Documents/demo.

All

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     # Define the URL
7
8     url = 'https://bit.ly/2zp7YxL'
9
10    # Download the Image
11
12    data = requests.get(url, stream=True).raw
13    img = PIL.Image.open(data)
14
15    # Get the Size
16
17    img.size
```

1

- 245 code_regions.py

Python

@@pynt@ein:c\$@h@K

unix | 5: 0 All

SPC w m

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     # Define the URL
7
8     url = 'https://bit.ly/2zp7YxL'
9
10    # Download the Image
11
12    data = requests.get(url, stream=True).raw
13    img = PIL.Image.open(data)
14
15    # Get the Size
16
17    img.size
```

IP[9]: /1\ /2\ [+] |python3|

h1:[]

code_regions.image_size`

In []\$

```
url = 'https://bit.ly/2zp7YxL'
url
```

In []\$

```
data = requests.get(url, stream=True).raw
data
```

In []\$

```
img = PIL.Image.open(data)
img
```

In []\$

```
img.size
```

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     # Define the URL
7
8     url = 'https://bit.ly/2zp7YxL'
9
10    # Download the Image
11
12    data = requests.get(url, stream=True).raw
13    img = PIL.Image.open(data)
14
15    # Get the Size
16
17    img.size
~
~
```

IP[13]: /1\ /2\ [+] |python3|

h1: `code_regions.image_size`

In [10]:
url = 'https://bit.ly/2zp7YxL'
url
Out [10]:
'https://bit.ly/2zp7YxL'

In [11]:
data = requests.get(url, stream=True).raw
data
Out [11]:
<requests.packages.urllib3.response.HTTPResponse at >

In [12]:
img = PIL.Image.open(data)
img
Out [12]:



code-notebook scrolling

```
def get_ygyl_id():
    """Fetch a YGYL thread ID from the 4chan API

    Returns:
        `thread_id` where http://boards.4chan.org/wsg/thread/{thread\_id}.json
        YGYL thread

    """
    pages = requests.get('https://a.4cdn.org/wsg/catalog.json').json()
    for page in pages:
        for thread in page['threads']:
            if 'ygyl' in thread.get('sub', 'ylly').lower():
                return thread['no']

def get_ygyl_thread_posts():
    """Get the posts from a YGYL thread"""
    ygyl_id = get_ygyl_id()
    thread = requests.get(f'https://a.4cdn.org/wsg/thread/{ygyl\_id}.json')
    return thread['posts']
```

```
[14]: /1\ /2\ [+] |python3
h1: []
`views.get_ygyl_id`  

markdown:
Fetch a YGYL thread ID from the 4chan API

Returns:
`thread_id` where http://boards.4chan.org/wsg/thread/{thread\_id}
YGYL thread  

h1:  

Arguments  

h1:  

Body  

In [9]:
pages = requests.get('https://a.4cdn.org/wsg/catalog.json').json()
pages
Out [9]:
[{'page': 1,
 'threads': [{'capcode': 'mod',
   'closed': 1,
   'com': '1. Please check the Catalog before you post. Popular t...',
   'images': 0,
   'last_modified': 1489416188,
   'last_replies': [{'capcode': 'mod',
     'com': 'Current limits for WebM files on 4chan are:<br>Maxim...',
     'name': 'Anonymous',
     'no': 1587960,
     'replies': 0}]}]
```

setup state via doctest

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     # Define the URL
7
8     url = 'https://bit.ly/2zp7YxL'
9
10    # Download the Image
11
12    data = requests.get(url, stream=True).raw
13    img = PIL.Image.open(data)
14
15    # Get the Size
16
17    img.size
```

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~

1

- 245 code_regions.py

Python

@pynt®®h®

unix | 1: 0 | All

Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     # Download the Image
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10
11    # Get the Size
12
13    img.size
~
```

①

* 187 doctest.py Python @pynt

Y@h

unix | 1: 0 All

Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     # Download the Image
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10
11    # Get the Size
12
13    img.size
~
```

IP[0]: /1\ /2\ [+] |python3|

h1:

doctest.image_size`

h1:

Arguments

In []:

url

h1:

Body

In []:

```
data = requests.get(url, stream=True).raw
data
```

In []:

```
img = PIL.Image.open(data)
img
```

In []:

img.size

~

~

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     # Download the Image
7
8     data = requests.get(url, stream=True).raw
9     img = PIL.Image.open(data)
10
11    # Get the Size
12
13    img.size
~
~
```

```
IP[1]: /1\ /2\ [+] |python3|  
  
h1:  
`doctest.image_size`  
  
h1:  
Arguments  
  
In [1] $  
url  
  
Truncated Traceback (Use C-c C-x to view full TB):  
<ipython-input-1-20be803fa762> in <module>()  
----> 1 url  
  
NameError: name 'url' is not defined  
  
h1:  
Body  
  
In [ ] $  
data = requests.get(url, stream=True).raw  
data  
  
In [ ] $  
img = PIL.Image.open(data)  
img
```

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9
10    """
11    # Download the Image
12
13    data = requests.get(url, stream=True).raw
14    img = PIL.Image.open(data)
15
16    # Get the Size
17
18    img.size
```

1

* 244 doctest.py

Python

@@pynt@ein:c\$@@h@

unix | 8: 4

All

SPC w m

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9
10    """
11    # Download the Image
12
13    data = requests.get(url, stream=True).raw
14    img = PIL.Image.open(data)
15
16    # Get the Size
17
18    img.size
~
```

IP[3]: /1\ /2\ [+] |python3|

h1: □

doctest.image_size`

In []:

```
url = 'https://bit.ly/2zp7YxL'  
url
```

h1:

Arguments

In []:

```
url
```

h1:

Body

In []:

```
data = requests.get(url, stream=True).raw  
data
```

In []:

```
img = PIL.Image.open(data)  
img
```

In []:

```
img.size
```

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9
10    """
11    # Download the Image
12
13    data = requests.get(url, stream=True).raw
14    img = PIL.Image.open(data)
15
16    # Get the Size
17
18    img.size
```

IP[8]: /1\ /2\ [+] |python3|

h1: `doctest.image_size``

In [4]:
url = 'https://bit.ly/2zp7YxL'
url

Out [4]:
'https://bit.ly/2zp7YxL'

h1: `Arguments`

In [5]:
url

Out [5]:
'https://bit.ly/2zp7YxL'

h1: `Body`

In [6]:
data = requests.get(url, stream=True).raw
data

Out [6]:
<requests.packages.urllib3.response.HTTPResponse at >

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9
10    """
11    # Download the Image
12
13    data = requests.get(url, stream=True).raw
14    img = PIL.Image.open(data)
15
16    # Get the Size
17
18    img.size
~
```

IP[8]: Checkpoint created. | /1\ /2\ [+] | python3|

In [5] ↵

url

Out [5]:

'https://bit.ly/2zp7YxL'

h1:

Body

In [6] ↵

data = requests.get(url, stream=True).raw

data

Out [6]:

<requests.packages.urllib3.response.HTTPResponse at >

In [7] ↵

img = PIL.Image.open(data)

img

Out [7]:



```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     Args:
9         url (str) : the image url
10
11    Returns:
12        img_size (tuple) : height and width
13
14    >>> url = 'https://bit.ly/2zp7YxL'
15
16    """
17
18    # Download the Image
19
20    data = requests.get(url, stream=True).raw
21    img = PIL.Image.open(data)
22
23    # Get the Size
24
25    img.size
~
~
```

```
IP[8]: Checkpoint created. | /1\ /2\ [+] |python3|
```

h1: `doctest.image_size``

markdown:
Compute image size at `url`

Args:
 url (str) : the image url

Returns:
 img_size (tuple) : height and width

```
In [ ]>
url = 'https://bit.ly/2zp7YxL'
url
```

h1:
Arguments

```
In [ ]>
url
```

h1:
Body

```
In [ ]>
```

code lifting

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9
10    """
11    # Download the Image
12    data = requests.get(url, stream=True).raw
13    img = PIL.Image.open(data)
14
15    # Get the Size
16    return img.size
```

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~

①

* 276 **lift_try.py**

Python

@pyntY@hK

|

unix | 5: 0 All

Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9
10    """
11    # Download the Image
12    try:
13        data = requests.get(url, stream=True).raw
14        img = PIL.Image.open(data)
15    except ConnectionError as e:
16        print('uh oh')
17        return (-1, -1)
18
19    # Get the Size
20    return img.size
```

~
~
~
~
~
~
~
~
~
~
~
~
~

①

- 373 lift_try.py

Python

@pyntY@hK

unix | 5: 0 All

Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9
10    """
11    # Download the Image
12    try:
13        data = requests.get(url, stream=True).raw
14        img = PIL.Image.open(data)
15    except ConnectionError as e:
16        print('uh oh')
17        return (-1, -1)
18
19    # Get the Size
20    return img.size
```

~
~
~
~
~
~
~
~
~
~
~
~
~
~

IP[5]: /1\ /2\ [+] |python3|

url

In []

Arguments

In []

url

In []

Body

In []

try:

```
    data = requests.get(url, stream=True).raw
    img = PIL.Image.open(data)
```

except ConnectionError as e:

```
    print('uh oh')
```

```
    RETURN = -1, -1
```

```
    raise Exception('return')
```

In []

RETURN = img.size

RETURN

In []

```
raise Exception('return')
```

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9
10    """
11    # Download the Image
12    try:
13        data = requests.get(url, stream=True).raw
14        img = PIL.Image.open(data)
15    except ConnectionError as e:
16        print('uh oh')
17        return (-1, -1)
18
19    # Get the Size
20    return img.size
```

IP[10]: /1\ /2\ [+] |python3|

h1:

Arguments

In [7] ↵

url

Out [7]:

'https://bit.ly/2zp7YxL'

h1:

Body

In [8] ↵

try:

```
    data = requests.get(url, stream=True).raw
    img = PIL.Image.open(data)
```

except ConnectionError as e:

```
    print('uh oh')
```

```
    RETURN = -1, -1
```

```
    raise Exception('return')
```

In [9] ↵

RETURN = img.size

RETURN

Out [9]:

(256, 256)

In [10] ↵

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9
10    """
11    # Download the Image
12    try:
13        data = requests.get(url, stream=True).raw
14        img = PIL.Image.open(data)
15    except ConnectionError as e:
16        print('uh oh')
17        return (-1, -1)
18
19    # Get the Size
20    return img.size
```

IP[22]: /1\ /2\ [+] |python3|

h1:

Arguments

In [19]⚡

url

Out [19]:

'https://bit.ly/2zp7YxL'

h1:

Body

h2:

try

In []⚡

data = requests.get(url, stream=True).raw
data

In []⚡

img = PIL.Image.open(data)
img

h2:

except ConnectionError as e

In []⚡

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9
10    """
11    # Download the Image
12    try:
13        data = requests.get(url, stream=True).raw
14        img = PIL.Image.open(data)
15    except ConnectionError as e:
16        print('uh oh')
17        return (-1, -1)
18
19    # Get the Size
20    return img.size
```

IP[10]: /1\ /2\ [+] |python3|

In [7] ↵

url

Out [7]:

'https://bit.ly/2zp7YxL'

h1:

Body

h2:

try

In [] ↵

```
data = requests.get(url, stream=True).raw
data
```

In [] ↵

```
img = PIL.Image.open(data)
img
```

h2:

except ConnectionError as e

In [] ↵

```
print('uh oh')
```

Tn [] ↵

```

1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9
10    """
11    # Download the Image
12    try:
13        data = requests.get(url, stream=True).raw
14        img = PIL.Image.open(data)
15    except ConnectionError as e:
16        print('uh oh')
17        return (-1, -1)
18
19    # Get the Size
20    return img.size
~
~
~
```

IP[17]: /1\ /2\ [+] |python3|

In [12] ↵
url
Out [12]:
'https://bit.ly/2zp7YxL'

Body

try

In [13] ↵
data = requests.get(url, stream=True).raw
data
Out [13]:
<requests.packages.urllib3.response.HTTPResponse at 0x1000000000000000>

In [14] ↵
img = PIL.Image.open(data)
img
Out [14]:



```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9
10    """
11    # Download the Image
12    try:
13        data = requests.get(url, stream=True).raw
14        img = PIL.Image.open(data)
15    except ConnectionError as e:
16        print('uh oh')
17        return (-1, -1)
18
19    # Get the Size
20    return img.size
```

IP[17]: /1\ /2\ [+] |python3|
'https://bit.ly/2zp7YxL'

Body

try

In [13] ↵
data = requests.get(url, stream=True).raw
data
Out [13]:
<requests.packages.urllib3.response.HTTPResponse at 0x1000000000000000>

In [14] ↵
img = PIL.Image.open(data)
img
Out [14]:



Lifting a try/except

```
try:  
    <body>  
except:  
    <exprs>
```



```
# try  
<body>  
# except  
<exprs>
```

```
1 import PIL
2 import requests
3
4
5 def image_size(url, new_size):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> new_size = (128, 128)
10
11     """
12     # Download the Image
13     data = requests.get(url, stream=True).raw
14     img = PIL.Image.open(data)
15
16     # Resize the Image
17     if new_size:
18         img = img.resize(new_size)
19
20     # Get the Size
21     return img.size
```

~
~
~
~
~
~
~
~
~
~

①

* 392 lift_if.py Python @@pyntYein:c\$@®h®

Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)

unix | 5: 0 All

```
1 import PIL
2 import requests
3
4
5 def image_size(url, new_size):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> new_size = (128, 128)
10
11     """
12     # Download the Image
13     data = requests.get(url, stream=True).raw
14     img = PIL.Image.open(data)
15
16     # Resize the Image
17     if new_size:
18         img = img.resize(new_size)
19
20     # Get the Size
21     return img.size
```

~
~
~
~
~
~
~
~
~

1

* 392 lift_if.py Python @@pynt@ein:c\$@@h®

3 lines unix | 18: 0 All

Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)

```
1 import PIL
2 import requests
3
4
5 def image_size(url, new_size):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> new_size = (128, 128)
10
11     """
12     # Download the Image
13     data = requests.get(url, stream=True).raw
14     img = PIL.Image.open(data)
15
16     # Resize the Image
17     if new_size:
18         img = img.resize(new_size)
19
20     # Get the Size
21     return img.size
```

```
IP[3]: Checkpoint created. | /1\ /2\ [+] | python3| url
In [ ]$ new_size
h1: Body
In [ ]$ data = requests.get(url, stream=True).raw
data
In [ ]$ img = PIL.Image.open(data)
img
In [ ]$ if new_size:
        img = img.resize(new_size)
RETURN = img.size
RETURN
raise Exception('return')
```

```
1 import PIL
2 import requests
3
4
5 def image_size(url, new_size):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> new_size = (128, 128)
10
11     """
12     # Download the Image
13     data = requests.get(url, stream=True).raw
14     img = PIL.Image.open(data)
15
16     # Resize the Image
17     if new_size:
18         img = img.resize(new_size)
19
20     # Get the Size
21     return img.size
22
23
24
25
26
27
28
29
30
31
32
33
34
```

```
IP[12]: /1\ /2\ [+] |python3|
img = PIL.Image.open(data)
img
Out [9]:

```

```
In [10]⚡
if new_size:
    img = img.resize(new_size)
```

```
In [11]⚡
RETURN = img.size
RETURN
Out [11]:
(128, 128)
```

```
In [12]⚡
raise Exception('return')
```

Truncated Traceback (Use C-c C-v to view full TR) ·

```
1 import PIL
2 import requests
3
4
5 def image_size(url, new_size):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> new_size = (128, 128)
10
11     """
12     # Download the Image
13     data = requests.get(url, stream=True).raw
14     img = PIL.Image.open(data)
15
16     # Resize the Image
17     if new_size:
18         img = img.resize(new_size)
19
20     # Get the Size
21     return img.size
```

IP[12]: /1\ /2\ [+] |python3|
img
Out [9]:


h2:
if new_size

In [] ↵
new_size

In [] ↵
img = img.resize(new_size)
img

In [11] ↵
RETURN = img.size
RETURN
Out [11]:

```
1 import PIL
2 import requests
3
4
5 def image_size(url, new_size):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> new_size = (128, 128)
10
11     """
12     # Download the Image
13     data = requests.get(url, stream=True).raw
14     img = PIL.Image.open(data)
15
16     # Resize the Image
17     if new_size:
18         img = img.resize(new_size)
19
20     # Get the Size
21     return img.size
```

IP[10]: /1\ /2\ [+] |python3|
img
Out [6]:


h2:
if new_size

In [7]⚡
new_size
Out [7]:
(128, 128)

In [8]⚡
img = img.resize(new_size)
img

Out [8]:


Lifting a conditional

```
if <test>:  
    <body>  
else:  
    <exprs>
```



```
# if  
<test>  
<body>  
# else  
<exprs>
```

```
1 import PIL
2 import requests
3
4
5 def image_sizes(urls):
6     """Compute image sizes at `urls`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> urls = [url]*2
10
11     """
12     sizes = []
13     for url in urls:
14         # Download the Image
15         data = requests.get(url, stream=True).raw
16         img = PIL.Image.open(data)
17
18         # Get the Size
19         sizes.append(img.size)
20
21     return sizes
```

~
~
~
~
~
~
~
~
~
~

```
1 import PIL
2 import requests
3
4
5 def image_sizes(urls):
6     """Compute image sizes at `urls`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> urls = [url]*2
10
11     """
12     sizes = []
13     for url in urls:
14         # Download the Image
15         data = requests.get(url, stream=True).ra>
16         img = PIL.Image.open(data)
17
18         # Get the Size
19         sizes.append(img.size)
20
21     return sizes
~
~
~
~
~
~
~
```

```
IP[7]: /1\ /2\ [+]\ |python3|
urls

h1:
Body

In [ ]⚡
sizes = []
sizes

In [ ]⚡
for url in urls:
    data = requests.get(url, stream=True).raw
    img = PIL.Image.open(data)
    sizes.append(img.size)

In [ ]⚡
RETURN = sizes
RETURN

In [ ]⚡
raise Exception('return')
~
~
~
~
~
```

```
1 import PIL
2 import requests
3
4
5 def image_sizes(urls):
6     """Compute image sizes at `urls`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> urls = [url]*2
10
11     """
12     sizes = []
13     for url in urls:
14         # Download the Image
15         data = requests.get(url, stream=True).ra>
16         img = PIL.Image.open(data)
17
18         # Get the Size
19         sizes.append(img.size)
20
21     return sizes
```

```
IP[14]: /1\ /2\ [+] |python3|
h1:
Body

In [11]⚡
sizes = []
sizes
Out [11]:
[]

In [12]⚡
for url in urls:
    data = requests.get(url, stream=True).raw
    img = PIL.Image.open(data)
    sizes.append(img.size)

In [13]⚡
RETURN = sizes
RETURN
Out [13]:
[(256, 256), (256, 256)]

In [14]⚡
raise Exception('return')

Truncated Traceback (Use C-c C-x to view full TB):
<ipython-input-14-ad9eb56641f3> in <module>()
----> 1 raise Exception('return')
```

```
1 import PIL
2 import requests
3
4
5 def image_sizes(urls):
6     """Compute image sizes at `urls`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> urls = [url]*2
10
11     """
12     sizes = []
13     for url in urls:
14         # Download the Image
15         data = requests.get(url, stream=True).raw
16         img = PIL.Image.open(data)
17
18         # Get the Size
19         sizes.append(img.size)
20
21     return sizes
```

IP[7]: /1\ /2\ [+] |python3|

h1:

Body

In [4] ↵

```
sizes = []
```

sizes

Out [4]:

```
[]
```

h2:

`for url in urls ...`

In [] ↵

```
urls
```

In [] ↵

```
url = next(iter(urls))
```

url

In [] ↵

```
data = requests.get(url, stream=True).raw
```

data

In [] ↵

```
img = PIL.Image.open(data)
```

img

```
1 import PIL
2 import requests
3
4
5 def image_sizes(urls):
6     """Compute image sizes at `urls`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> urls = [url]*2
10
11     """
12     sizes = []
13     for url in urls:
14         # Download the Image
15         data = requests.get(url, stream=True).raw
16         img = PIL.Image.open(data)
17
18         # Get the Size
19         sizes.append(img.size)
20
21     return sizes
```

```
IP[18]: /1\ /2\ [+] |python3|
h1:
Body

In [11]⚡
sizes = []
sizes
Out [11]:
[]

h2:
`for url in urls ...`
```

```
In [12]⚡
urls
Out [12]:
['https://bit.ly/2zp7YxL', 'https://bit.ly/2zp7YxL']
```

```
In [13]⚡
url = next(iter(urls))
url
Out [13]:
'https://bit.ly/2zp7YxL'
```

```
In [14]⚡
data = requests.get(url, stream=True).raw
data
Out [14]:
```

```

1 import PIL
2 import requests
3
4
5 def image_sizes(urls):
6     """Compute image sizes at `urls`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> urls = [url]*2
10
11     """
12     sizes = []
13     for url in urls:
14         # Download the Image
15         data = requests.get(url, stream=True).ra>
16         img = PIL.Image.open(data)
17
18         # Get the Size
19         sizes.append(img.size)
20
21     return sizes

```

~
~
~
~
~
~
~
~
~
~
~

IP[18]: /1\ /2\ [+] |python3|

h2:

`for url in urls ...`

In [12] ↵

urls

Out [12]:

['https://bit.ly/2zp7YxL', 'https://bit.ly/2zp7YxL']

In [13] ↵

url = next(iter(urls))

url

Out [13]:

'https://bit.ly/2zp7YxL'

In [14] ↵

data = requests.get(url, stream=True).raw

data

Out [14]:

<requests.packages.urllib3.response.HTTPResponse at >

In [15] ↵

img = PIL.Image.open(data)

img

Out [15]:



```
1 import PIL
2 import requests
3
4
5 def image_sizes(urls):
6     """Compute image sizes at `urls`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> urls = [url]*2
10
11     """
12     sizes = []
13     for url in urls:
14         # Download the Image
15         data = requests.get(url, stream=True).ra>
16         img = PIL.Image.open(data)
17
18         # Get the Size
19         sizes.append(img.size)
20
21     return sizes
```

```
IP[18]: /1\ /2\ [+] |python3|
urls
Out [12]:
['https://bit.ly/2zp7YxL', 'https://bit.ly/2zp7YxL']>

In [13]⚡
url = next(iter(urls))
url
Out [13]:
'https://bit.ly/2zp7YxL'

In [14]⚡
data = requests.get(url, stream=True).raw
data
Out [14]:
<requests.packages.urllib3.response.HTTPResponse at >

In [15]⚡
img = PIL.Image.open(data)
img
Out [15]:

```

```

1 import PIL
2 import requests
3
4
5 def image_sizes(urls):
6     """Compute image sizes at `urls`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9     >>> urls = [url]*2
10
11     """
12     sizes = []
13     for url in urls:
14         # Download the Image
15         data = requests.get(url, stream=True).ra>
16         img = PIL.Image.open(data)
17
18         # Get the Size
19         sizes.append(img.size)
20
21     return sizes
~
~
~
```

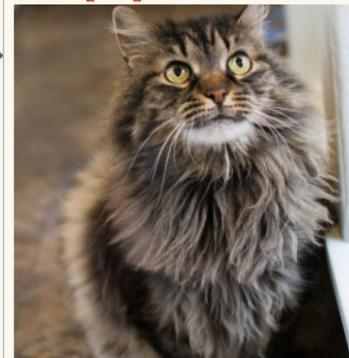
IP[18]: /1\ /2\ [+] |python3|
'https://bit.ly/2zp7YxL'

In [14] ↴
data = requests.get(url, stream=True).raw
data

Out [14]:
<requests.packages.urllib3.response.HTTPResponse at >

In [15] ↴
img = PIL.Image.open(data)
img

Out [15]:



In [16] ↴
sizes.append(img.size)

In [17] ↴
DETERMINE = sizes

Lifting a for loop

```
for i in <iterable>:  
    <body>
```



```
# for i in <iterable>  
i = next(iter(<iterable>))  
<body>
```

breakpoints

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     >>> url = 'https://bit.ly/2zp7YxL'
9
10    """
11    # Download the Image
12    data = requests.get(url, stream=True).raw
13    img = PIL.Image.open(data)
14
15    # Get the Size
16    return img.size
```

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~

①

* 276 lift_try.py

Python

@pyntY@hK

unix | 5: 0 All

Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
```

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
~
~
```

```
IP[0]: /1\ /2\ [+] |python3|
h1:
`breakpoint.image_size`  
  
markdown:  
Compute image size at `url`  
  
h1:  


## Arguments

  
In [ ]$  
url  
  
h1:  


## Body

  
In [ ]$  
data = requests.get(url, stream=True).raw  
data  
  
In [ ]$  
img = PIL.Image.open(data)  
img  
  
In [ ]$  
RETURN = img.size  
RETURN
```

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
```

IP[1]: /1\ /2\ [+] |python3|

h1:

breakpoint.image_size`

markdown:

Compute image size at `url`

h1:

Arguments

In [1] ↵

url

Truncated Traceback (Use C-c C-x to view full TB):

<ipython-input-1-20be803fa762> in <module>()

----> 1 url

NameError: name 'url' is not defined

h1:

Body

In [] ↵

data = requests.get(url, stream=True).raw

data

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
14
15
16 if __name__ == '__main__':
17     url = 'https://bit.ly/2zp7YxL'
18     image_size(url)
```

~
~
~
~
~
~
~
~
~
~
~
~
~
~

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
14
15
16 if __name__ == '__main__':
17     url = 'https://bit.ly/2zp7YxL'
18     image_size(url)
~
~
```

```
[IPython]: /1\ /2\ [+] |python3|
```

h1:
`breakpoint.image_size`

markdown:
Compute image size at `url`

h1:
Arguments

In []:
url

h1:
Body

In []:
data = requests.get(url, stream=True).raw
data

In []:
img = PIL.Image.open(data)
img

In []:
RETURN = img.size
RETURN

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
14
15
16 if __name__ == '__main__':
17     url = 'https://bit.ly/2zp7YxL'
18     image_size(url)
```

① - 315 breakpoint.py unix | 5: 0 All

Welcome to the Emacs shell

~/Documents/demo λ python breakpoint.py |

IP[1]: /1\ /2\ [+] |python3|

h1:

`breakpoint.image_size`

markdown:

Compute image size at `url`

h1:

Arguments

In [1] ↵

url

Truncated Traceback (Use C-c C-x to view full TB):
<ipython-input-1-20be803fa762> in <module>()
----> 1 url

NameError: name 'url' is not defined

3

* 68 *eshell-1*

Eshell

@Paredit@h®

utf-8 | 3:40

All

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
14
15
16 if __name__ == '__main__':
17     url = 'https://bit.ly/2zp7YxL'
18     image_size(url)
~
~
```

① - 315 breakpoint.py

unix | 5: 0 All

IP[1]: Checkpoint created. | /1\ /2\ [+] | python3|

h1:

`breakpoint.image_size`

markdown:

Compute image size at `url`

h1:

Arguments

In [1] ↵

url

Truncated Traceback (Use C-c C-x to view full TB):
<ipython-input-1-20be803fa762> in <module>()
----> 1 url

NameError: name 'url' is not defined

To read more about this, see <https://github.com/ipython/ipython/issues/2049>

To connect another client to this kernel, use:

--existing kernel-55136.json

③

* 743 *eshell-1*

Eshell

@Paredit@h®

utf-8 | 16: 0 Bottom

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
14
15
16 if __name__ == '__main__':
17     url = 'https://bit.ly/2zp7YxL'
18     image_size(url)
~
~
```

① - 315 breakpoint.py

unix | 5: 0 All

IP[1]: /1\ /2\ [+] |python3|

h1:

`breakpoint.image_size`

markdown:

Compute image size at `url`

h1:

Arguments

In [1] ↵

url

Truncated Traceback (Use C-c C-x to view full TB):
<ipython-input-1-20be803fa762> in <module>()
----> 1 url

NameError: name 'url' is not defined

To connect another client to this kernel, use:

--existing kernel-44589.json



③ * 743 *eshell-1* Eshell

Really restart kernel? (y or n)

utf-8 | 16: 0 Bottom

```
import PIL
import requests

def image_size(url):
    """Compute image size at `url`"""

    # Download the Image
    data = requests.get(url, stream=True).raw
    img = PIL.Image.open(data)

    # Get the Size
    return img.size

if __name__ == '__main__':
    url = 'https://bit.ly/2zp7YxL'
    image_size(url)
~
```

IP[1]: /1\ /2\ [+]|python3|

h1:

`breakpoint.image_size`

markdown:

Compute image size at `url`

h1:

Arguments

In [1] ↵

url

Out [1]:

'https://bit.ly/2zp7YxL'

h1:

Body

① - 315 breakpoint.py

unix | 5: 0 All

2

Top

To connect another client to this kernel, use:

--existing kernel-44589.json



③ * 743 *eshell-1* Eshell

utf-8 | 16: 0 Bottom

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
14
15
16 if __name__ == '__main__':
17     url = 'https://bit.ly/2zp7YxL'
18     image_size(url)
```

1 - 315 breakpoint.py

unix | 10: 0 All

IP[6]: /1\ /2\ [+] |python3|

h1:

Body

In [3] ↵

```
data = requests.get(url, stream=True).raw
data
```

Out [3]:

```
<requests.packages.urllib3.response.HTTPResponse at >
```

In [4] ↵

```
img = PIL.Image.open(data)
img
```

Out [4]:



To connect another client to this kernel, use:

```
--existing kernel-44589.json
```



3 * 743 *eshell-1* Eshell

utf-8 | 16: 0 Bottom

Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)

exception catching

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
14
15
16 if __name__ == '__main__':
17     url = 'https://bit.ly/2zp7YxL'
18     image_size(url)
```

~
~
~
~
~
~
~
~
~
~
~
~
~
~

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
14
15
16 if __name__ == '__main__':
17     url = 'https://this-is-a-bogus-url.com'
18     image_size(url)
```

1

* 324 catch_exc.py Python @pynt®®@h®

Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)

unix | 5: 0 All

```
1 import PIL
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
14
15
16 if __name__ == '__main__':
17     url = 'https://this-is-a-bogus-url.com'
18     image_size(url)
```

~

~

① * 324 catch_exc.py Python

unix | 5: 0 All

Welcome to the Emacs shell

~/Documents/demo λ python catch_exc.py |

② * 67 *eshell-1* Eshell Ⓜ Paredit Ⓝ h Ⓞ

utf-8 | 3:39 All

History item: 49

```
1 import PIL.Image
2 import requests
3
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
14
15 if __name__ == '__main__':
16     url = 'https://this-is-a-bogus-url.com'
17     image_size(url)
```

~
~
~

① - 329 catch_exc.py Python

unix | 3: 0 All

During handling of the above exception, another exception occurred:

```
Traceback (most recent call last):
  File "catch_exc.py", line 17, in <module>
    image_size(url)
  File "catch_exc.py", line 9, in image_size
    data = requests.get(url, stream=True).raw
  File "/Users/ebanner/.anaconda/lib/python3.6/site-packages/requests/api.py", line 70, in get
```

② * 4.2k *eshell-1* Eshell @Paredit@h®

utf-8 | 41: 0 68%

```
1 import PIL.Image
2 import requests
3 
4
5 def image_size(url):
6     """Compute image size at `url`"""
7
8     # Download the Image
9     data = requests.get(url, stream=True).raw
10    img = PIL.Image.open(data)
11
12    # Get the Size
13    return img.size
14
15 if __name__ == '__main__':
16     url = 'https://this-is-a-bogus-url.com'
17     image_size(url)
```

```
~
```

~

~

① * 329 catch_exc.py Python unix | 3: 0 All

```
resp = self.send(prep, **send_kwargs)
File "/Users/ebanner/.anaconda/lib/python3.6/site-packages/requests/sessions.py", line 609, in send
    r = adapter.send(request, **kwargs)
File "/Users/ebanner/.anaconda/lib/python3.6/site-packages/requests/adapters.py", line 487, in send
    raise ConnectionError(e, request=request)
requests.exceptions.ConnectionError: HTTPSConnectionPool(host='this-is-a-bogus-url.com', port=443): Max r>
~/Documents/demo λ |
```

② * 4.2k *eshell-1* Eshell @Paredit@h®

utf-8 | 57:19 Bottom

SPC w j

```
1 import PIL.Image
2 import requests
3 import codebook
4 codebook.register()
5
6
7 def image_size(url):
8     """Compute image size at `url`"""
9
10    # Download the Image
11    data = requests.get(url, stream=True).raw
12    img = PIL.Image.open(data)
13
14    # Get the Size
15    return img.size
16
17 if __name__ == '__main__':
18     url = 'https://this-is-a-bogus-url.com'
19     image_size(url)
```

1 * 365 catch_exc.py Python @pynt®®h®

```
resp = self.send(prep, **send_kwargs)
File "/Users/ebanner/.anaconda/lib/python3.6/site-packages/requests/sessions.py", line 609, in send
    r = adapter.send(request, **kwargs)
File "/Users/ebanner/.anaconda/lib/python3.6/site-packages/requests/adapters.py", line 487, in send
    raise ConnectionError(e, request=request)
requests.exceptions.ConnectionError: HTTPSConnectionPool(host='this-is-a-bogus-url.com', port=443): Max r>
~/Documents/demo λ |
```

2 * 4.2k *eshell-1* Eshell

```
utf-8 | 57:19 Bottom
Server error: TypeError("__init__() got an unexpected keyword argument 'environment'",)
```

```
1 import PIL.Image
2 import requests
3 import codebook
4 codebook.register()
5
6
7 def image_size(url):
8     """Compute image size at `url`"""
9
10    # Download the Image
11    data = requests.get(url, stream=True).raw
12    img = PIL.Image.open(data)
13
14    # Get the Size
15    return img.size
16
17 if __name__ == '__main__':
18     url = 'https://this-is-a-bogus-url.com'
19     image_size(url)
```

```
① - 365 catch_exc.py Python                               unix | 3: 0 All
    resp = self.send(prep, **send_kwargs)
File "/Users/ebanner/.anaconda/lib/python3.6/site-packages/requests/sessions.py", line 609, in send
    r = adapter.send(request, **kwargs)
File "/Users/ebanner/.anaconda/lib/python3.6/site-packages/requests/adapters.py", line 487, in send
    raise ConnectionError(e, request=request)
requests.exceptions.ConnectionError: HTTPSConnectionPool(host='this-is-a-bogus-url.com', port=443): Max r>
~/Documents/demo λ python catch_exc.py |
```

```
1 import PIL.Image
2 import requests
3 import codebook
4 codebook.register()
5
6
7 def image_size(url):
8     """Compute image size at `url`"""
9
10    # Download the Image
11    data = requests.get(url, stream=True).raw
12    img = PIL.Image.open(data)
13
14    # Get the Size
15    return img.size
16
17 if __name__ == '__main__':
18     url = 'https://this-is-a-bogus-url.com'
19     image_size(url)
~
```

① - 365 catch_exc.py Python

unix | 3: 0 All

To read more about this, see <https://github.com/ipython/ipython/issues/2049>

To connect another client to this kernel, use:

--existing kernel-45249.json

② * 9.0k *eshell-1* Eshell @Paredit@h®

utf-8 | 124: 0 Bottom

```
1 import PIL.Image
2 import requests
3 import codebook
4 codebook.register()
5
6
7 def image_size(url):
8     """Compute image size at `url`"""
9
10    # Download the Image
11    data = requests.get(url, stream=True).raw
12    img = PIL.Image.open(data)
13
14    # Get the Size
15    return img.size
16
17 if __name__ == '__main__':
18     url = 'https://this-is-a-bogus-url.com'
19     image_size(url)
```

① - 365 catch_exc.py unix | 7: 0 All

To read more about this, see <https://github.com/ipython/ipython/issues/2049>

To connect another client to this kernel, use:
--existing kernel-45249.json

IP[0]: Checkpoint created. | /1\ /2\ [+]| python3|

catch_exc.image_size

markdown:
Compute image size at `url`

Arguments

In []
url

Body

In []
data = requests.get(url, stream=True).raw

③ * 9.0k *eshell-1* Eshell

ein: [info] Opened notebook list at http://localhost:8889 with path Documents/demo.

utf-8 | 124: 0 Bottom

```
1 import PIL.Image
2 import requests
3 import codebook
4 codebook.register()
5
6
7 def image_size(url):
8     """Compute image size at `url`"""
9
10    # Download the Image
11    data = requests.get(url, stream=True).raw
12    img = PIL.Image.open(data)
13
14    # Get the Size
15    return img.size
16
17 if __name__ == '__main__':
18     url = 'https://this-is-a-bogus-url.com'
19     image_size(url)
~
```

① - 365 catch_exc.py

unix | 7: 0 All

2

Top

To read more about this, see <https://github.com/ipython/ipython/issues/2049>

To connect another client to this kernel, use:
--existing kernel-45249.json

③ * 9.0k *eshell-1* Eshell

IP[1]: /1\ /2\ [+] python3

h1:

catch_exc.image_size

markdown:

Compute image size at `url`

h1:

Arguments

In [1] ↵

url

Out [1]:

'https://this-is-a-bogus-url.com'

h1:

Body

utf-8 | 124: 0 Bottom

```
1 import PIL.Image
2 import requests
3 import codebook
4 codebook.register()
5
6
7 def image_size(url):
8     """Compute image size at `url`"""
9
10    # Download the Image
11    data = requests.get(url, stream=True).raw
12    img = PIL.Image.open(data)
13
14    # Get the Size
15    return img.size
16
17 if __name__ == '__main__':
18     url = 'https://this-is-a-bogus-url.com'
19     image_size(url)
```

1 - 365 catch_exc.py

unix | 7: 0 All

2

Top

IP[1]: /1\ /2\ [+] python3

h1:

catch_exc.image_size

markdown:

Compute image size at `url`

h1:

Arguments

In [*] \$

url

h1:

Body

In [] \$

data = requests.get(url, stream=True).raw

To connect another client to this kernel, use:

--existing kernel-45249.json

^ \

quit: 3

~/Documents/demo λ |

③ * 9.0k *eshell-1* Eshell

utf-8 | 126:19 Bottom

Feature recap

- Jupyter notebook management
 - Scrolling
- Code dumping
- Unpacking (syntax rewrites)
 - try/except
 - if
 - for
- Setting up state
 - doctest docstring
 - Attach to an external command (i.e jupyter notebook --existing)
- Top-level exception handler
 - i.e. python -m pdb download.py

pynt discussion

- What I has pynt helped build?
- Early lessons/observations
- Other similar tools
- The future

What have I built with pynt?

- pynt (!)
 - <https://github.com/ebanner/pynt>
- Django web app
 - <https://github.com/ebanner/ygyl-player>

Early lessons

pynt makes it **really** easy to write bad code

```
480
481     def visit_Try(self, tryexp):
482         """Unpack a try/except line.
483
484         >>> self = UnpackTry('foo', False)
485         >>> code = '''
486         ...
487         ... try:
488         ...     data = requests.get(url, stream=True).raw
489         ... except:
490         ...     print('error!')
491         ...
492         ...
493         >>> module = ast.parse(code)
494         >>> tryexp, = module.body
495
496         '''
497         nodes = []
498         nodes.append(make_annotation(buffer=self.buffer, content='try', cell_type='2', lineno=tryexp.lineno))
499         nodes.extend(tryexp.body)
500         if self.only_try:
501             return nodes
502
503         for handler in tryexp.handlers:
504             handler_toks = ['except']
505             if handler.type:
506                 handler_type = astor.to_source(handler.type).strip()
507                 handler_toks.append(handler_type)
508             if handler.name:
509                 handler_toks.extend(['as', handler.name])
510             handler_str = ' '.join(handler_toks)
511             nodes.append(make_annotation(buffer=self.buffer, content=handler_str, cell_type='2', lineno=tryexp.lineno))
512             nodes.extend(handler.body)
513         if tryexp.orelse:
514             nodes.append(make_annotation(buffer=self.buffer, content='else', cell_type='2', lineno=tryexp.lineno))
515             nodes.extend(tryexp.orelse)
516         if tryexp.finalbody:
517             nodes.append(make_annotation(buffer=self.buffer, content='finally', cell_type='2', lineno=tryexp.lineno))
518             nodes.extend(tryexp.finalbody)
519
520         return nodes
```

It's easy to get tunnel
vision.

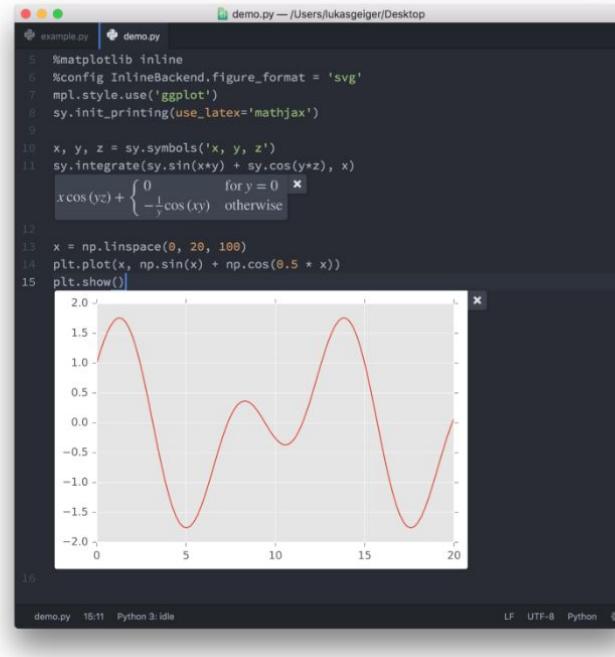
pynt is in trouble

Barrier to entry barrier is 🔥 .

Other tools?

- 1

- Hydrogen (Atom)



What is the future for pynt?

- Standalone code to notebook converter?
- Adding jupyter notebook --existing to jupyter?
- JupyterLab plugin?
- Go with Hydrogen or PyCharm?
- What is the niche for pynt?

Thank you.

Questions?