PyVis: Python Module Visualization

115305288@qq.com　　https://pyvisvue3d3.yingshinet.com

0- Download and Prepare for Running

When you download the PyVis, unzip it in your disk, such as:

Next please open the project in your own compile software,set pycharm as an example here.

Then install the python module needed here: flask, flask_cores, inspect, importlib, requests...... you can directly run pip install -r requirements.txt to install all modules needed and compile main.py to run the whole project.
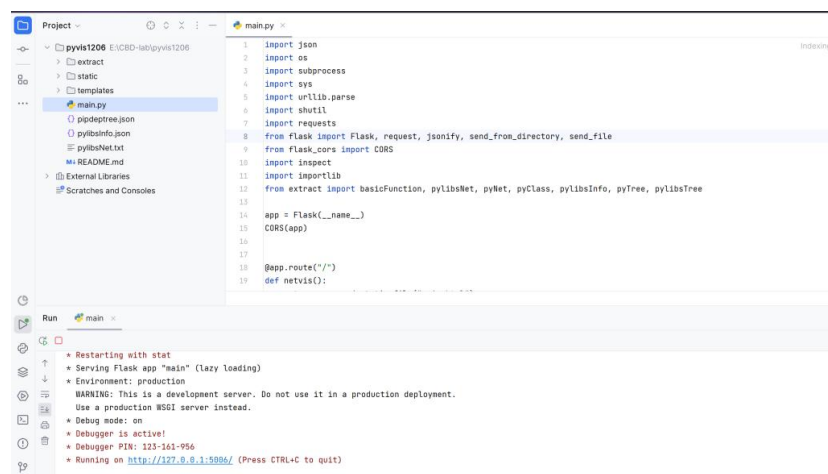

Open the link http://127.0.0.1:5006/ on the browser to use the tool
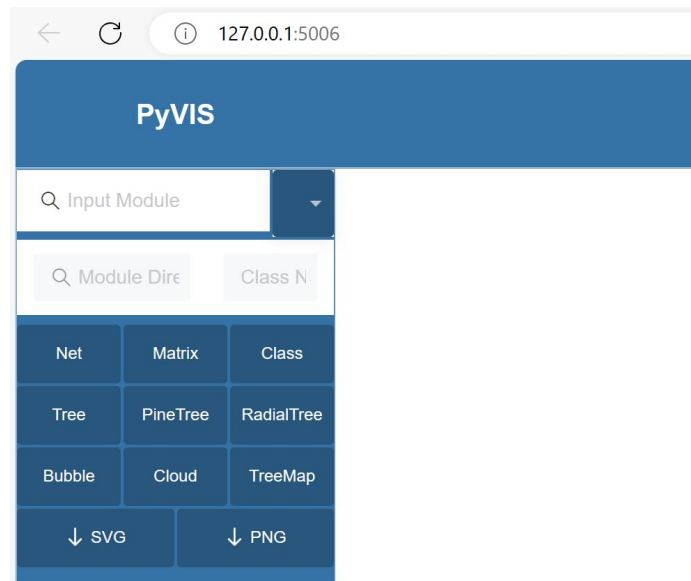
## 1- Extract python modules in your local environment

Focus on the left region and click the button [ExtractAll] to extract all modules in your local environment, the backend programming will scan your site-packages directory and analyze all modules to generate some .json files provided for visualization and module analysis. Also if you want to view details of certain module or to update the specific information, It's available to input the module name and click the button [ExtractSingle] to extract message of the module again.
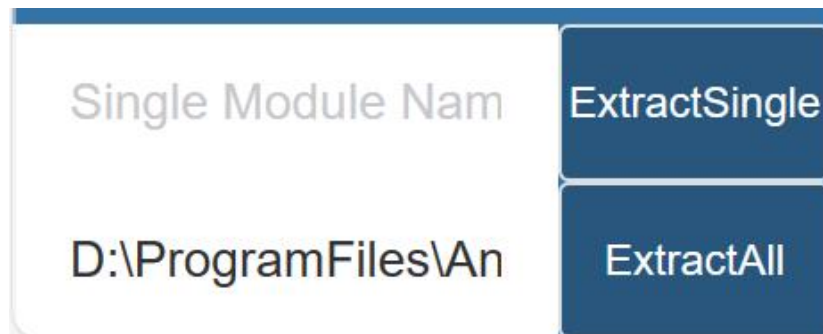
2- Entire Package Environment Visualization

(1) Net json file

Once open the link, the default whole environment module visualization file would be called to show the relations between all inner modules with a force-directed graph. Each circle on the graph refers to certain module in your python environment. Links refer to the relations between modules: Requires or required by.
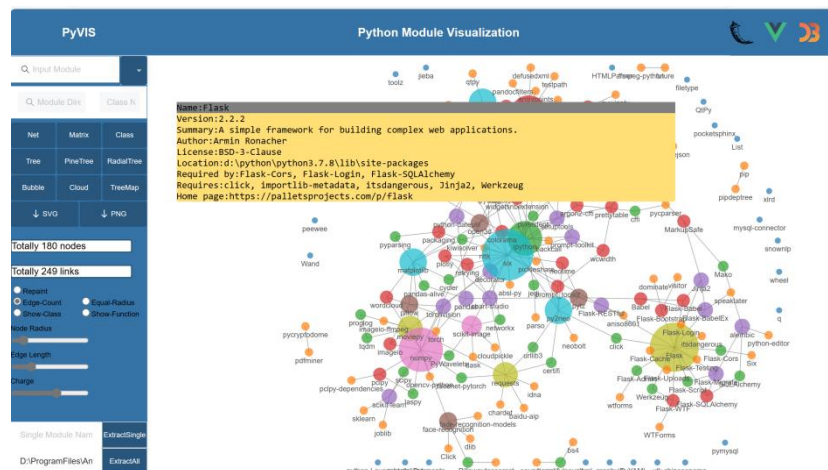
If you click the module name on the force-directed graph, the system will search the module selected automatically.

Also, the [Matrix] graph, [Class] graph also shows the net json file in different kind of forms.

(2) Tree json file

Click the [tree] button to show the hierarchical tree graph of the extracted tree json file. Different forms of graph are display in rest 5 buttons,including PineTree, RadinalTree, Bubble, Cloud and TreeMap. The detailed message of each module will be described in chapter 3.
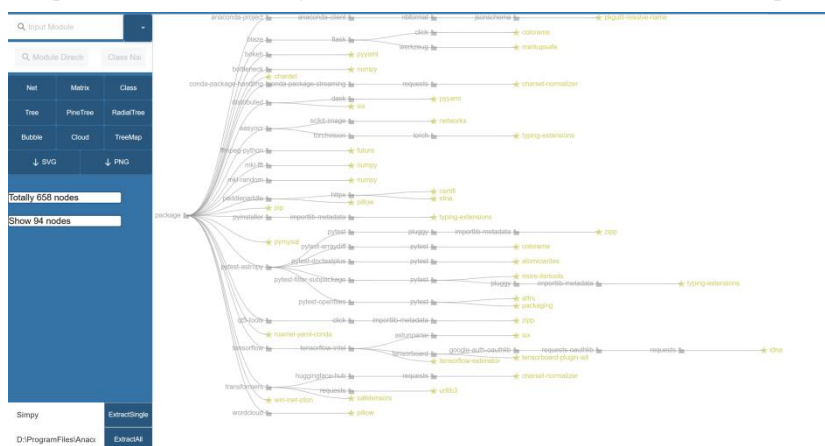

3- Single Package Visualization

(1) Retrieval

Input the module name in the first search box and press enter, also it is available to pull the dropdownlist to choose the module name without input. After this operation, the whole visualization is based on the module selected.
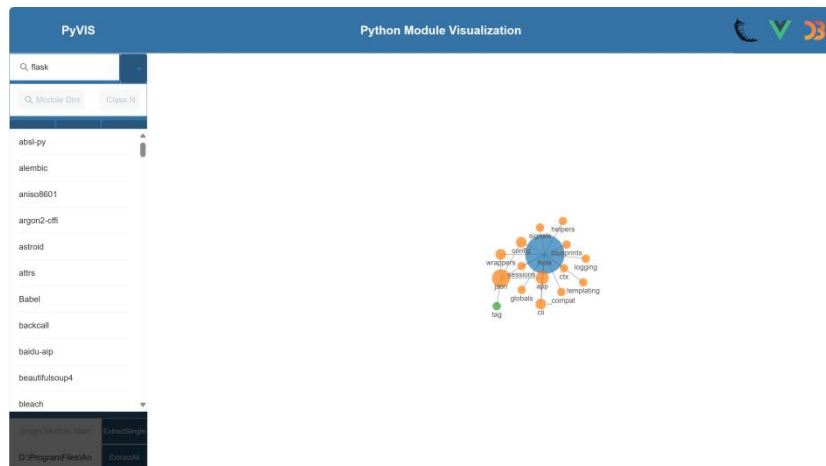
For the second search box, input the module directory in the left column, the class name in the right column(not necessary), thus it is available to retrieval python document and code at any directory and file.
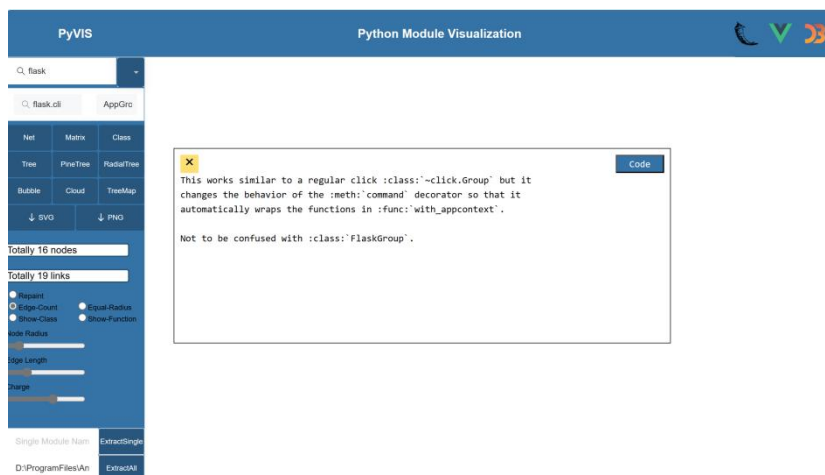
(2) Net Module: force-directed graph visualization

Click the [Net] button to display a force-directed graph illustrating the relationships among related directories and files. A pop-up window will show module information upon mouseover operation. Additionally, a pop-up window will display the module code and documentation upon mouse-click operation.
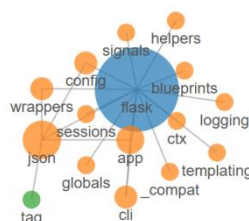
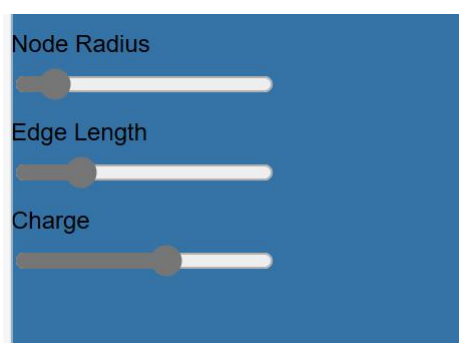```
×   Code

flask.json
~~~~~~~~~~

:copyright: 2010 Pallets
:license: BSD-3-Clause
```

If you want to change the style of the force-directed graph, a control area list three option regions for you, in which you can change the node radius,edge length and charge of the graph for better visualization result.



### (3) Matrix Module: matrix diagram visualization

Click the [Matrix] button to display a Matrix diagram illustrating the relationships among related directories and files. The left area shows the names of packages and files. A pop-up window will show module information when the text is clicked. Additionally, The matrix points and circles in the page have full interaction with the text.
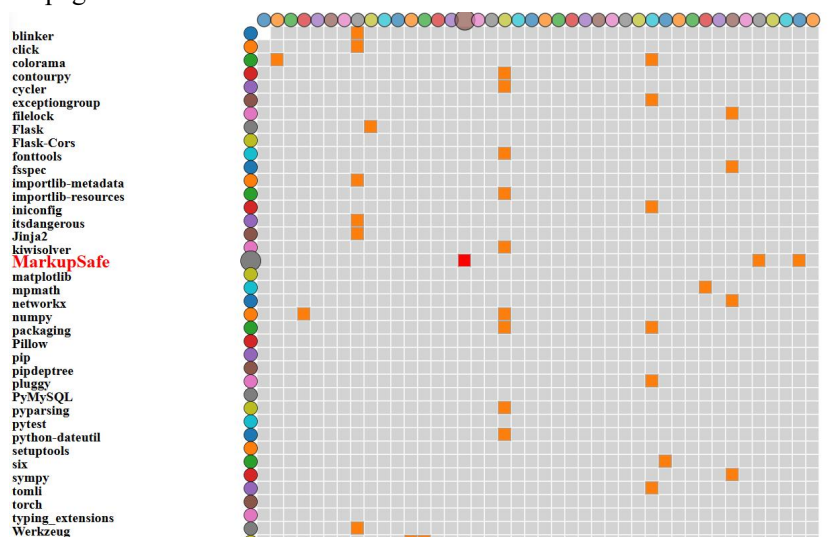
```
Wrapper functions to more user-friendly calling of certain math functions
whose output data-type is different than the input data-type in certain
domains of the input.

For example, for functions like `log` with branch cuts, the versions in this
module provide the mathematically valid answers in the complex plane::

    >>> import math
    >>> np.emath.log(-math.exp(1)) == (1+1j*math.pi)
    True
```
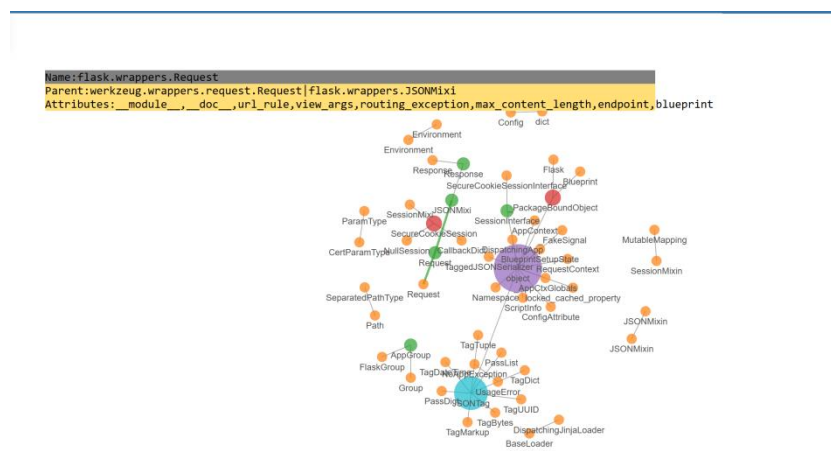
## (4) Class Module: class force-directed graph visualization

The class module provide a force-directed graph that displays the information of all classes contained in the module, also you can call the information window by mouse-over option or call the document and code window by click option. Control region is available as well as the [Net] module.



## (5) Tree Module:   Horizontal graph visualization

The tree graph provides a hierarchy structure of the environment's module directory, set torch module as an example here, the extracted .json file contains the module information as well as the doc and related online links, so the visualization is based on the extracted information mentioned above. Each node is clickable, for the leaf node, related file links are provided aside,for none-leaf node, each node is extendable once you click the node icon for the convenient visualization of large packages.
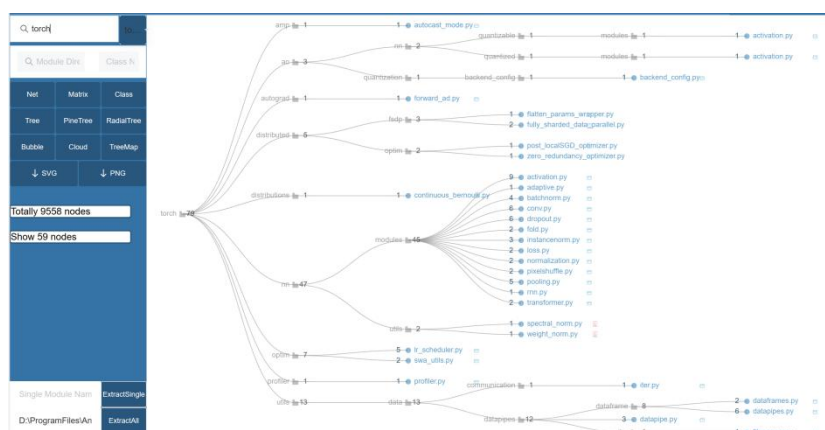
Once you click the leaf node, detailed information is provided with a pop-up window which includes build-in classes and build-out classes tree. Also each class is clickable and points to its code or doc window.
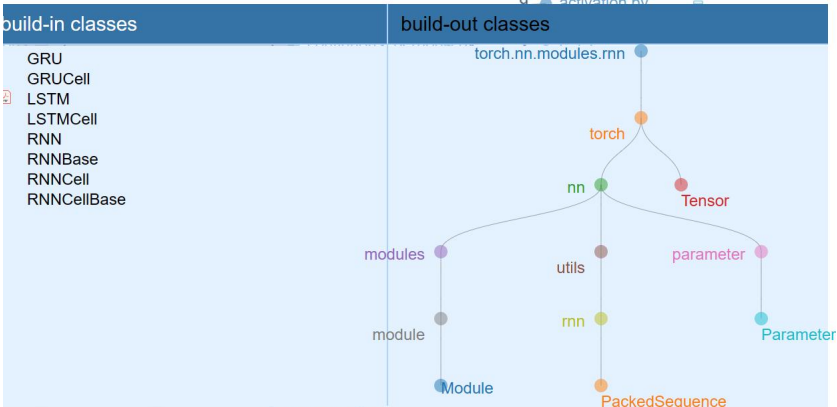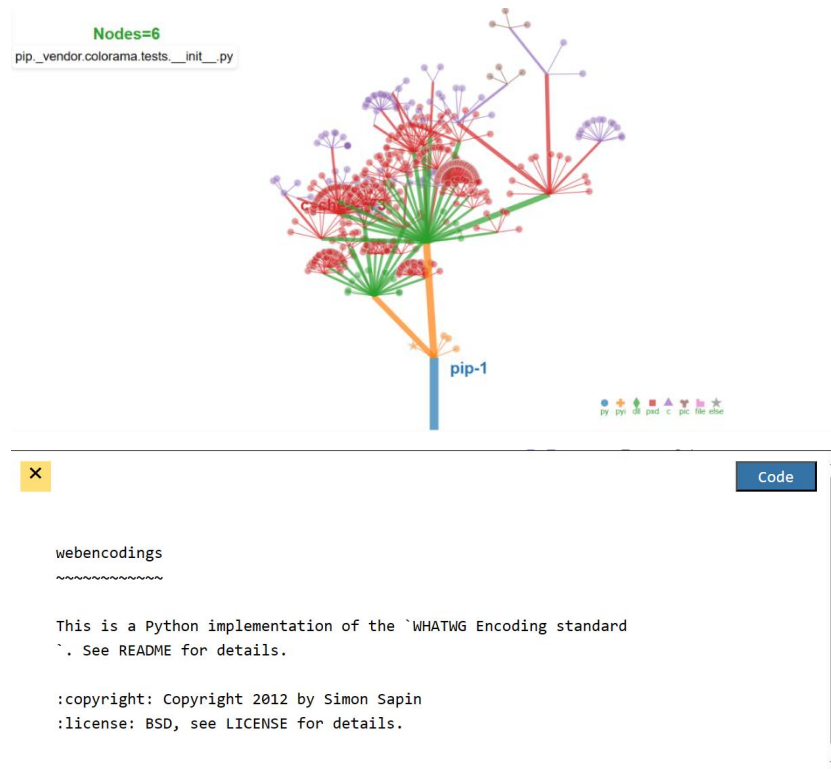


On click the build-in class name, a pop-up window which lists the variables, functions and document of the certain class name.



### (6) PineTree Module: tree diagram visualization

Click the [PineTree] button to display a tree map indicating the hierarchical structure of the relevant directories and files. The upper left corner pop-up window will show the full level information of the module upon mouseover operation. Additionally, a pop-up window will display the module code and documentation upon mouse-click operation. Several different colors appear on the right side of the page, and the same color text in the word cloud will disappear upon mouse-click operation, and when you click again, it will appear.
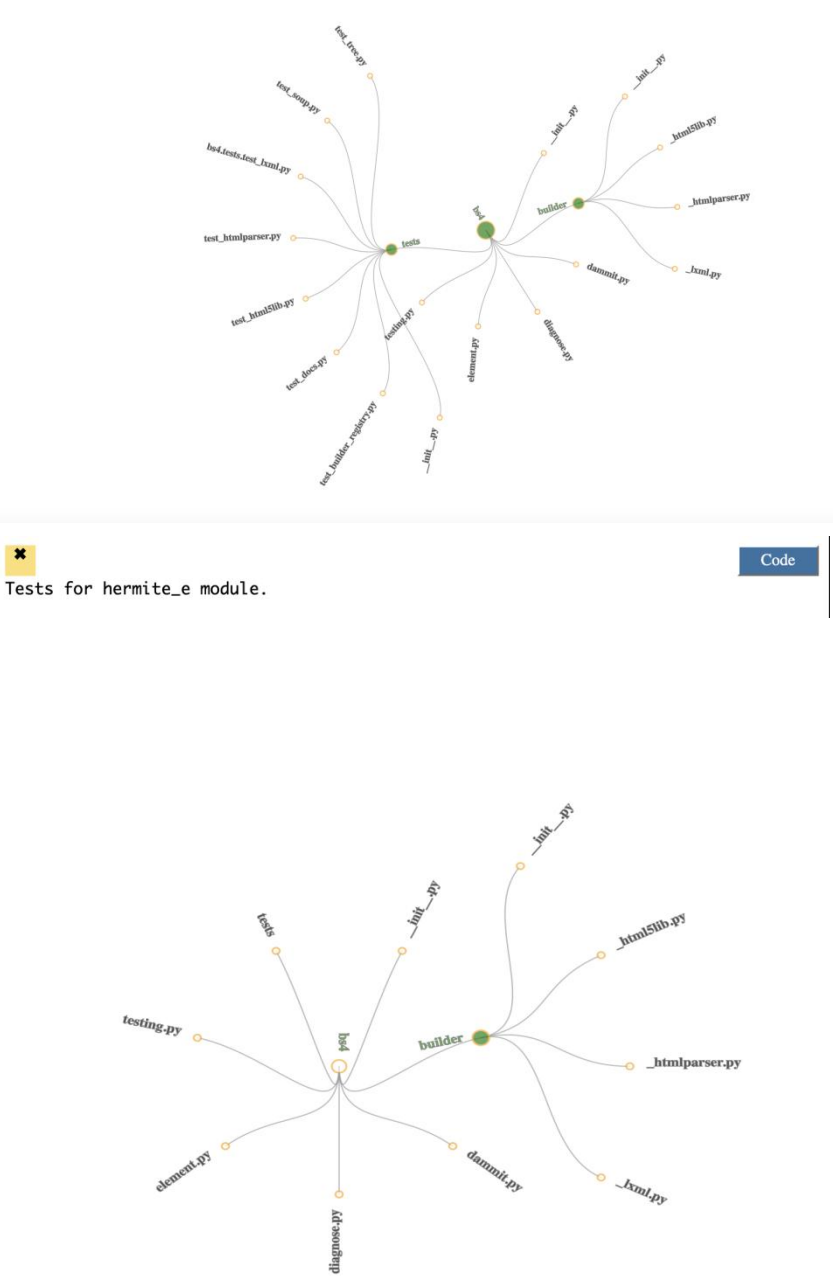
Nodes=6
pip._vendor.colorama.tests.__init__.py

pip-1

py pyi dll pxd c pic file else

```
×                                                            Code

     webencodings
     ~~~~~~~~~~~~

     This is a Python implementation of the `WHATWG Encoding standard
     `. See README for details.

     :copyright: Copyright 2012 by Simon Sapin
     :license: BSD, see LICENSE for details.
```

At the same time, the left control area shows the number of nodes on the page，"Length"，"Angel"，"Rate"three adjustable sliders and the PDF and GitHub selection boxes appear. You can adjust the length of the branches, the Angle between the branches, and the length ratio between the branches at each level of the tree using the three sliders. If the selection box is selected, a pop-up window will display the PDF file or GitHub file contained in the current page,and if no, it will appear the corresponding prompt. The label selection box is used to determine whether to display node information on the page.

Totally 6 nodes

Length

Angel

Rate

☑ Label

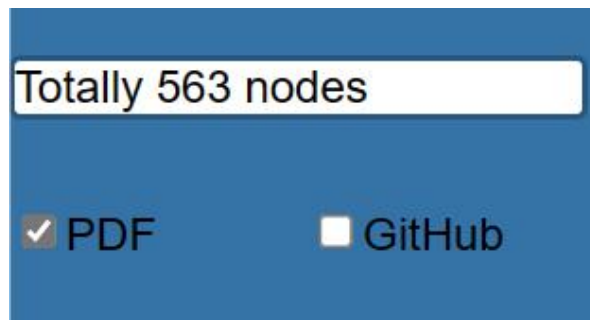○ Repaint

☐ PDF   ☐ GitHub

(7) RadialTree Module: radialtree visualization

Click the [RadialTree] button to display a Bubble diagram indicating the hierarchical status of the relevant directories and files.When hovering over node names, the complete module names will be shown. Additionally, upon a mouse-click operation on the text, a window will pop up

displaying the module's code and documentation; clicking on nodes allows for collapsing operations on those nodes.

At the same time, the left control area shows the number of nodes on the page and the PDF and GitHub selection boxes appear.If the selection box is selected, a pop-up window will display the PDF file or GitHub file contained in the current page,and if no, it will appear the corresponding prompt.

(8) Bubble Module: bubble diagram visualization

Click the [Bubble] button to display a Bubble diagram indicating the hierarchical status of the relevant directories and files. The deeper the level of the module, the smaller the radius of the circle. A pop-up window will show the full level information of the module upon mouseover operation. Additionally, a pop-up window will display the module code and documentation upon mouse-click operation. Several different colors appear on the right side of the page, and the same color text in the word cloud will disappear upon mouse-click operation, and when you click again, it will appear.

At the same time, the left control area shows the number of nodes on the page and the PDF and GitHub selection boxes appear.If the selection box is selected, a pop-up window will display the PDF file or GitHub file contained in the current page,and if no, it will appear the corresponding prompt.
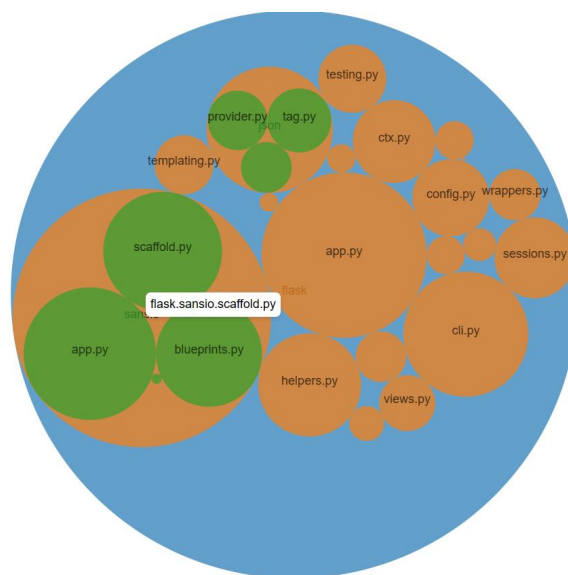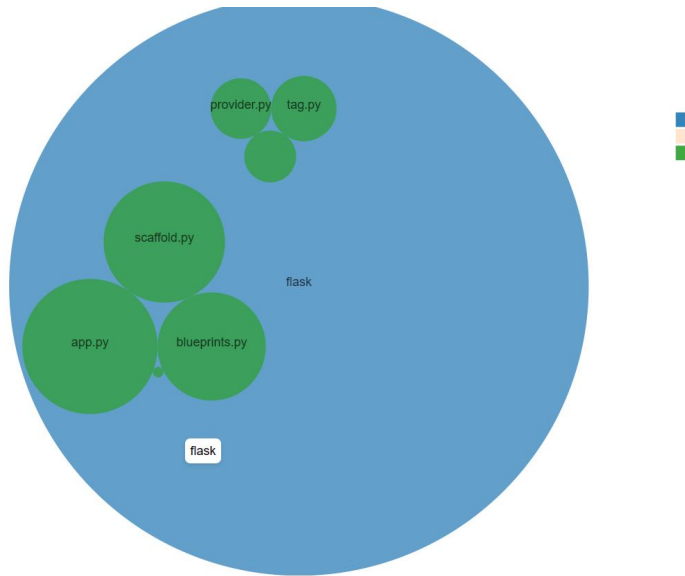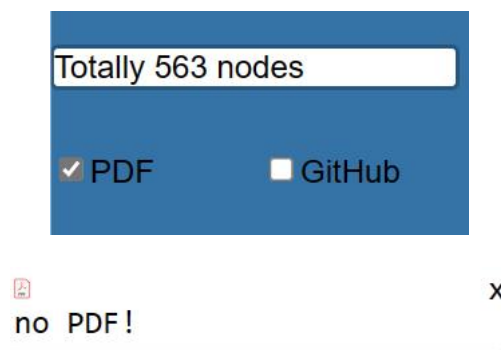

(9) Cloud Module: wordcloud visualization

Click the [Cloud] button to display a word cloud map indicating the hierarchical status of the relevant directories and files. The deeper the level of the module, the smaller the font .The upper left corner pop-up window will show the full level information of the module upon mouseover operation. Additionally, a pop-up window will display the module code and documentation upon mouse-click operation. Several different colors appear on the right side of the page, and the same color text in the word cloud will disappear upon mouse-click operation, and when you click again, it will appear.
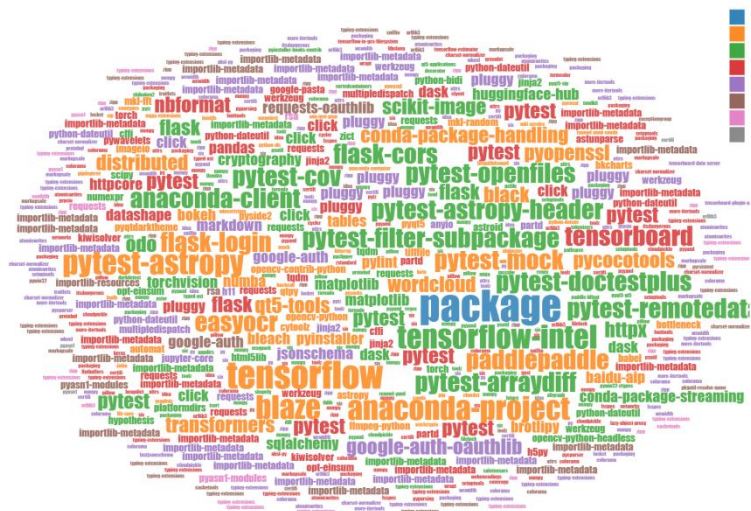
```
×                                                    Code

pip._vendor is for vendoring dependencies of pip to prevent needing pip to
depend on something external.

Files inside of pip._vendor should be considered immutable and should only be
updated to versions from upstream.
```

At the same time, the left control area shows the number of nodes on the page and the PDF and GitHub selection boxes appear.If the selection box is selected, a pop-up window will display the PDF file or GitHub file contained in the current page,and if no, it will appear the corresponding prompt.

(10) TreeMap Module: treemap visualization

Click the [TreeMap] button to display a treemap indicating the hierarchical status of the

relevant directories and files. The larger the value of the module, the larger the displayed rectangular block. When hovering over a rectangular block, the complete name and value information of the module will be shown. Additionally, upon a mouse-click operation, a window will pop up displaying the module's code and documentation. On the left side of the page, two different scales and two click buttons are displayed. Dragging the zoom scale adjusts the module sizes proportionally, while dragging the Layer progress bar deepens the color of nodes at that level on the map; the number on the right of the Layer progress bar indicates the module's hierarchy. Clicking the Exchange button switches the color of the rectangular blocks, with all descendants of the same node displaying one color. Clicking the Raw button toggles data size, allowing for switching between raw data and square root-processeddata.

At the same time, the left control area shows the number of nodes on the page and the PDF and GitHub selection boxes appear.If the selection box is selected, a pop-up window will display the PDF file or GitHub file contained in the current page,and if no, it will appear the corresponding prompt.

4- Download figures

On the left control region two download buttons are provided for obtaining the visualization figures if you need.

| ↓ SVG | ↓ PNG |

Thanks for your attention and suggestion.

2023.12.16