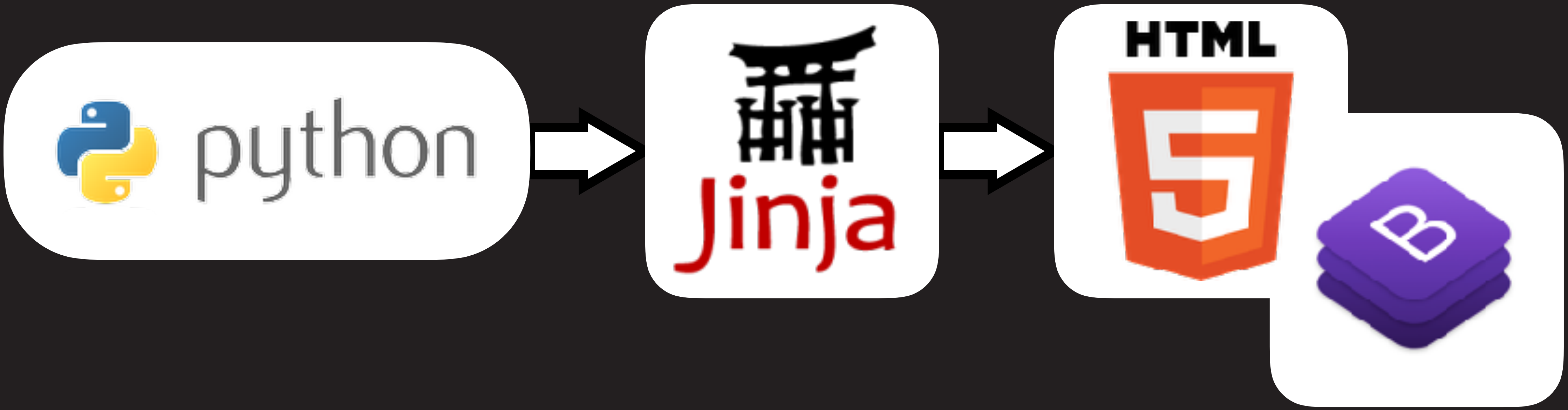
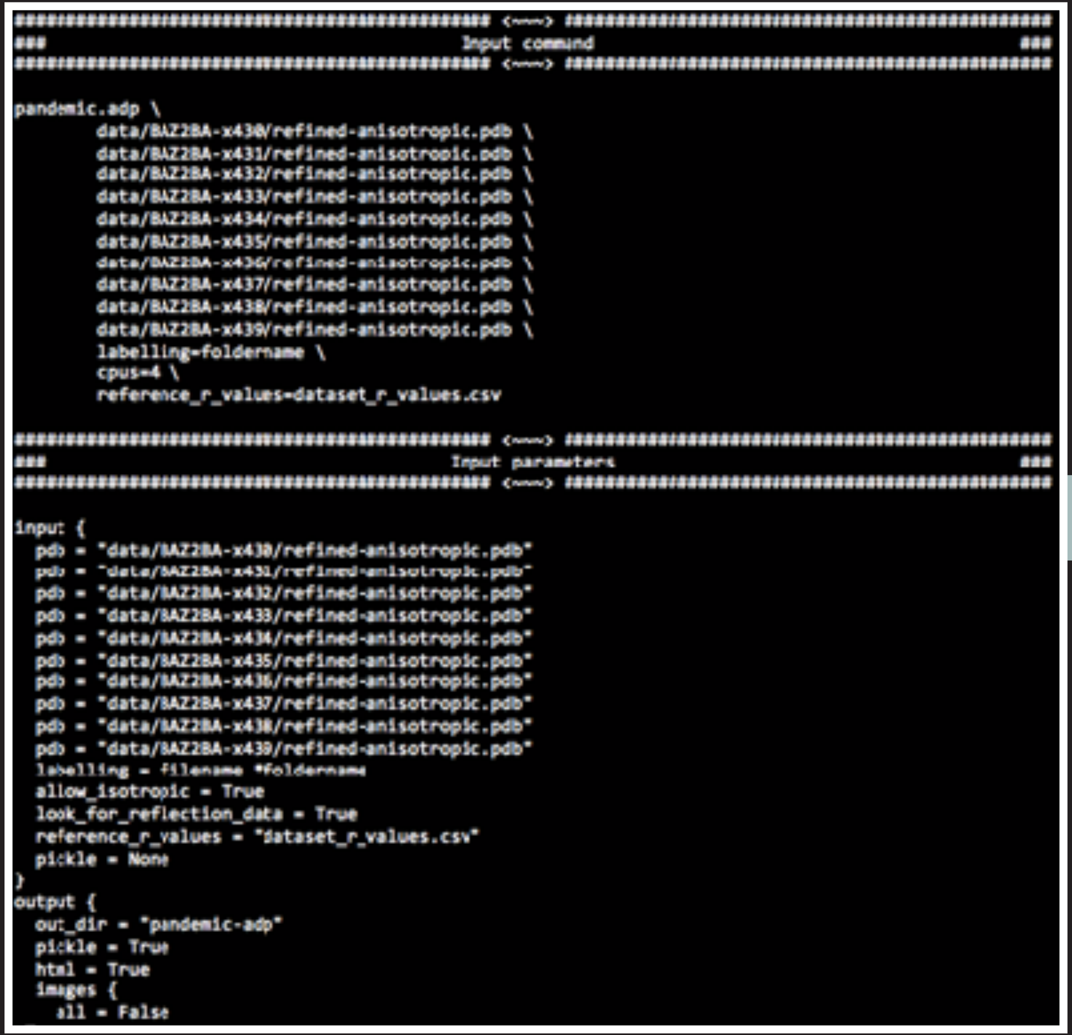
PRESENTING RESULTS TO USERS

Exporting from python → html with Jinja2



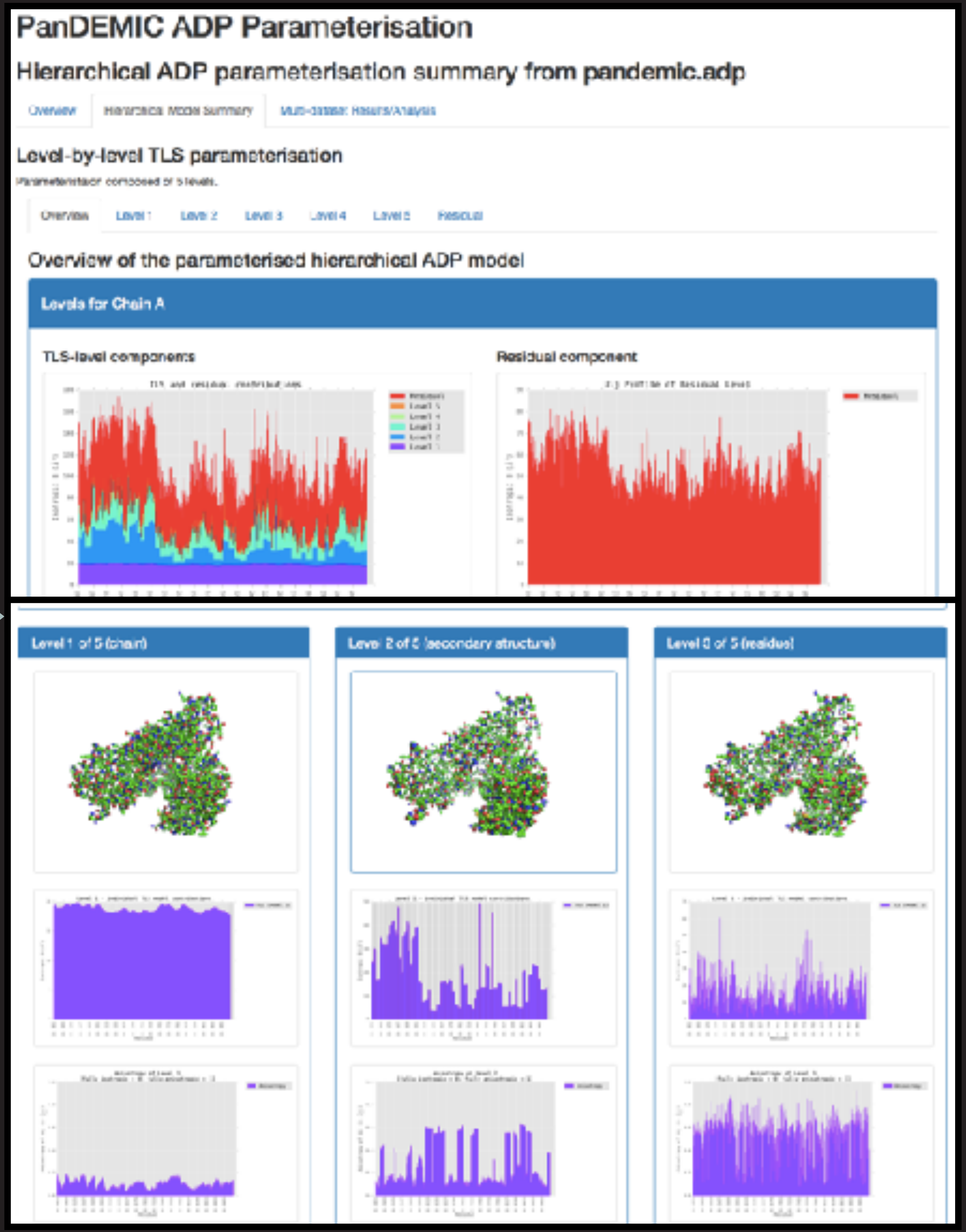
NICHOLAS PEARCE — UNIVERSITY OF UTRECHT

WHY?



```
############################################## <~~~> ##################################################
###                                         Input command                                            ###
############################################## <~~~> ##################################################

pandemic.adp \
        data/BAZ2BA-x430/refined-anisotropic.pdb \
        data/BAZ2BA-x431/refined-anisotropic.pdb \
        data/BAZ2BA-x432/refined-anisotropic.pdb \
        data/BAZ2BA-x433/refined-anisotropic.pdb \
        data/BAZ2BA-x434/refined-anisotropic.pdb \
        data/BAZ2BA-x435/refined-anisotropic.pdb \
        data/BAZ2BA-x436/refined-anisotropic.pdb \
        data/BAZ2BA-x437/refined-anisotropic.pdb \
        data/BAZ2BA-x438/refined-anisotropic.pdb \
        data/BAZ2BA-x439/refined-anisotropic.pdb \
        labelling=foldername \
        cpus=4 \
        reference_r_values=dataset_r_values.csv

############################################## <~~~> ##################################################
###                                        Input parameters                                          ###
############################################## <~~~> ##################################################

input {
  pdb = "data/BAZ2BA-x430/refined-anisotropic.pdb"
  pdb = "data/BAZ2BA-x431/refined-anisotropic.pdb"
  pdb = "data/BAZ2BA-x432/refined-anisotropic.pdb"
  pdb = "data/BAZ2BA-x433/refined-anisotropic.pdb"
  pdb = "data/BAZ2BA-x434/refined-anisotropic.pdb"
  pdb = "data/BAZ2BA-x435/refined-anisotropic.pdb"
  pdb = "data/BAZ2BA-x436/refined-anisotropic.pdb"
  pdb = "data/BAZ2BA-x437/refined-anisotropic.pdb"
  pdb = "data/BAZ2BA-x438/refined-anisotropic.pdb"
  pdb = "data/BAZ2BA-x439/refined-anisotropic.pdb"
  labelling = filename *foldername
  allow_isotropic = True
  look_for_reflection_data = True
  reference_r_values = "dataset_r_values.csv"
  pickle = None
}
output {
  out_dir = "pandemic-adp"
  pickle = True
  html = True
  images {
    all = False
```

LOG FILES ARE TERRIBLE



PanDEMIC ADP Parameterisation

Hierarchical ADP parameterisation summary from pandemic.adp

Overview | Hierarchical Model Summary | Multi-dataset Results/Analysis

Level-by-level TLS parameterisation

Parameterisation composed of 5 levels.

Overview | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 | Residual

Overview of the parameterised hierarchical ADP model

Levels for Chain A

TLS-level components

Residual component

Level 1 of 5 (chain)

Level 2 of 5 (secondary structure)

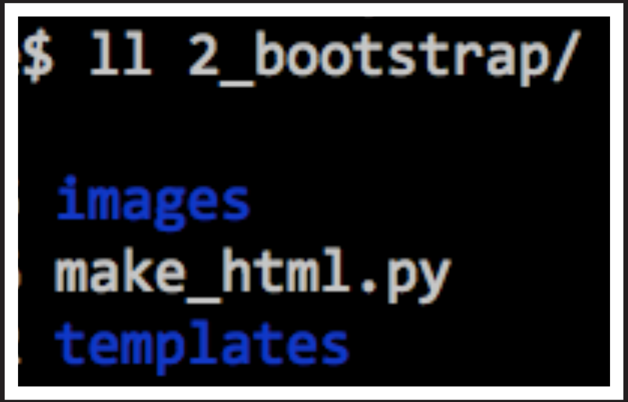
Level 5 of 5 (residue)

SETUP

- Requirements
  - python (2.7 or 3.6)
  - python packages: jinja2, lorem, pandas
- Documentation:
  - http://jinja.pocoo.org
  - https://getbootstrap.com/docs/3.3/
- Tutorials:
  - git clone https://github.com/nicholas-pearce/jinja2-tutorial.git

RUNNING THE EXAMPLES

- Simple!
  - python make_html.py
  - open "output.html" in web browser
  - (other required files contained in folders)



```
$ ll 2_bootstrap/

 images
 make_html.py
 templates
```

- Tutorials:
  - git clone https://github.com/nicholas-pearce/jinja2-tutorial.git

TEMPLATING WITH JINJA2

A SIMPLE TEMPLATE

```
This is a jinja2 template. It gets populated with {{ something }}.
Anything between the curly brackets gets processed by jinja2. This
allows you to write the webpage outline, and then fill it directly
from {{ somewhere }}.
```

```
>> render(something=“text or numbers”, somewhere=“your program”)
```

```
This is a jinja2 template. It gets populated with text or numbers.
Anything between the curly brackets gets processed by jinja2. This
allows you to write the webpage outline, and then fill it directly
from your program.
```

TEMPLATING WITH JINJA2

POPULATING TEMPLATES

```
>>> from jinja2 import Template
>>> template = Template('Hello {{ name }}!')
>>> template.render({'name':'Steve', …})
u'Hello Steve!’
>>> template.render(name='Steve', …)
u'Hello Steve!'
```

SHORTCUTS & STATEMENTS

```
{{ value }}
```

➤ dumps “value” if defined or nothing

```
{{ value | default(10) }}
```

➤ dumps “value” if defined, otherwise “10”

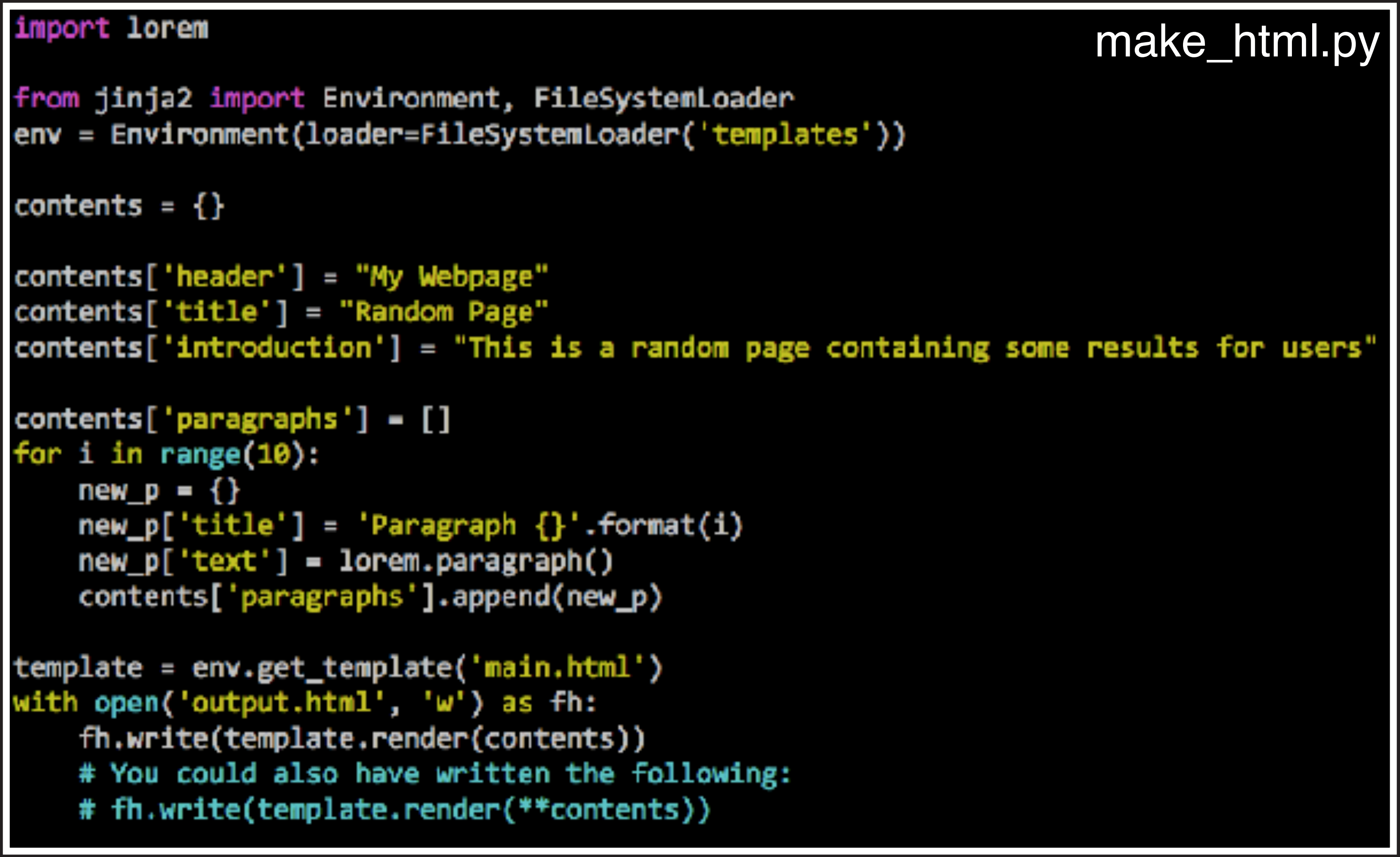
```
{% for v in value %}
  {{ v }}
{% endfor %}
```

➤ for loop

```
{% if value is defined %}
  {{ value }}
{% endif %}
```

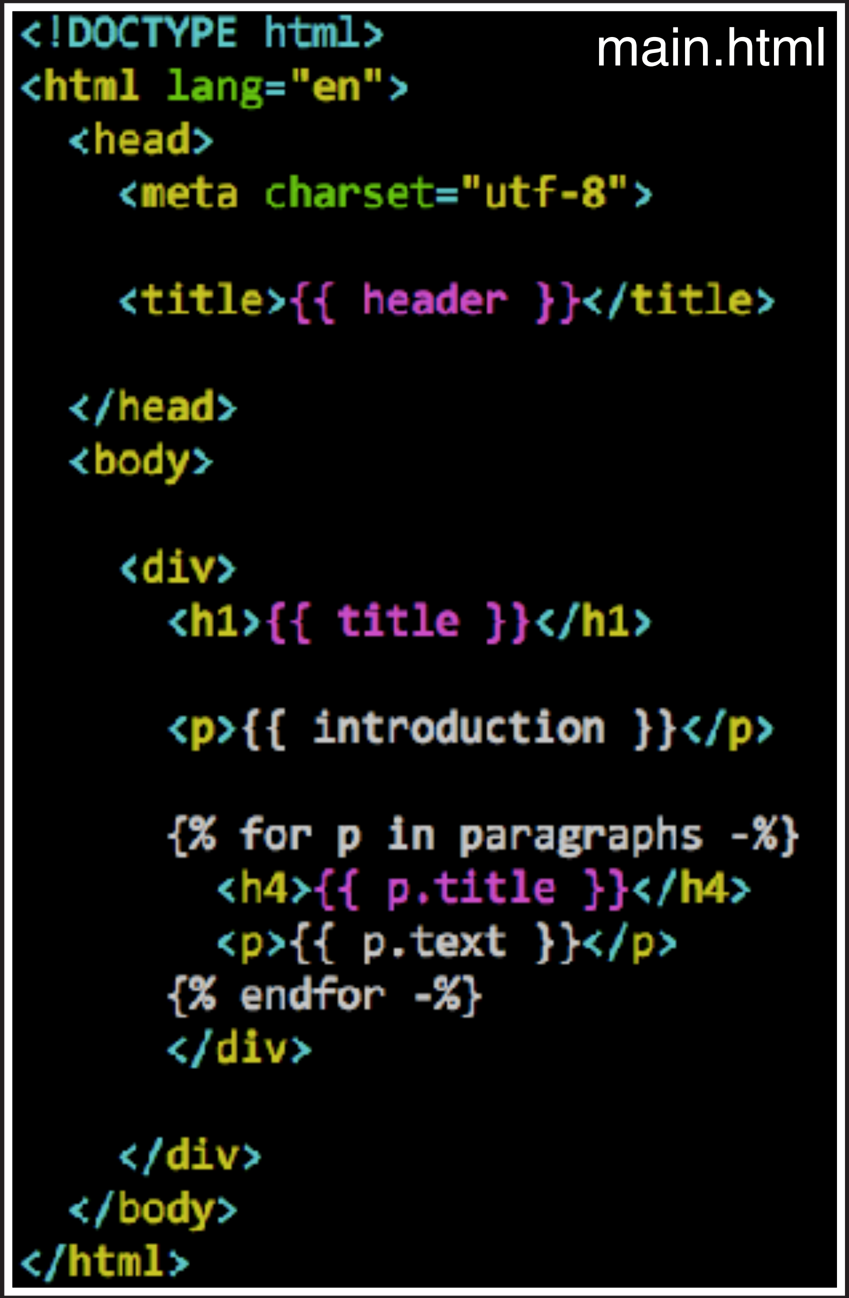
➤ if statement

EXAMPLE 1 – BASICS



make_html.py

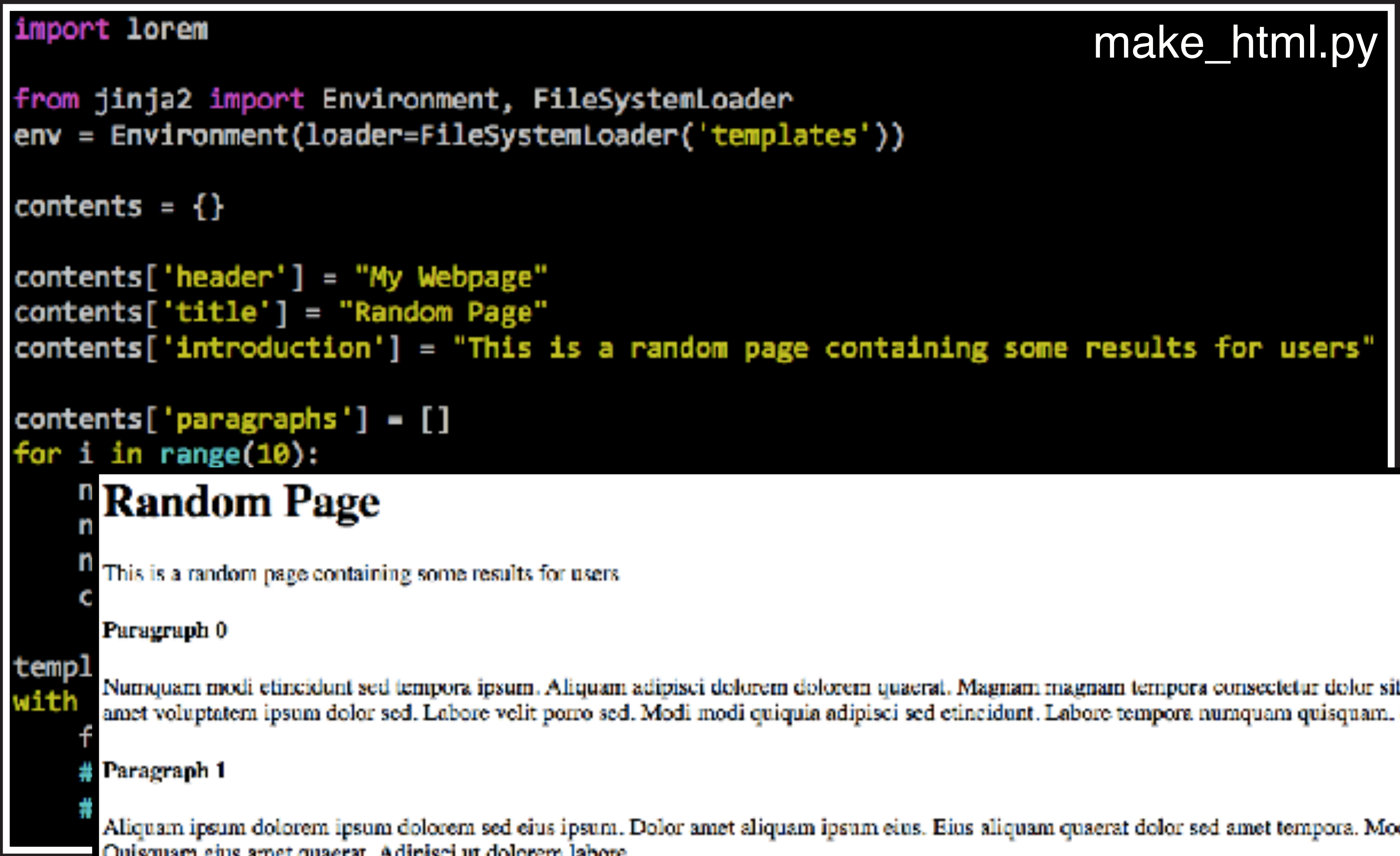
```python
import lorem

from jinja2 import Environment, FileSystemLoader
env = Environment(loader=FileSystemLoader('templates'))

contents = {}

contents['header'] = "My Webpage"
contents['title'] = "Random Page"
contents['introduction'] = "This is a random page containing some results for users"

contents['paragraphs'] = []
for i in range(10):
    new_p = {}
    new_p['title'] = 'Paragraph {}'.format(i)
    new_p['text'] = lorem.paragraph()
    contents['paragraphs'].append(new_p)

template = env.get_template('main.html')
with open('output.html', 'w') as fh:
    fh.write(template.render(contents))
    # You could also have written the following:
    # fh.write(template.render(**contents))
```



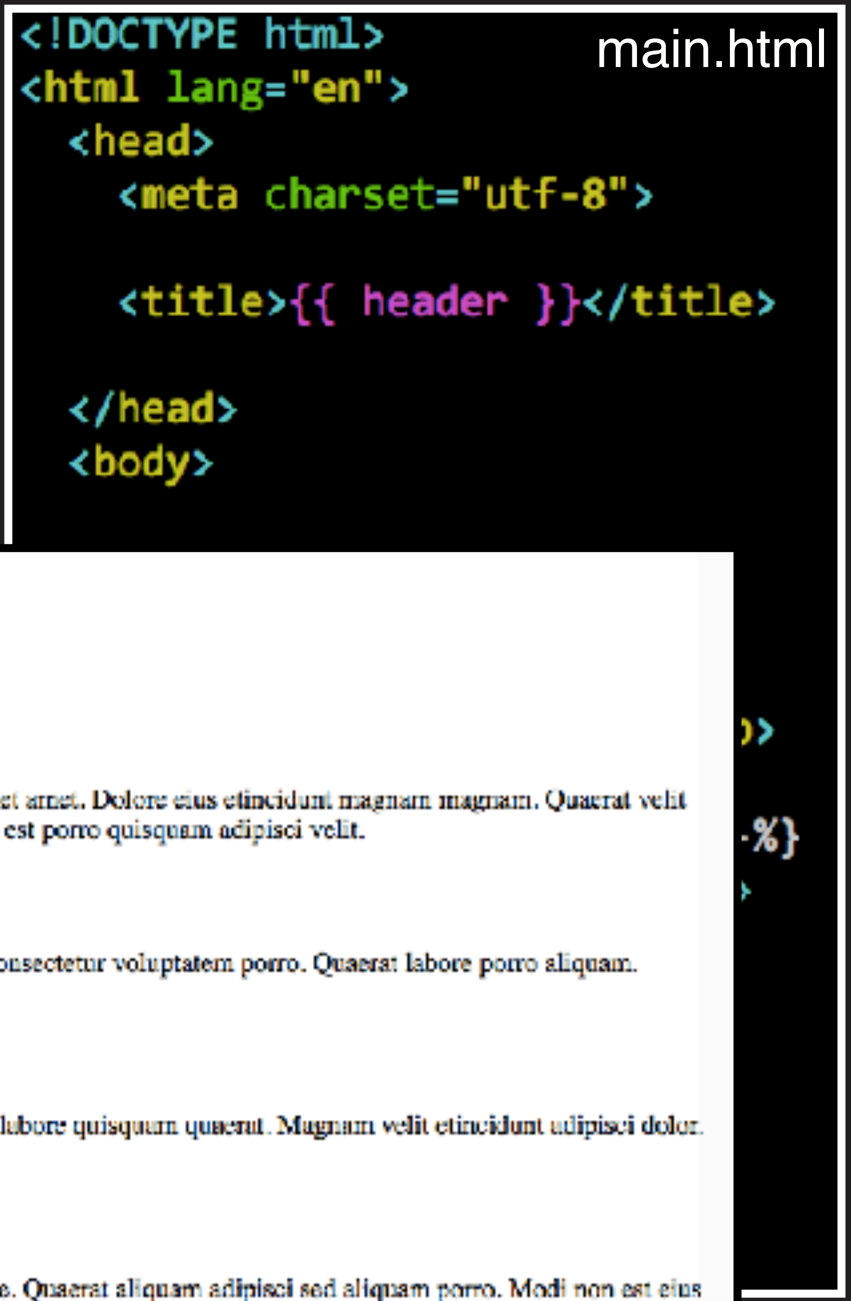
main.html

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">

    <title>{{ header }}</title>

  </head>
  <body>

    <div>
      <h1>{{ title }}</h1>

      <p>{{ introduction }}</p>

      {% for p in paragraphs -%}
        <h4>{{ p.title }}</h4>
        <p>{{ p.text }}</p>
      {% endfor -%}
      </div>

    </div>
  </body>
</html>
```

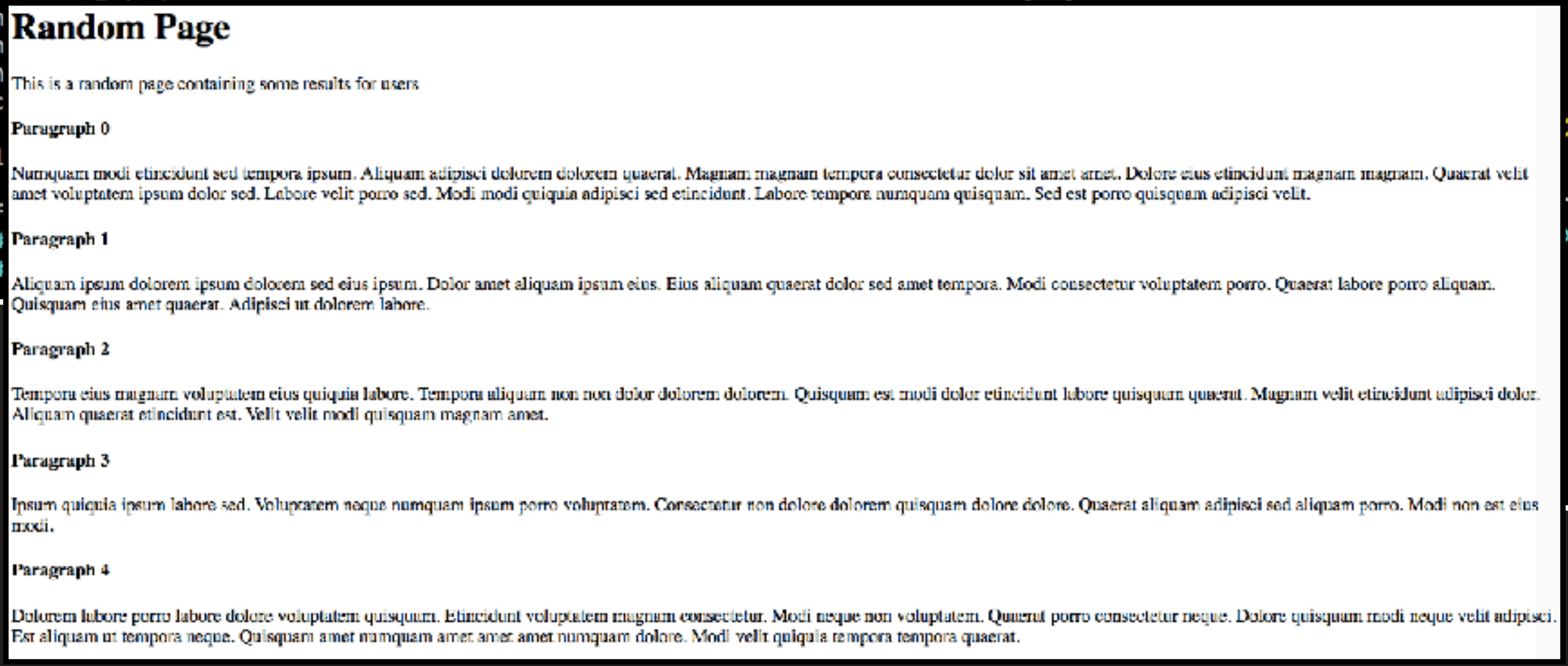
EXAMPLE 1 – BASICS


make_html.py

```
import lorem

from jinja2 import Environment, FileSystemLoader
env = Environment(loader=FileSystemLoader('templates'))

contents = {}

contents['header'] = "My Webpage"
contents['title'] = "Random Page"
contents['introduction'] = "This is a random page containing some results for users"

contents['paragraphs'] = []
for i in range(10):
    n
    n
    n
    c

templ
with
    f
    #
    #
```


main.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">

    <title>{{ header }}</title>

  </head>
  <body>
```



Random Page

This is a random page containing some results for users

Paragraph 0

Numquam modi etincidunt sed tempora ipsum. Aliquam adipisci dolorem dolorem quaerat. Magnam magnam tempora consectetur dolor sit amet amet. Dolore eius etincidunt magnam magnam. Quaerat velit amet voluptatem ipsum dolor sed. Labore velit porro sed. Modi modi quiquia adipisci sed etincidunt. Labore tempora numquam quisquam. Sed est porro quisquam adipisci velit.

Paragraph 1

Aliquam ipsum dolorem ipsum dolorem sed eius ipsum. Dolor amet aliquam ipsum eius. Eius aliquam quaerat dolor sed amet tempora. Modi consectetur voluptatem porro. Quaerat labore porro aliquam. Quisquam eius amet quaerat. Adipisci ut dolorem labore.

Paragraph 2

Tempora eius magnam voluptatem eius quiquia labore. Tempora aliquam non non dolor dolorem dolorem. Quisquam est modi dolor etincidunt labore quisquam quaerat. Magnam velit etincidunt adipisci dolor. Aliquam quaerat etincidunt est. Velit velit modi quisquam magnam amet.

Paragraph 3

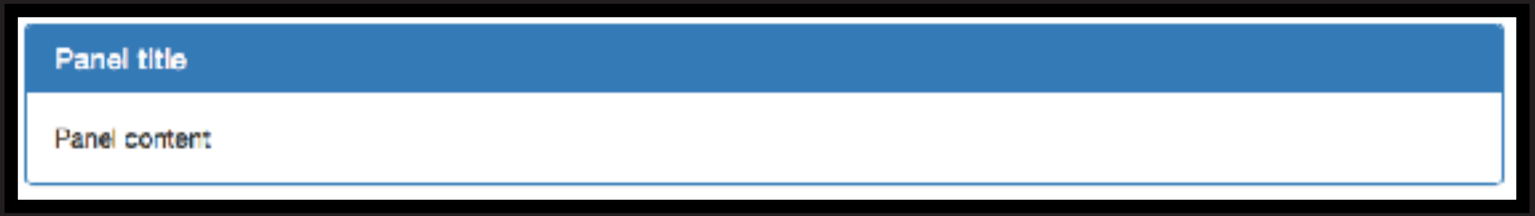
Ipsum quiquia ipsum labore sed. Voluptatem neque numquam ipsum porro voluptatem. Consectetur non dolore dolorem quisquam dolore dolore. Quaerat aliquam adipisci sed aliquam porro. Modi non est eius modi.

Paragraph 4

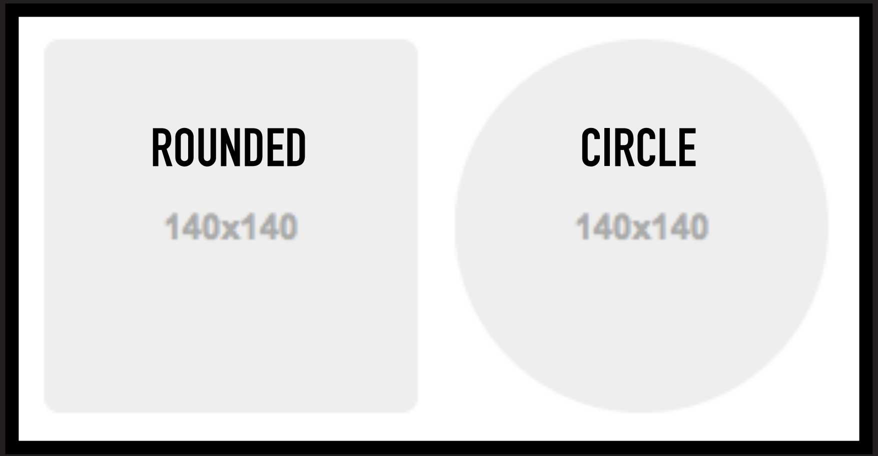
Dolorem labore porro labore dolore voluptatem quisquam. Etincidunt voluptatem magnam consectetur. Modi neque non voluptatem. Quaerat porro consectetur neque. Dolore quisquam modi neque velit adipisci. Est aliquam ut tempora neque. Quisquam amet numquam amet amet amet numquam dolore. Modi velit quiquia tempora tempora quaerat.

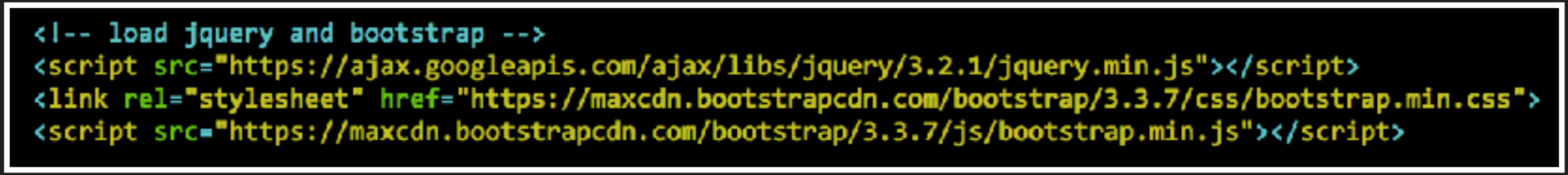
BOOTSTRAP

- html, css & js library
- makes pretty websites very easily



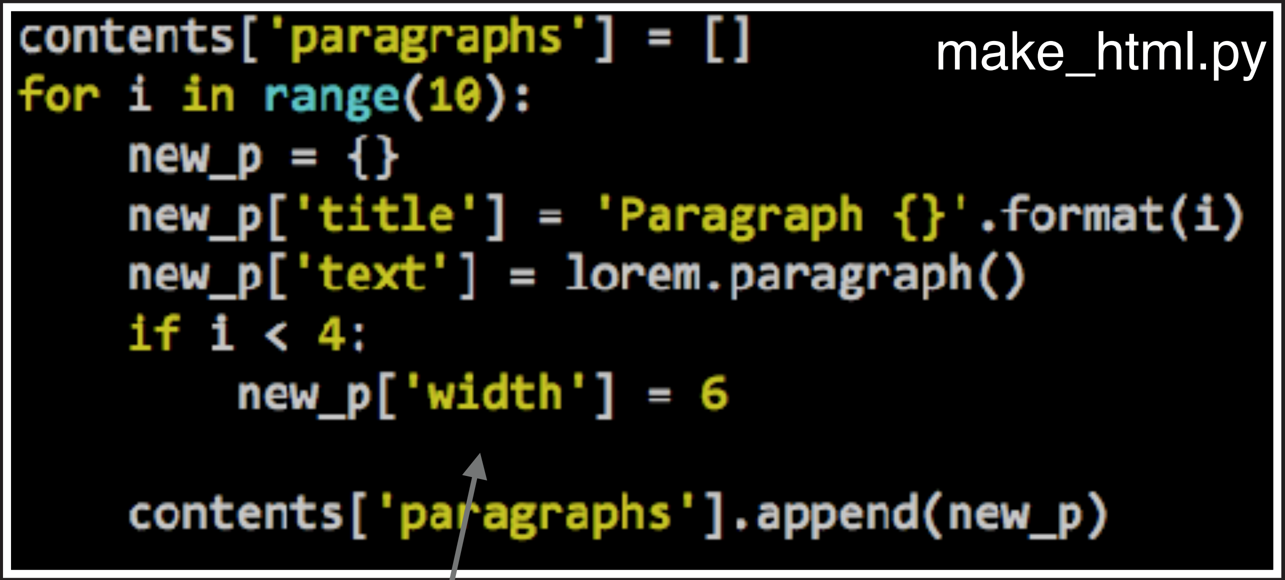
PANELS & IMAGES





```
<!-- load jquery and bootstrap -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```

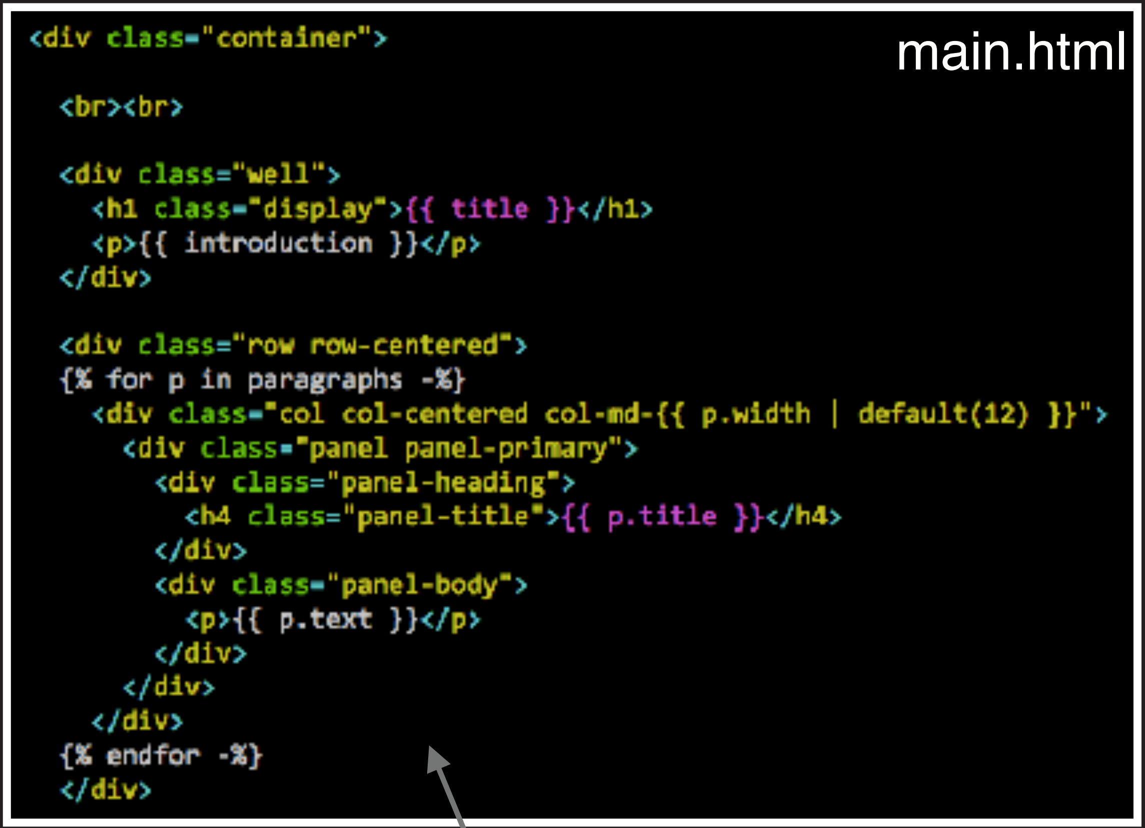
EXAMPLE 2- BOOTSTRAP



make_html.py

```python
contents['paragraphs'] = []
for i in range(10):
    new_p = {}
    new_p['title'] = 'Paragraph {}'.format(i)
    new_p['text'] = lorem.paragraph()
    if i < 4:
        new_p['width'] = 6

    contents['paragraphs'].append(new_p)
```

add widths to
some paragraphs

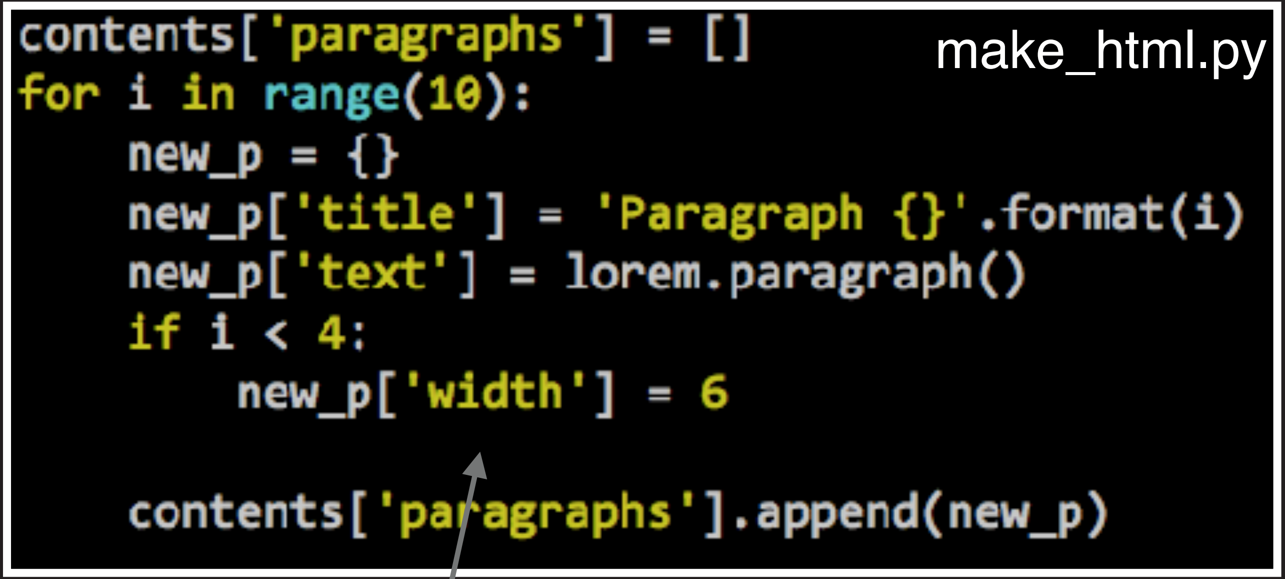


main.html

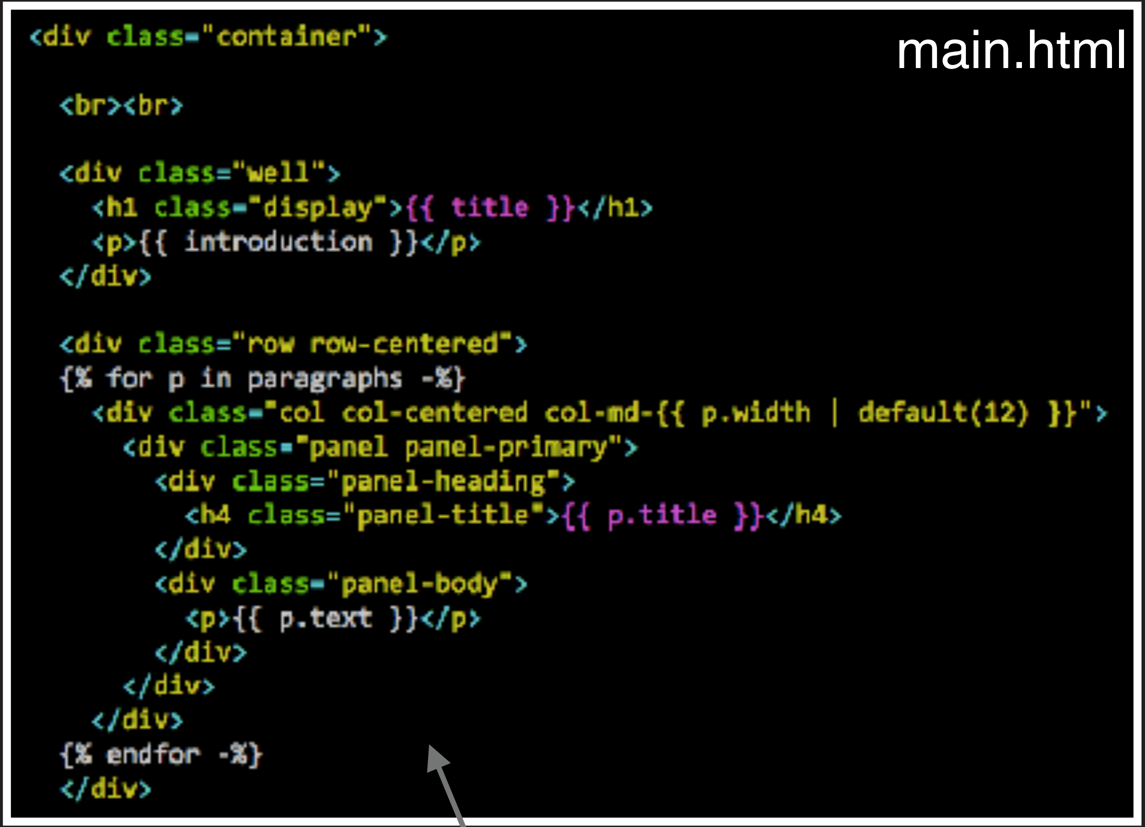
```html
<div class="container">

  <br><br>

  <div class="well">
    <h1 class="display">{{ title }}</h1>
    <p>{{ introduction }}</p>
  </div>

  <div class="row row-centered">
  {% for p in paragraphs -%}
    <div class="col col-centered col-md-{{ p.width | default(12) }}">
      <div class="panel panel-primary">
        <div class="panel-heading">
          <h4 class="panel-title">{{ p.title }}</h4>
        </div>
        <div class="panel-body">
          <p>{{ p.text }}</p>
        </div>
      </div>
    </div>
  {% endfor -%}
  </div>
```

add panel for
each paragraph

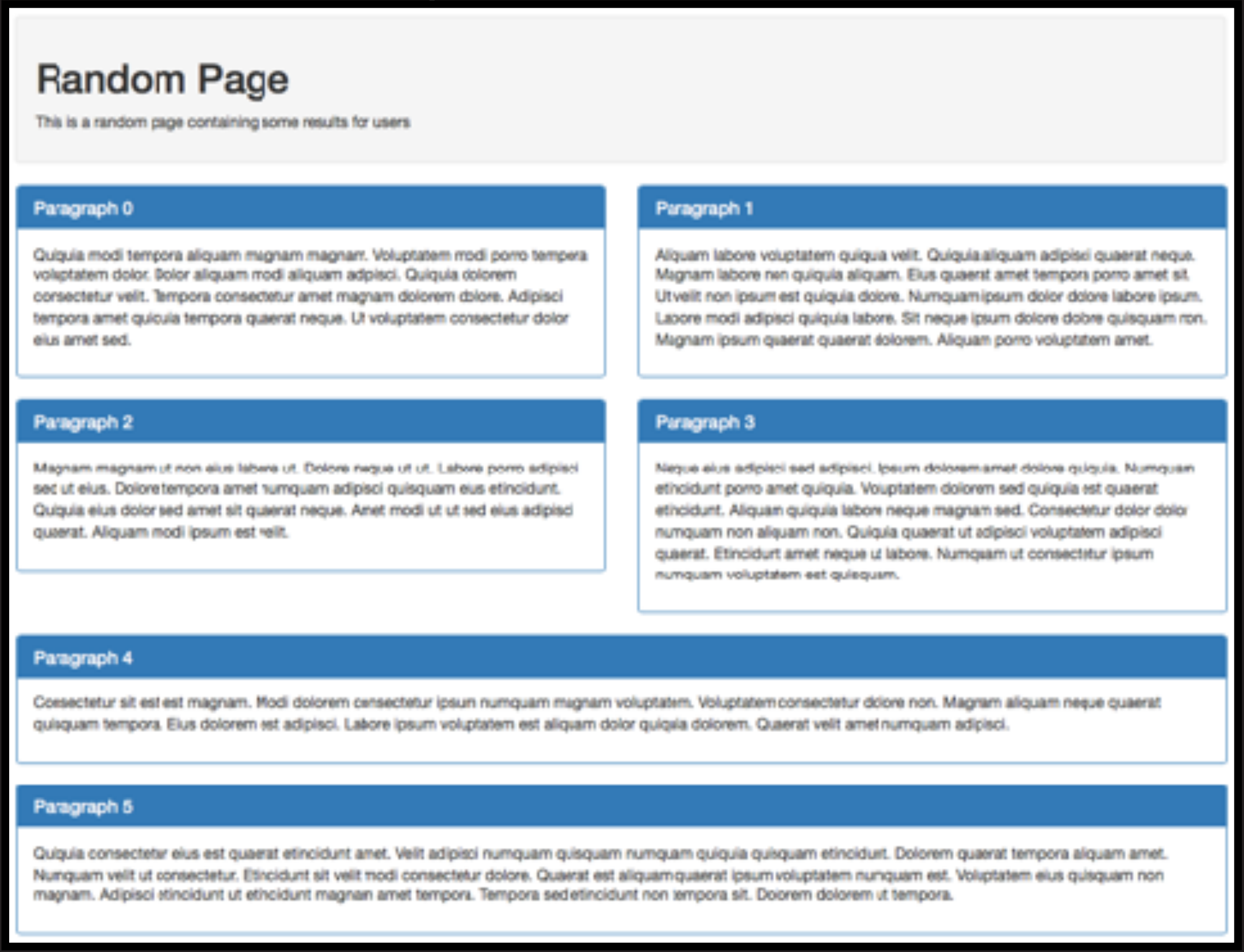
EXAMPLE 2- BOOTSTRAP


make_html.py

```python
contents['paragraphs'] = []
for i in range(10):
    new_p = {}
    new_p['title'] = 'Paragraph {}'.format(i)
    new_p['text'] = lorem.paragraph()
    if i < 4:
        new_p['width'] = 6

    contents['paragraphs'].append(new_p)
```


main.html

```html
<div class="container">

  <br><br>

  <div class="well">
    <h1 class="display">{{ title }}</h1>
    <p>{{ introduction }}</p>
  </div>

  <div class="row row-centered">
  {% for p in paragraphs -%}
    <div class="col col-centered col-md-{{ p.width | default(12) }}">
      <div class="panel panel-primary">
        <div class="panel-heading">
          <h4 class="panel-title">{{ p.title }}</h4>
        </div>
        <div class="panel-body">
          <p>{{ p.text }}</p>
        </div>
      </div>
    </div>
  {% endfor -%}
  </div>
```


Random Page

This is a random page containing some results for users

Paragraph 0

Quiquia modi tempora aliquam magnam magnam. Voluptatem modi porro tempera voluptatem dolor. Dolor aliquam modi aliquam adipisci. Quiquia dolorem consectetur velit. Tempora consectetur amet magnam dolorem dolore. Adipisci tempora amet quicuia tempora quaerat neque. Ut voluptatem consectetur dolor eius amet sed.

Paragraph 1

Aliquam labore voluptatem quiqua velit. Quiquia aliquam adipisci quaerat neque. Magnam labore non quiquia aliquam. Eius quaerat amet tempora porro amet sit. Ut velit non ipsum est quiquia dolore. Numquam ipsum dolor dolore labore ipsum. Labore modi adipisci quiquia labore. Sit neque ipsum dolore dolore quisquam non. Magnam ipsum quaerat quaerat dolorem. Aliquam porro voluptatem amet.

Paragraph 2

Magnam magnam ut non eius labore ut. Dolore neque ut ut. Labore porro adipisci sed ut eius. Dolore tempora amet numquam adipisci quisquam eius etincidunt. Quiquia eius dolor sed amet sit quaerat neque. Amet modi ut ut sed eius adipisci quaerat. Aliquam modi ipsum est velit.

Paragraph 3

Neque eius adipisci sed adipisci. Ipsum dolorem amet dolore quiquia. Numquam etincidunt porro amet quiquia. Voluptatem dolorem sed quiquia est quaerat etincidunt. Aliquam quiquia labore neque magnam sed. Consectetur dolor dolor numquam non aliquam non. Quiquia quaerat ut adipisci voluptatem adipisci quaerat. Etincidunt amet neque ut labore. Numquam ut consectetur ipsum numquam voluptatem est quisquam.

Paragraph 4

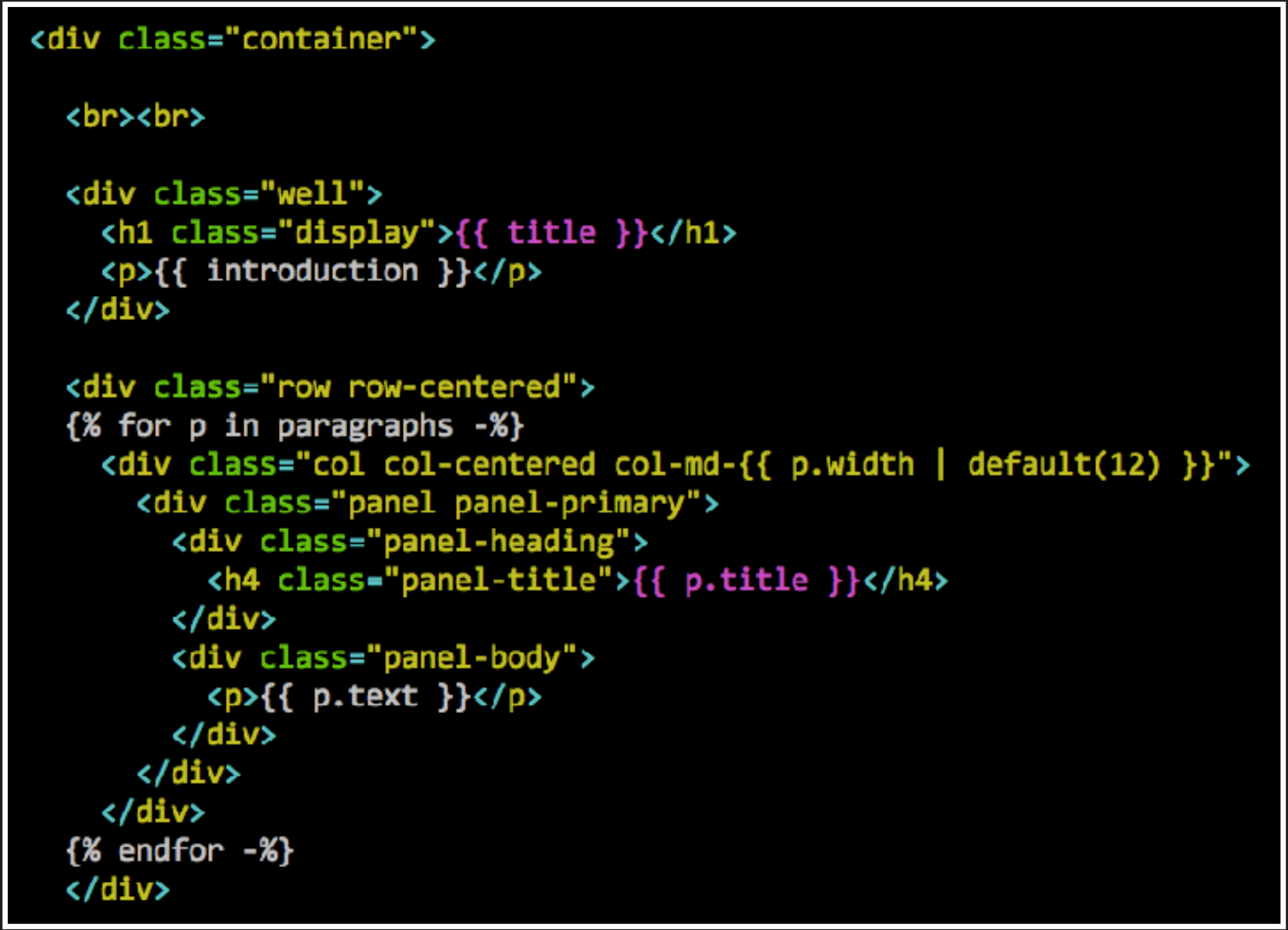
Consectetur sit est est magnam. Modi dolorem consectetur ipsum numquam magnam voluptatem. Voluptatem consectetur dolore non. Magnam aliquam neque quaerat quisquam tempora. Eius dolorem est adipisci. Labore ipsum voluptatem est aliquam dolor quiquia dolorem. Quaerat velit amet numquam adipisci.

Paragraph 5

Quiquia consectetur eius est quaerat etincidunt amet. Velit adipisci numquam quisquam numquam quiquia quisquam etincidunt. Dolorem quaerat tempora aliquam amet. Numquam velit ut consectetur. Etincidunt sit velit modi consectetur dolore. Quaerat est aliquam quaerat ipsum voluptatem numquam est. Voluptatem eius quisquam non magnam. Adipisci etincidunt ut etincidunt magnam amet tempora. Tempora sed etincidunt non tempora sit. Dolorem dolorem ut tempora.

add panel for each paragraph

EXAMPLE 2- BOOTSTRAP



```
<div class="container">

  <br><br>

  <div class="well">
    <h1 class="display">{{ title }}</h1>
    <p>{{ introduction }}</p>
  </div>

  <div class="row row-centered">
  {% for p in paragraphs -%}
    <div class="col col-centered col-md-{{ p.width | default(12) }}">
      <div class="panel panel-primary">
        <div class="panel-heading">
          <h4 class="panel-title">{{ p.title }}</h4>
        </div>
        <div class="panel-body">
          <p>{{ p.text }}</p>
        </div>
      </div>
    </div>
  {% endfor -%}
  </div>
```

FUNCTIONALISATION: MACROS

- “macros” allow you to use pre-defined functions

```
{% macro dump_string(s) %}
  {{ s }}
{% endmacro %}
```

DEFINE MACRO

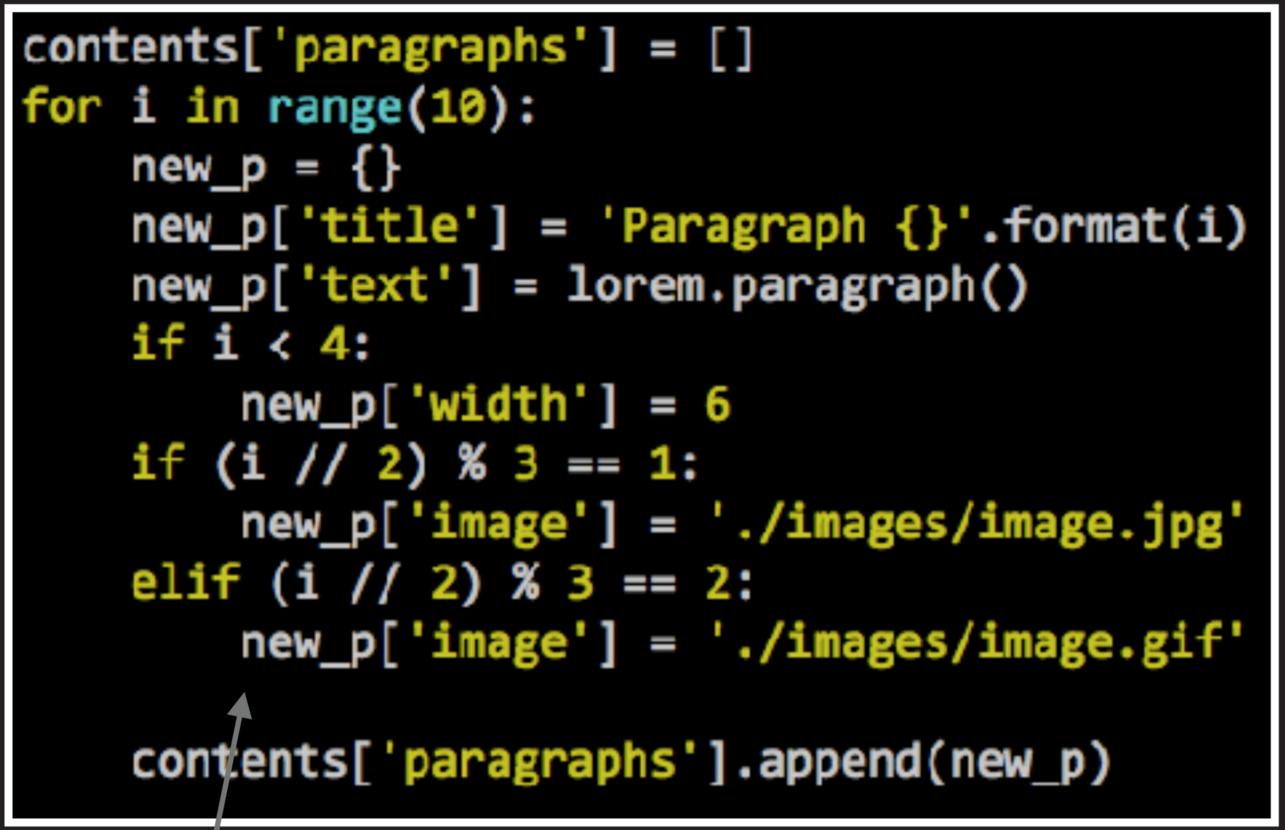
```
{% from ‘macros.html’ import dump_string %}
```

IMPORT MACRO

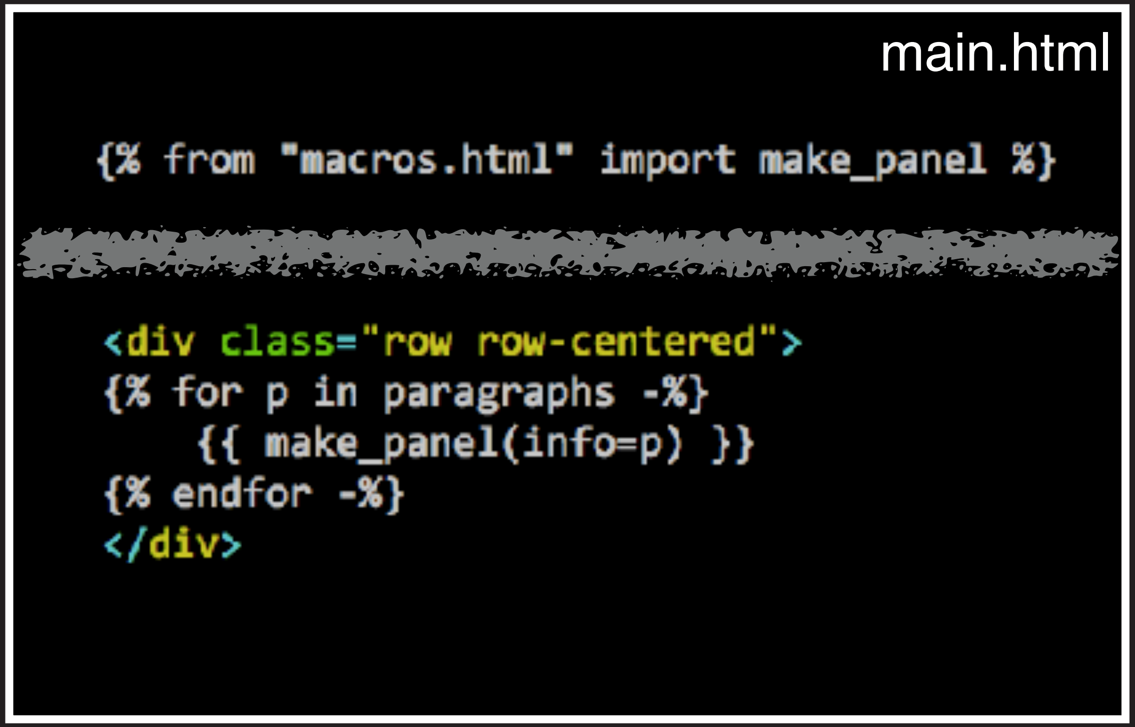
```
{{ dump_string(‘interesting string’) }}
```

USE MACRO

EXAMPLE 3 – BOOTSTRAP WITH MACROS

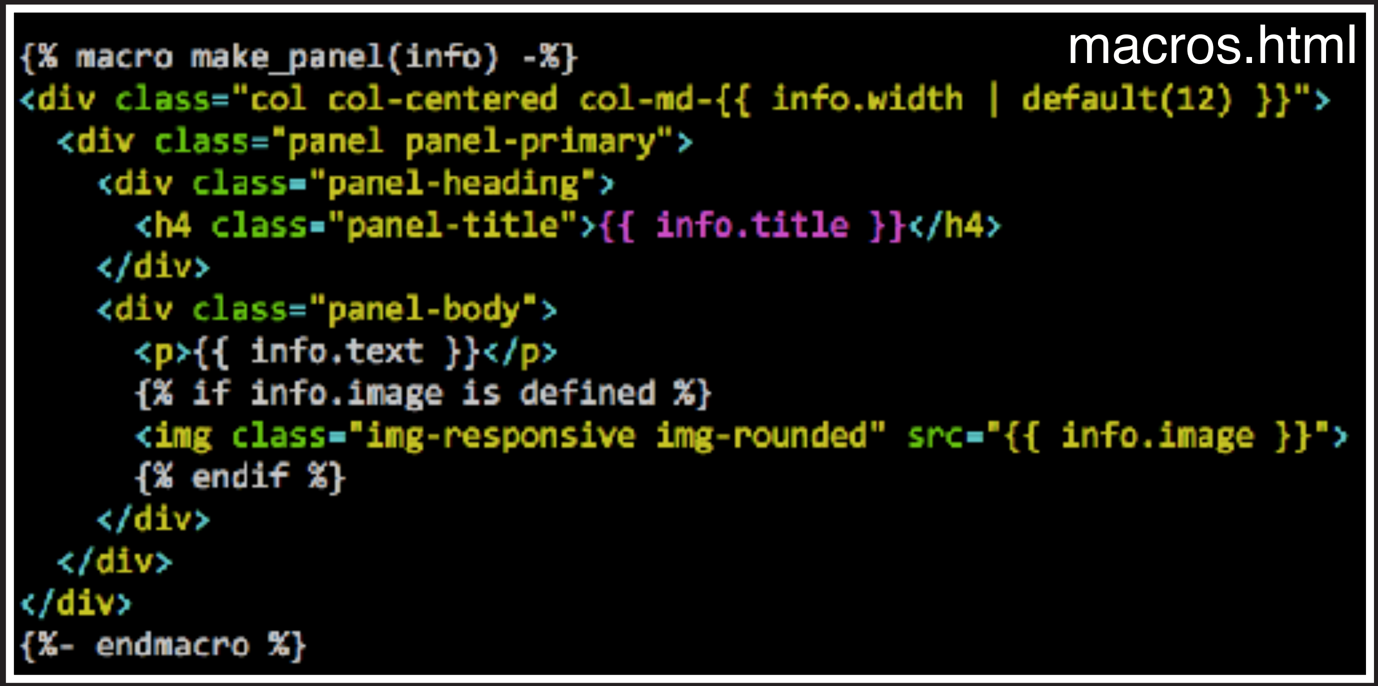


```python
contents['paragraphs'] = []
for i in range(10):
    new_p = {}
    new_p['title'] = 'Paragraph {}'.format(i)
    new_p['text'] = lorem.paragraph()
    if i < 4:
        new_p['width'] = 6
    if (i // 2) % 3 == 1:
        new_p['image'] = './images/image.jpg'
    elif (i // 2) % 3 == 2:
        new_p['image'] = './images/image.gif'

    contents['paragraphs'].append(new_p)
```



main.html

```html
{% from "macros.html" import make_panel %}

<div class="row row-centered">
{% for p in paragraphs -%}
    {{ make_panel(info=p) }}
{% endfor -%}
</div>
```

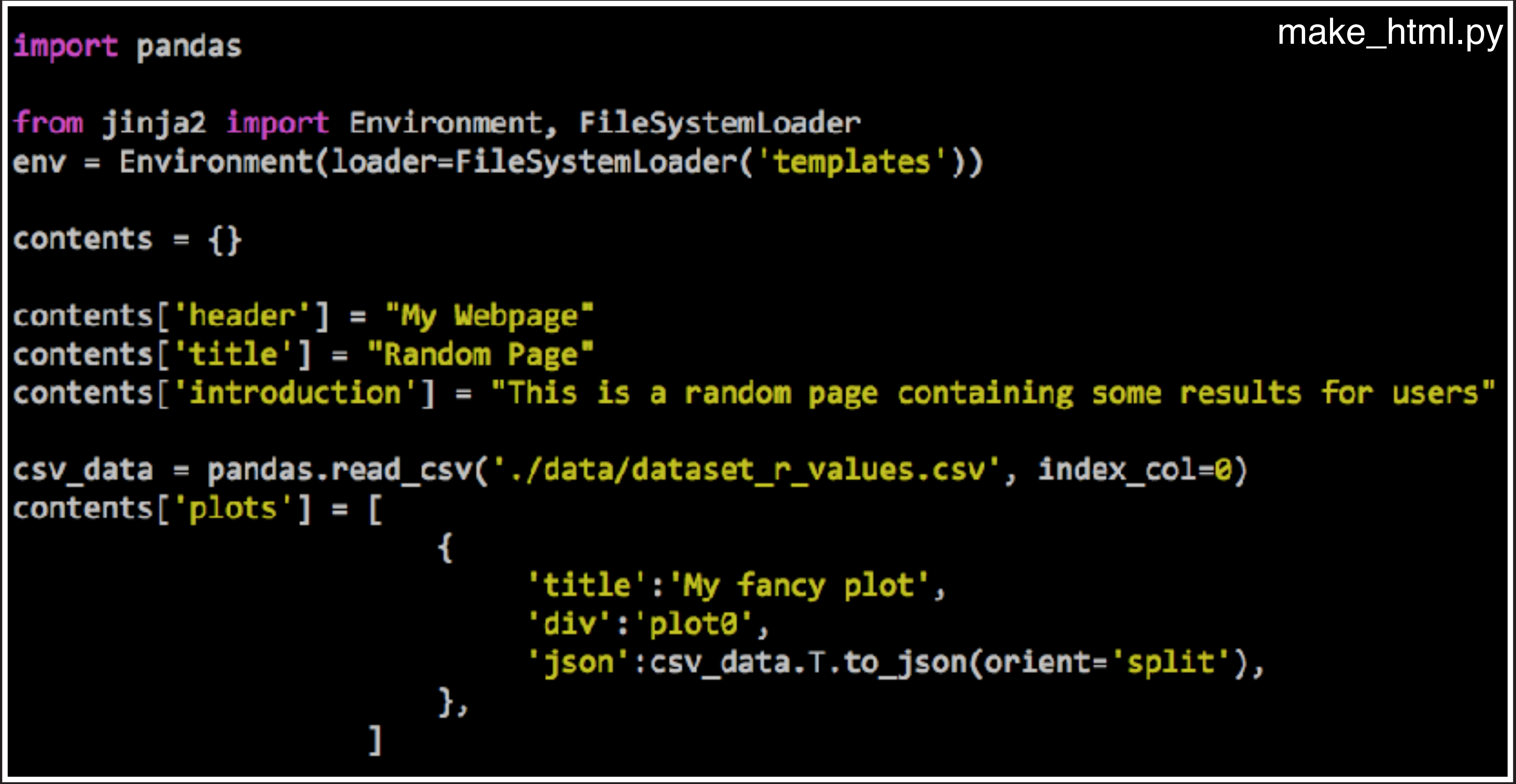


macros.html

```html
{% macro make_panel(info) -%}
<div class="col col-centered col-md-{{ info.width | default(12) }}">
  <div class="panel panel-primary">
    <div class="panel-heading">
      <h4 class="panel-title">{{ info.title }}</h4>
    </div>
    <div class="panel-body">
      <p>{{ info.text }}</p>
      {% if info.image is defined %}
      <img class="img-responsive img-rounded" src="{{ info.image }}">
      {% endif %}
    </div>
  </div>
</div>
{%- endmacro %}
```

add images to
some panels

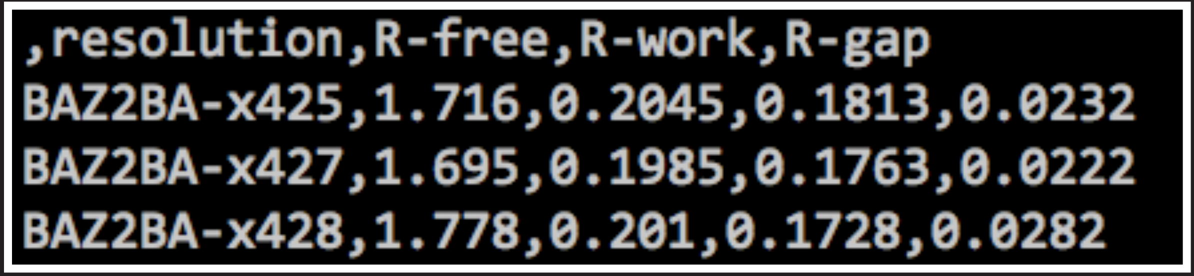
EXAMPLE 4 – PLOTS!



make_html.py

```
import pandas

from jinja2 import Environment, FileSystemLoader
env = Environment(loader=FileSystemLoader('templates'))

contents = {}

contents['header'] = "My Webpage"
contents['title'] = "Random Page"
contents['introduction'] = "This is a random page containing some results for users"

csv_data = pandas.read_csv('./data/dataset_r_values.csv', index_col=0)
contents['plots'] = [
                        {
                                'title':'My fancy plot',
                                'div':'plot0',
                                'json':csv_data.T.to_json(orient='split'),
                        },
                    ]
```

DATA:



```
,resolution,R-free,R-work,R-gap
BAZ2BA-x425,1.716,0.2045,0.1813,0.0232
BAZ2BA-x427,1.695,0.1985,0.1763,0.0222
BAZ2BA-x428,1.778,0.201,0.1728,0.0282
```

EXAMPLE 4 – PLOTS, THE DIFFICULT BIT…

```
{% from "plots.html" import make_plot, default_scripts, plotly_scripts %}
```

```
{{ default_scripts() }}
{{ plotly_scripts() }}
```

load libraries

```
  {% for p in plots -%}
    {{ make_plot(p) }}
  {% endfor %}
```

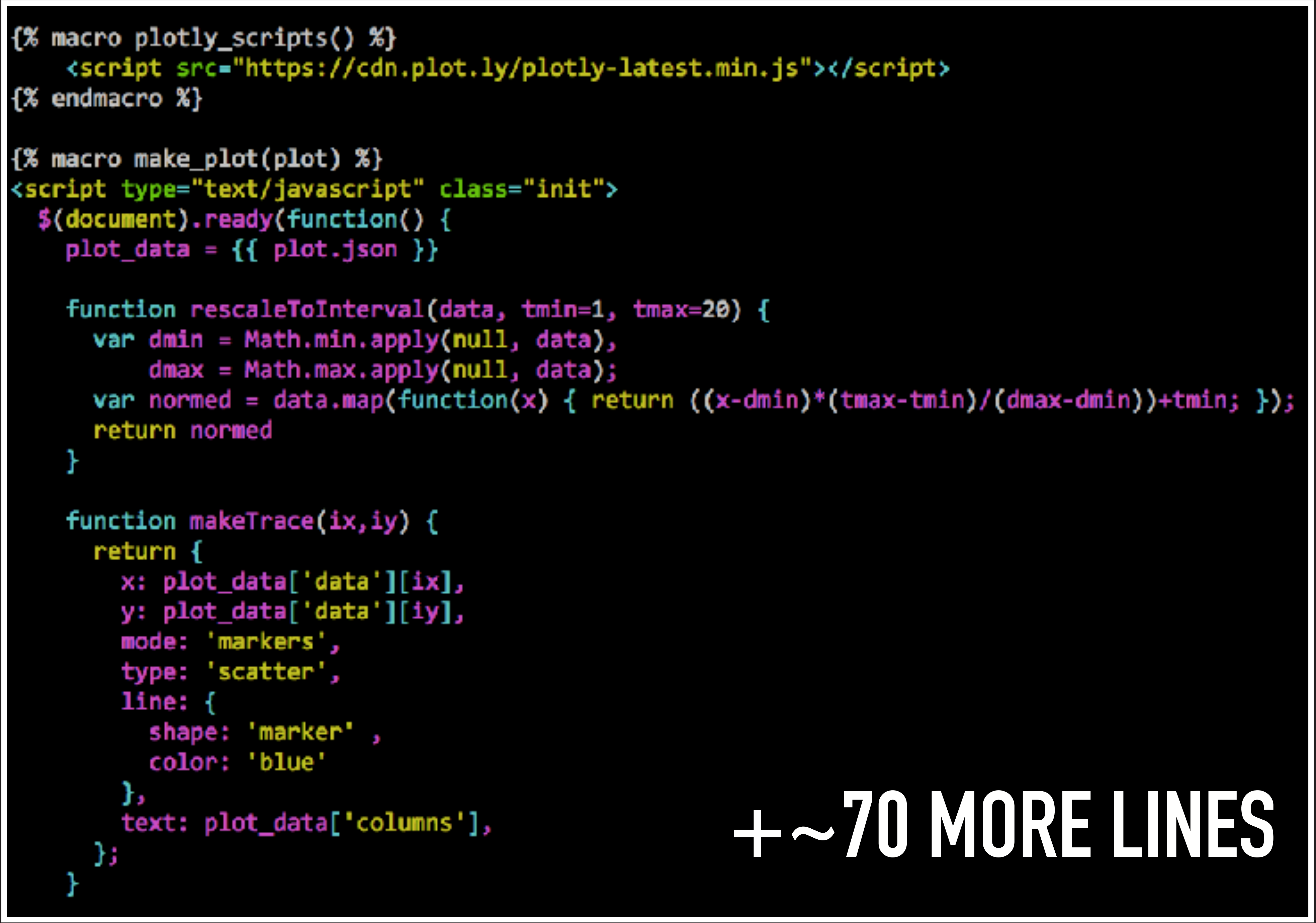
javascript

main.html

```
    <div class="row row-centered">
    {% for p in plots -%}
      <div class="col col-xs-12">
        <h1>{{ p.title }}</h1>
        <div id="{{ p.div }}">
        </div>
      </div>
    {% endfor -%}
    </div>
```

create blank div

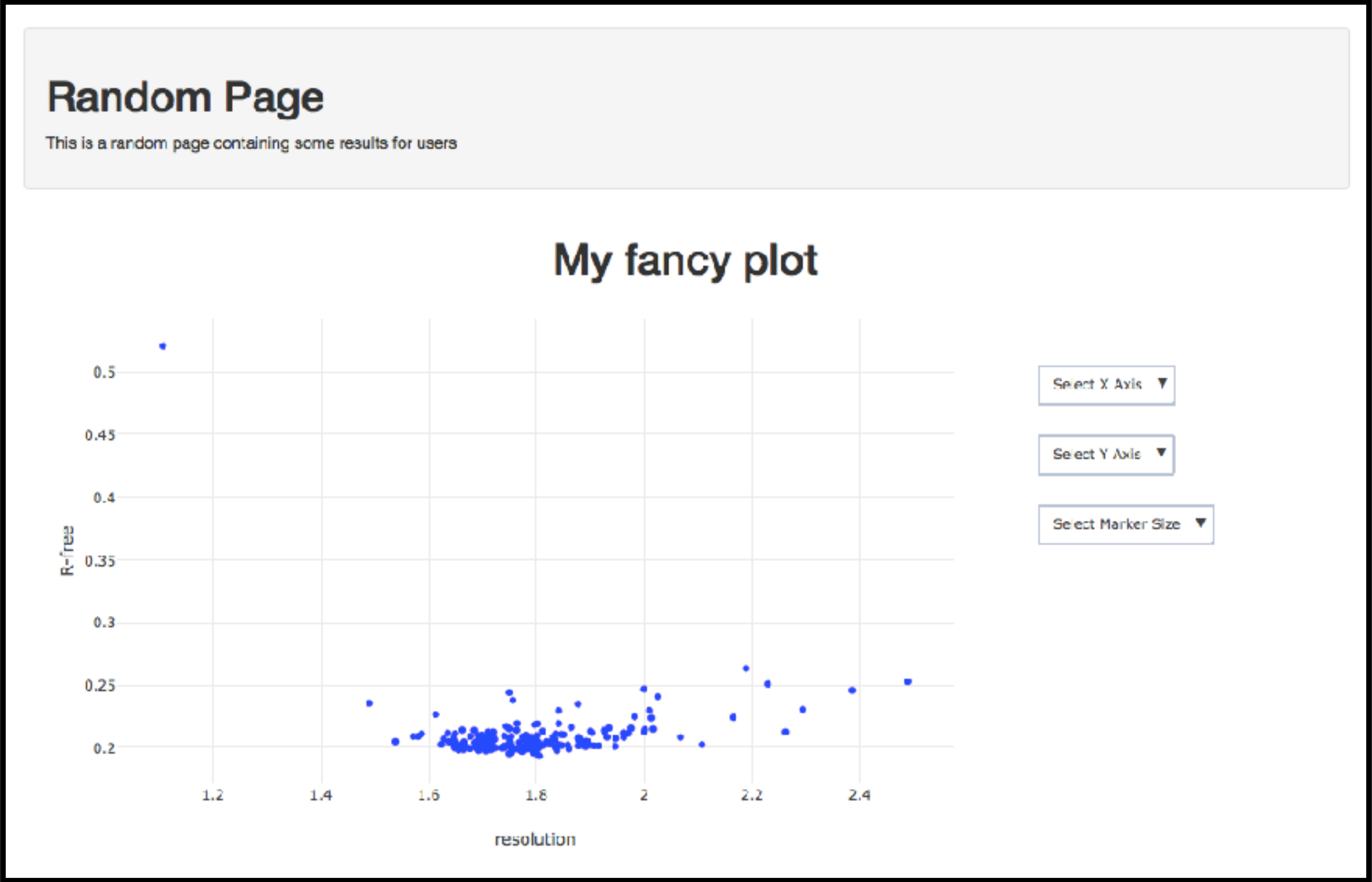
EXAMPLE 4 – PLOTS, THE UGLY BIT...



```
{% macro plotly_scripts() %}
    <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
{% endmacro %}

{% macro make_plot(plot) %}
<script type="text/javascript" class="init">
  $(document).ready(function() {
    plot_data = {{ plot.json }}

    function rescaleToInterval(data, tmin=1, tmax=20) {
      var dmin = Math.min.apply(null, data),
          dmax = Math.max.apply(null, data);
      var normed = data.map(function(x) { return ((x-dmin)*(tmax-tmin)/(dmax-dmin))+tmin; });
      return normed
    }

    function makeTrace(ix,iy) {
      return {
        x: plot_data['data'][ix],
        y: plot_data['data'][iy],
        mode: 'markers',
        type: 'scatter',
        line: {
          shape: 'marker' ,
          color: 'blue'
        },
        text: plot_data['columns'],
      };
    }
```

+~70 MORE LINES

EXAMPLE 4 – OUTPUT, THE PRETTY BIT…



Random Page

This is a random page containing some results for users

EXAMPLE 5 – A SANDBOX

- blank python dict and html