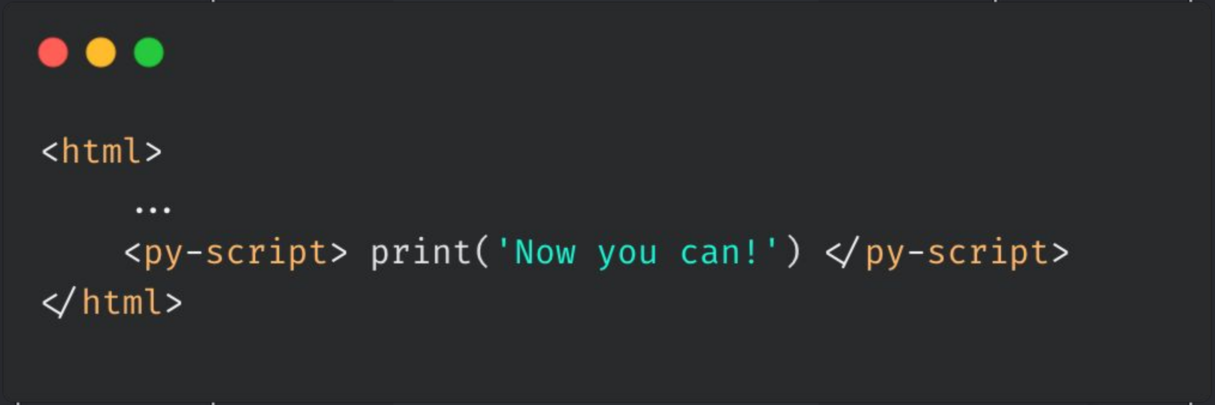
<py>



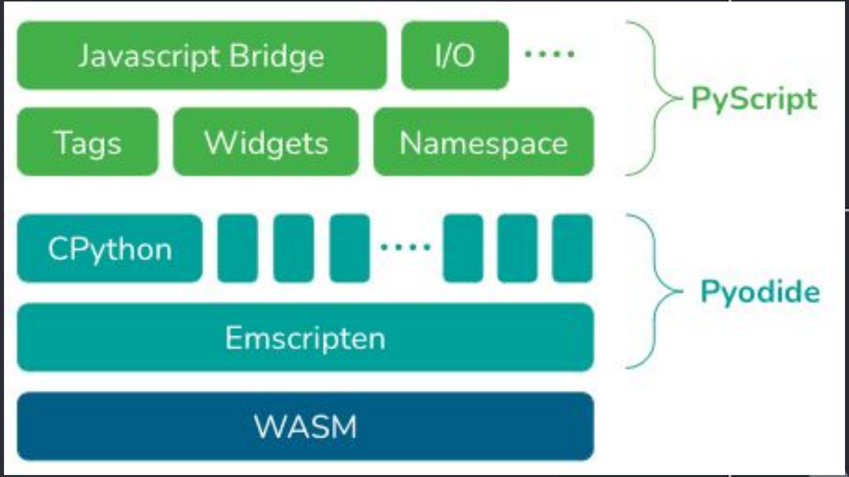
```
<html>
    ...
    <py-script> print('Now you can!') </py-script>
</html>
```

PyScript

A framework that allows you to run python script within HTML!

What is PyScript?

- A framework that allows users to create rich Python applications in the browser using HTML's interface.
- Aims to give users a first-class programming language that has consistent styling rules, is more expressive, and is easier to learn.
- There are some core components:
    - Python in the browser
    - Python ecosystem
    - Python with JavaScript
    - Environment management
    - Visual application development
    - Flexible framework



Sources: pyscript.net; https://anaconda.cloud/pyscript-python-in-the-browser

In conclusion, …

All that to say… PyScript is just HTML, only a bit (okay, maybe a lot) more powerful, thanks to the rich and accessible ecosystem of Python libraries.

In short, our mission is to bring programming for the 99%.

Sources: pyscript.net

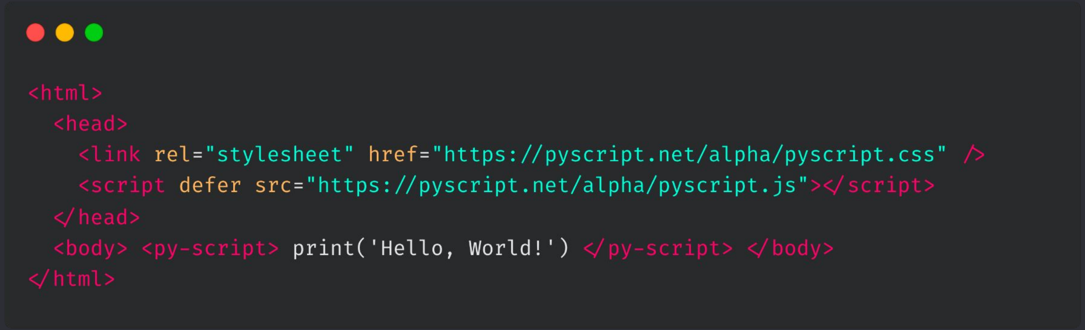
How to use PyScript?

PyScript lets you write Python in html using the following three main components:

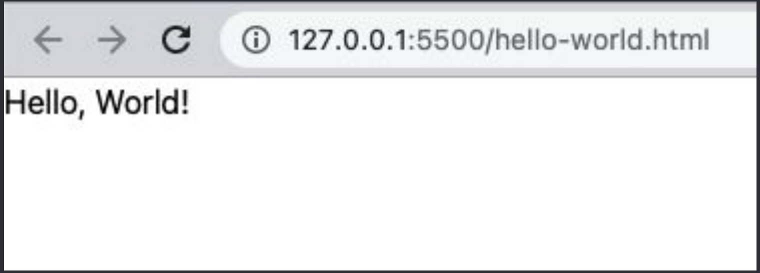
- **py-env** defines the Python packages needed to run your Python code.
- **py-script** is where you write your Python code that gets executed within the web page.
- **py-repl** creates a REPL (read-eval-print loop) component that evaluates the code users enter and displays the results.

Sources: https://anaconda.cloud/pyscript-python-in-the-browser

Sample codes: hello-world.html



```
<html>
  <head>
    <link rel="stylesheet" href="https://pyscript.net/alpha/pyscript.css" />
    <script defer src="https://pyscript.net/alpha/pyscript.js"></script>
  </head>
  <body> <py-script> print('Hello, World!') </py-script> </body>
</html>
```

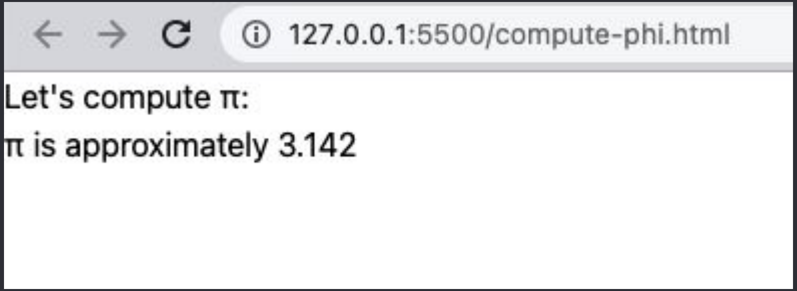


```
Hello, World!
```

Sources: https://github.com/pyscript/pyscript/blob/main/GETTING-STARTED.md

Sample codes: compute-phi.html

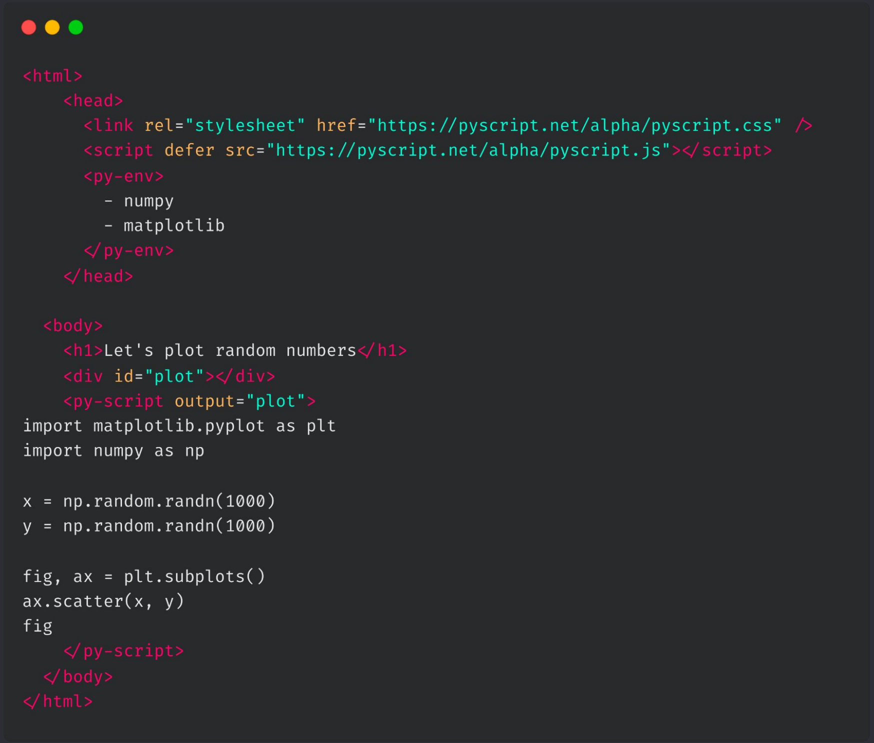
```
<html>
  <head>
    <link rel="stylesheet" href="https://pyscript.net/alpha/pyscript.css" />
    <script defer src="https://pyscript.net/alpha/pyscript.js"></script>
  </head>
  <body>
      <py-script>
print("Let's compute π:")
def wallis(n):
    pi = 2
    for i in range(1,n):
        pi *= 4 * i ** 2 / (4 * i ** 2 - 1)
    return pi

pi = wallis(100000)
s = f"π is approximately {pi:.3f}"
print(s)
      </py-script>
</html>
```



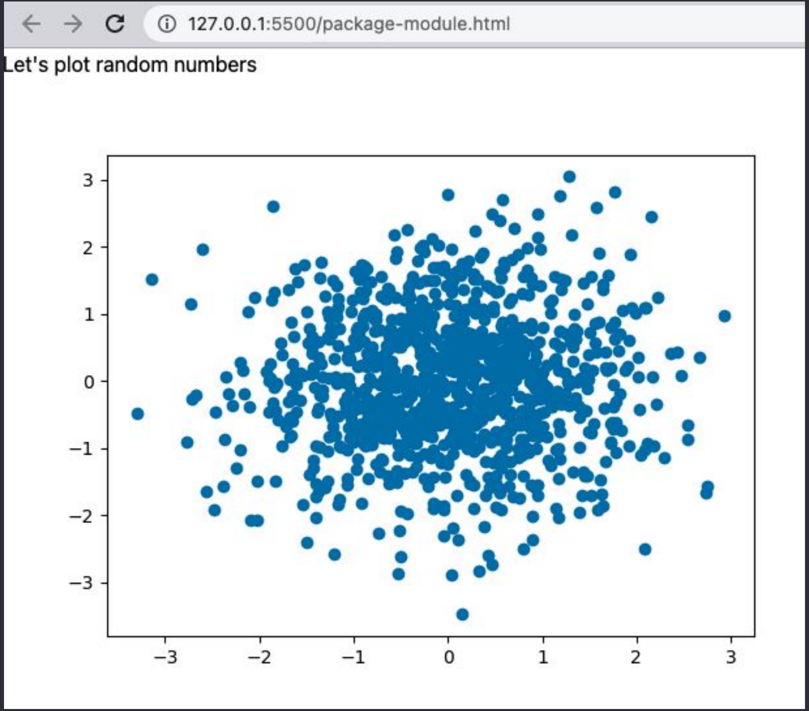
```
Let's compute π:
π is approximately 3.142
```

Sources: https://github.com/pyscript/pyscript/blob/main/GETTING-STARTED.md

Sample codes: package-module.html



```
<html>
    <head>
      <link rel="stylesheet" href="https://pyscript.net/alpha/pyscript.css" />
      <script defer src="https://pyscript.net/alpha/pyscript.js"></script>
      <py-env>
        - numpy
        - matplotlib
      </py-env>
    </head>

  <body>
    <h1>Let's plot random numbers</h1>
    <div id="plot"></div>
    <py-script output="plot">
import matplotlib.pyplot as plt
import numpy as np

x = np.random.randn(1000)
y = np.random.randn(1000)

fig, ax = plt.subplots()
ax.scatter(x, y)
fig
    </py-script>
  </body>
</html>
```



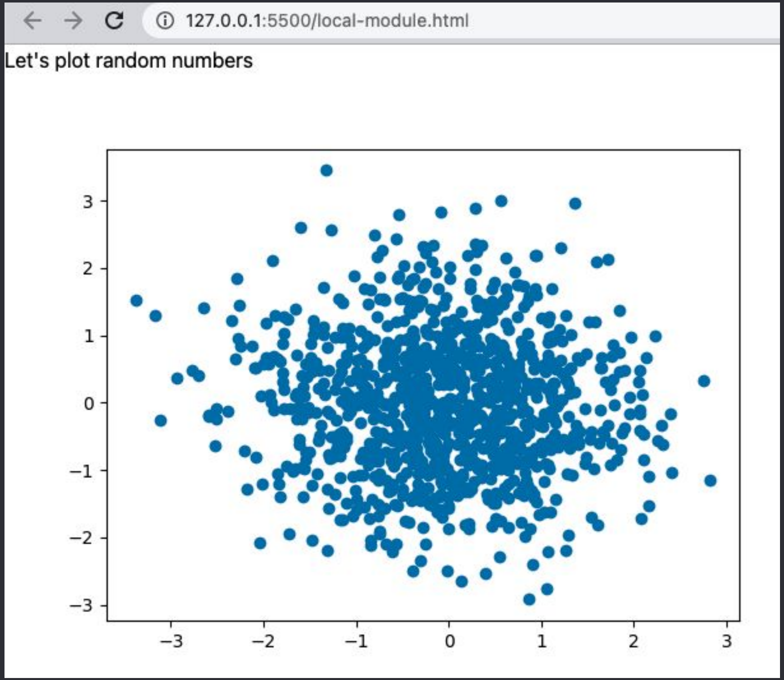
127.0.0.1:5500/package-module.html

Let's plot random numbers

Sources: https://github.com/pyscript/pyscript/blob/main/GETTING-STARTED.md

Sample codes: local-module.html

```
# data.py
import numpy as np

def make_x_and_y(n):
    x = np.random.randn(n)
    y = np.random.randn(n)
    return x, y

# local-module.html
<html>
    <head>
      <link rel="stylesheet" href="https://pyscript.net/alpha/pyscript.css" />
      <script defer src="https://pyscript.net/alpha/pyscript.js"></script>
      <py-env>
        - numpy
        - matplotlib
        - paths:
          - /data.py
      </py-env>
    </head>

  <body>
    <h1>Let's plot random numbers</h1>
    <div id="plot"></div>
    <py-script output="plot">
import matplotlib.pyplot as plt
from data import make_x_and_y

x, y = make_x_and_y(n=1000)

fig, ax = plt.subplots()
ax.scatter(x, y)
fig
    </py-script>
  </body>
</html>
```


127.0.0.1:5500/local-module.html

Let's plot random numbers

Sources: https://github.com/pyscript/pyscript/blob/main/GETTING-STARTED.md

What's next?

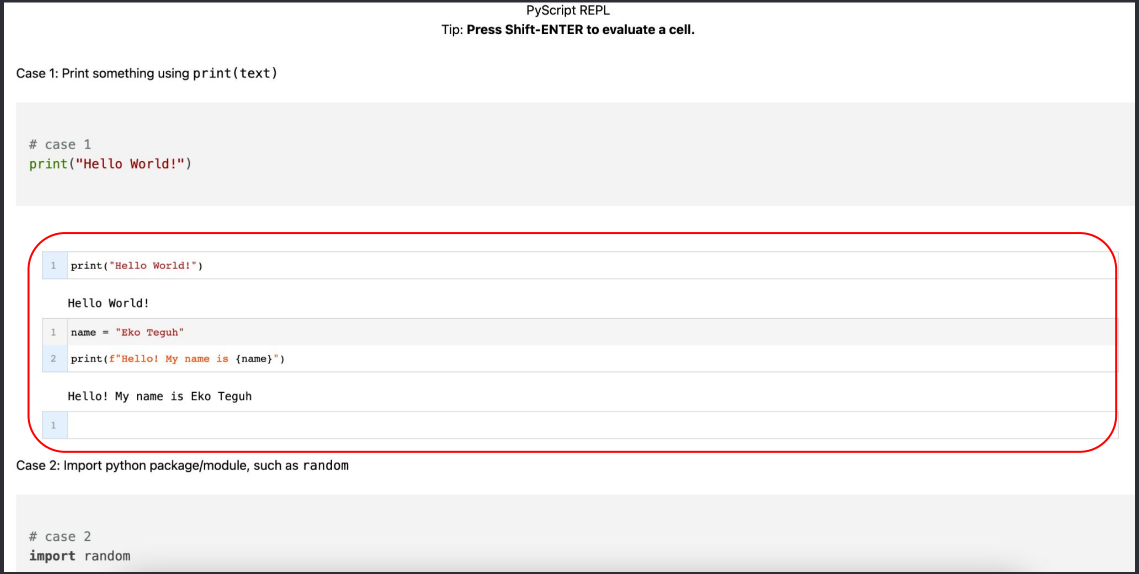
Trying another case and keep exercising!

This slide is made by Eko Teguh Widodo for well-documented format. Feel free to share...^_^

Additional codes: REPL.html

REPL (read-eval-print loop)

Tag py-repl creates a REPL (read-eval-print loop) component that evaluates the code users enter and displays the results.


PyScript REPL
Tip: **Press Shift-ENTER to evaluate a cell.**

Case 1: Print something using `print(text)`

```
# case 1
print("Hello World!")
```

```
print("Hello World!")
```

```
Hello World!
```

```
name = "Eko Teguh"
print(f"Hello! My name is {name}")
```

```
Hello! My name is Eko Teguh
```

Case 2: Import python package/module, such as `random`

```
# case 2
import random
```

```html
<div class="repl">
    <py-repl id="repl-4" auto-generate="true"></py-repl>
</div>
```

Sample codes: https://github.com/ekotwidodo/learning-pyscript/blob/main/REPL.html