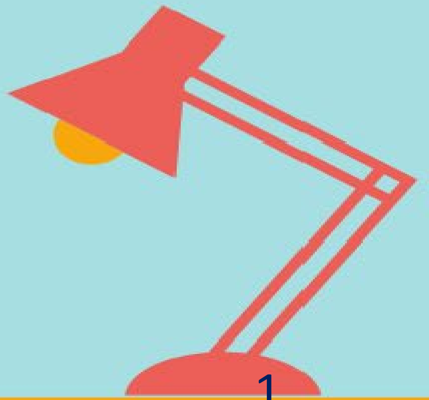
2023-Spring

Advanced Computer Programming (10)

CSIE, Asia Univ.



課程大綱

- W1-Introduction
- W2-Python libraries
- W3-BeautifulSoup(1)
- W4-BeautifulSoup(2)
- W5-
- W6-Scrapy(1)
- W7-Scrapy(2)
- W8-Storing Data
- W9-Midterm project
- W10-Web & HTTP
- W11-Flask
- W12-Flask Routes
- W13-Jinja template
- W14-Flask-form
- W15-Flask-mail
- W16-REST API
- W17-Project development(2)
- W18-Final presentation

Front-end Design

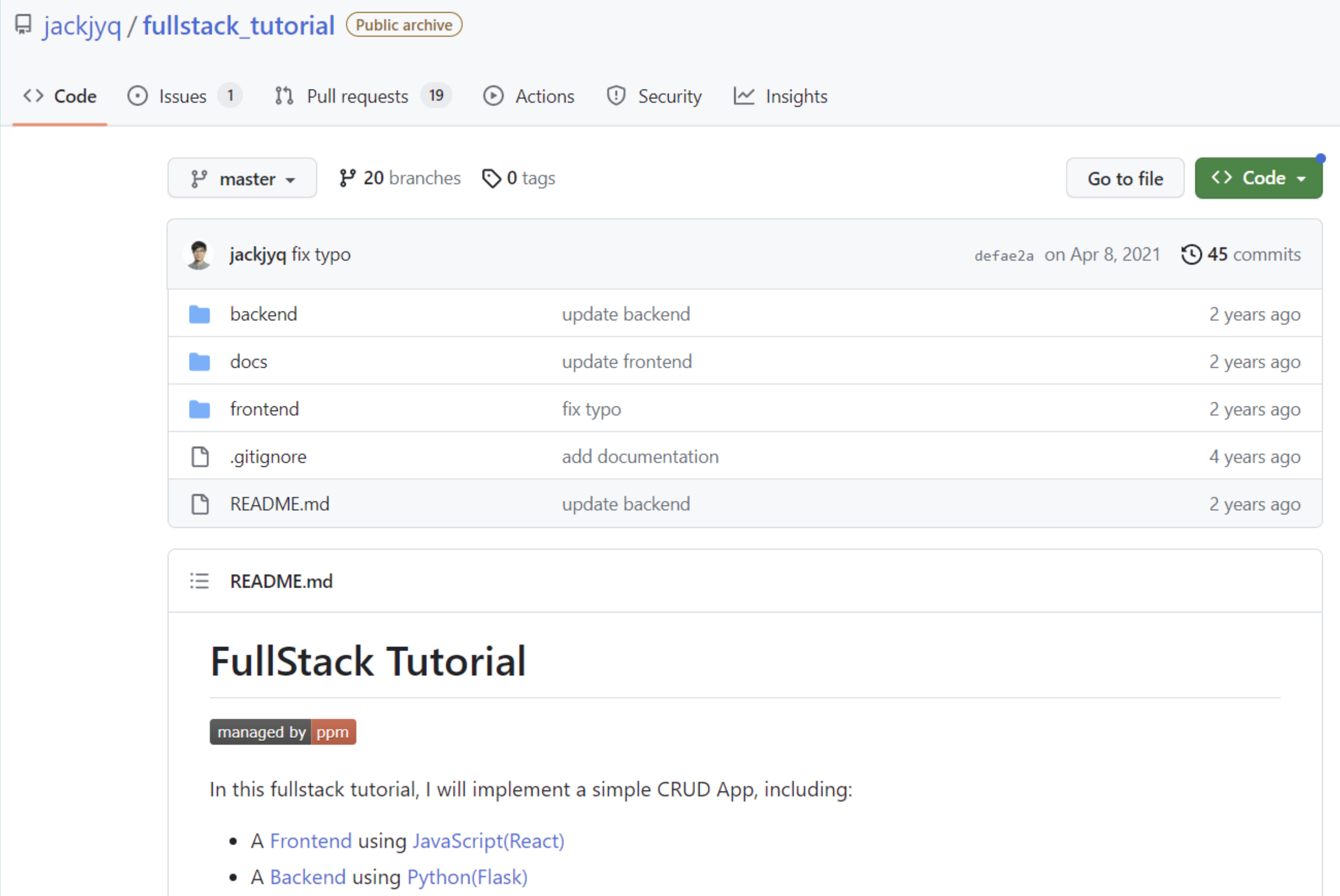
- Front-end covers design and implementation of user interface, interaction, animation effects, etc., including:
  - Front-end Development: Use HTML, CSS, JavaScript and other technologies to realize front-end design, including converting design into web pages, web animation, interactive effects, etc.
  - Cross-platform Design: Designers need to consider different browsers, operating systems, and devices so that web pages can run normally on various platforms.
  - User Experience Design (UX Design): Through user research and testing, designers design the structure, process, and content of the user interface to enhance user experience.
  - Web Design: Convert the visual design draft provided by the designer into an actual web page, design web page layout, color matching, fonts, etc., and consider the adaptability of different devices (computers, mobile phones, tablets).
  - Styling: Design the style of web page elements, such as color, font, size, border, background, etc., and implement these styles using CSS technology.
  - Interactive Design: Designers design how users interact with web pages, such as buttons, forms, slides, animation effects, etc.

Back-end Design

- Backend design refers to the process of designing and developing the server-side of a web application or software ：

1. Server-side programming languages: The choice of programming languages used to develop the backend logic, such as Java, Python, PHP, Ruby, or Node.js.
2. Databases: The choice of database technology and design of database schema to manage the application's data.
3. APIs: The design and implementation of APIs that allow the frontend of the application to communicate with the backend.
4. Authentication and Authorization: Designing the backend to handle user authentication and authorization, such as password management, permissions, and access control.
5. Server architecture: The design of the server architecture, including load balancing, caching, and other techniques to ensure scalability and high performance.

FullStack = Front-end + Back-end



jackjyq fix typo defae2a on Apr 8, 2021 45 commits

backend	update backend	2 years ago
docs	update frontend	2 years ago
frontend	fix typo	2 years ago
.gitignore	add documentation	4 years ago
README.md	update backend	2 years ago

README.md

FullStack Tutorial

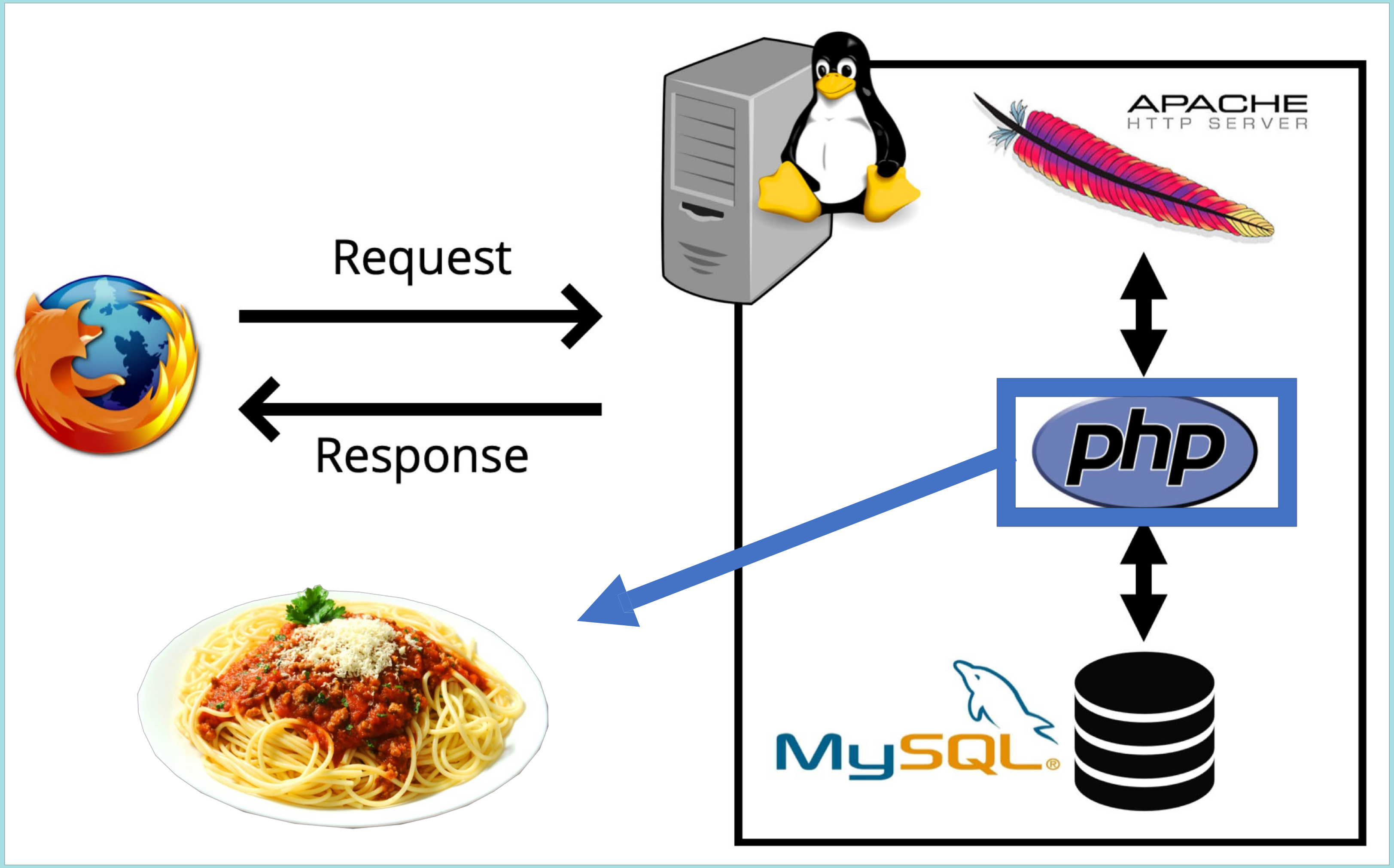
managed by ppm

In this fullstack tutorial, I will implement a simple CRUD App, including:

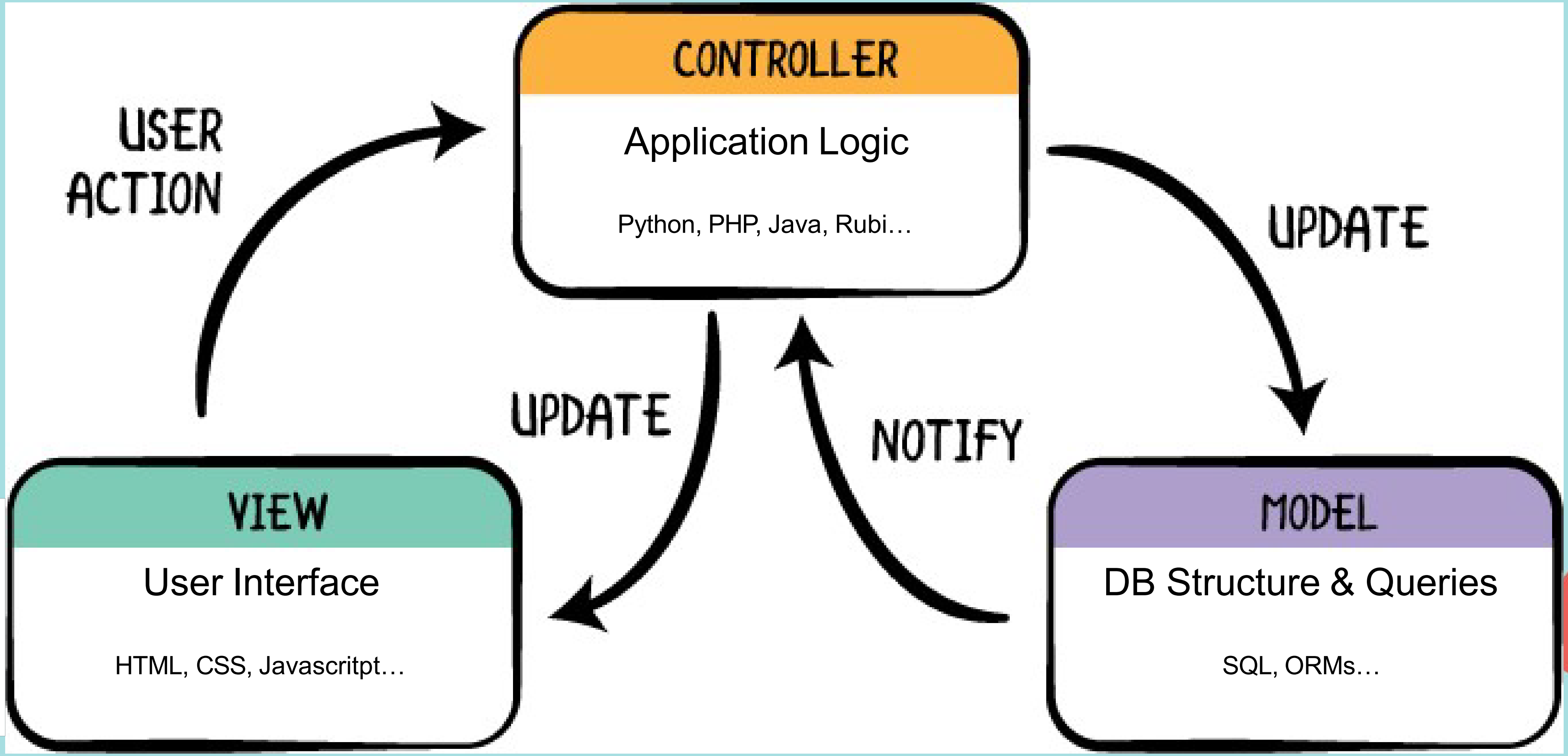
- A Frontend using JavaScript(React)
- A Backend using Python(Flask)

https://github.com/jackjyq/fullstack_tutorial

AMP: Apache, MySQL, PHP



MVC: Model View Controller



Application Logic

Python, PHP, Java, Rubi…

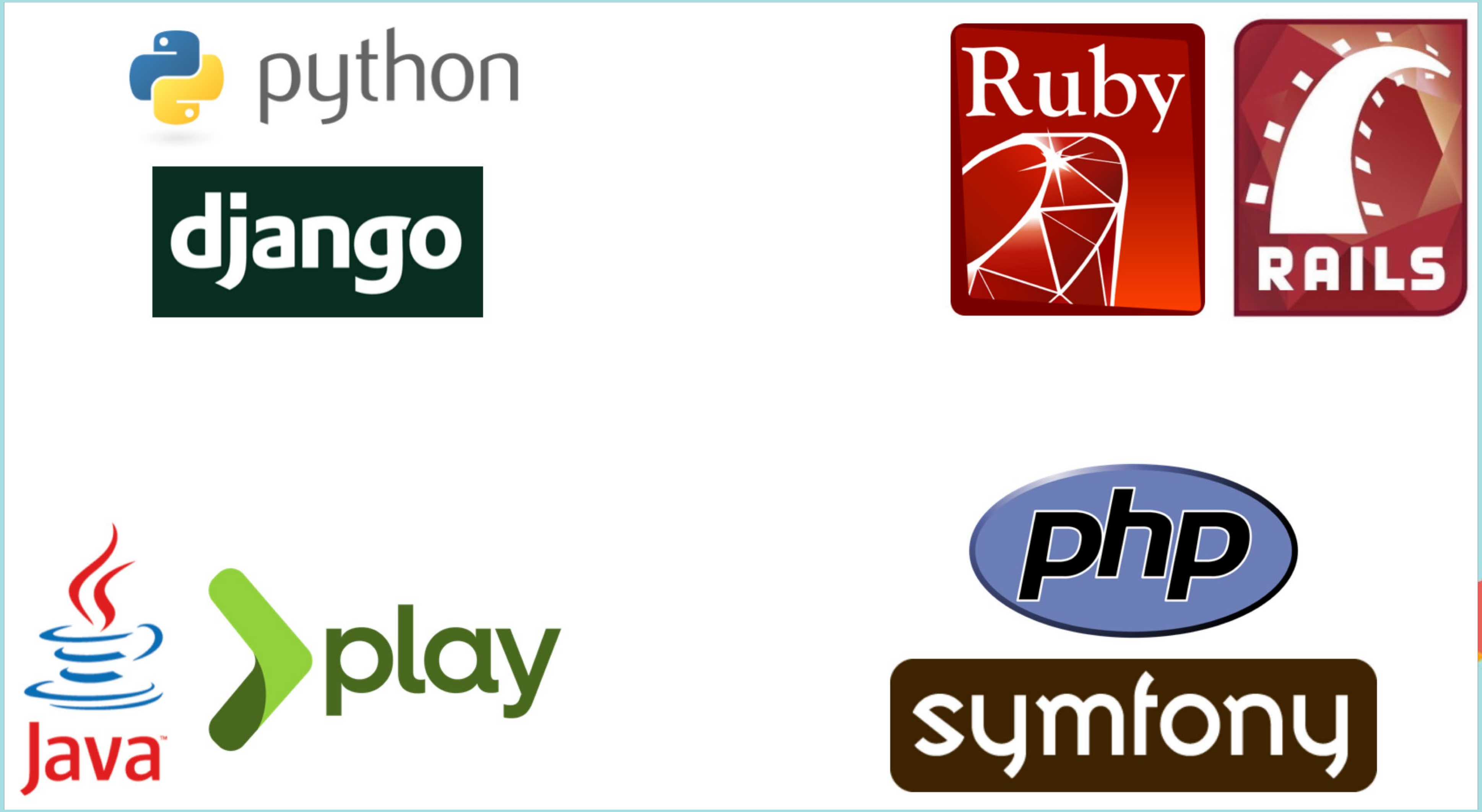
User Interface

HTML, CSS, Javascritpt…

DB Structure & Queries

SQL, ORMs…

(Big) Frameworks



(Micro µ) **Frameworks**

python
Flask
web development,
one drop at a time

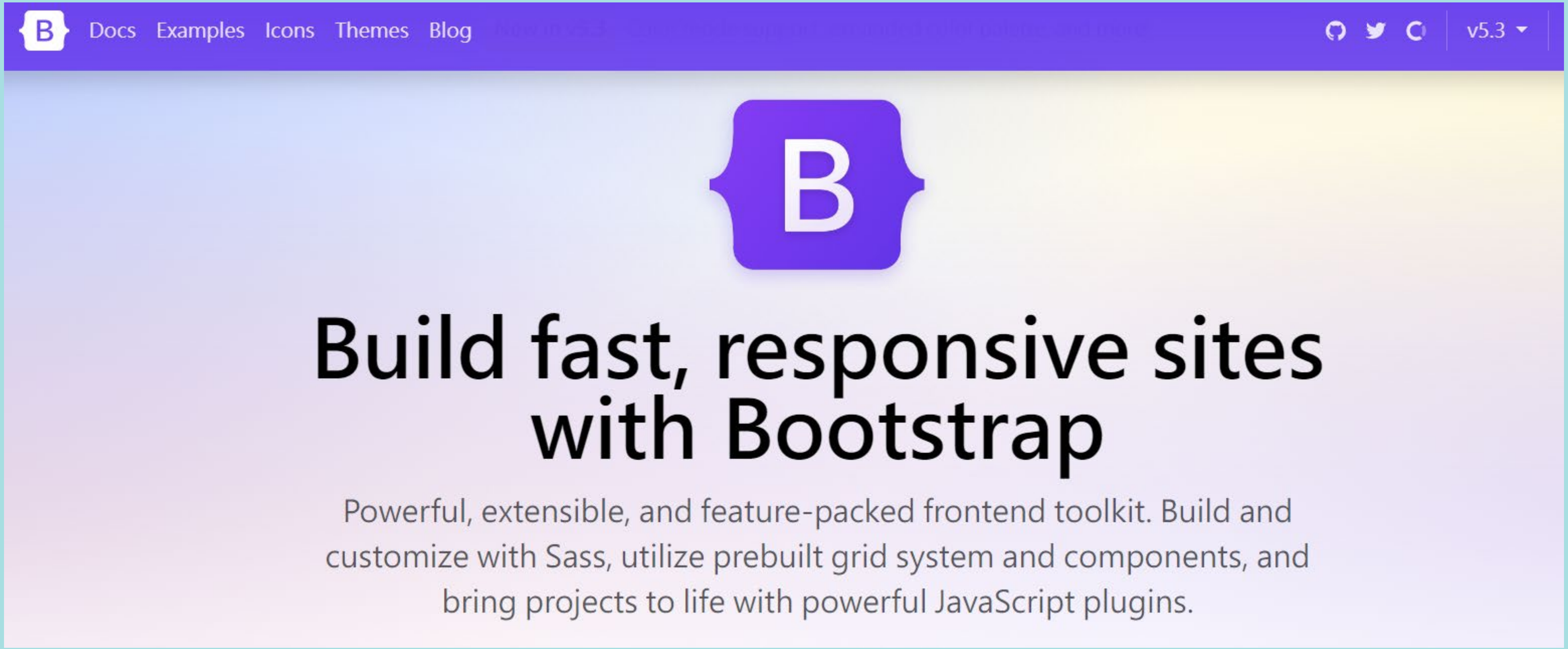
Ruby
Sinatra

node JS ™
Express

php
Slim
a micro framework for PHP

web development,
one drop at a time

Bootstrap



Build fast, responsive sites with Bootstrap

Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.

https://getbootstrap.com/

Thanks! Q&A