

# COM5961 DATA DRIVEN PRODUCTS & SERVICES DESIGN:

## LESSON 1 - BASIC PYTHON PROGRAMMING I

Bernard Suen  
 Center for Entrepreneurship  
 Chinese University of Hong Kong

# **Today's agenda.**

- 1. What is Full Stack Development (全栈开发)**
- 2. Learning Python with Anaconda Jupiter Notebook**
- 3. Why Anaconda? Why Python?**
- 4. Data Types, Variables and Operations**
- 5. Control operation flow with Conditionals and Loops**
- 6. Built-in data structures: List and Dictionary**

**Browser**  
Screen Display

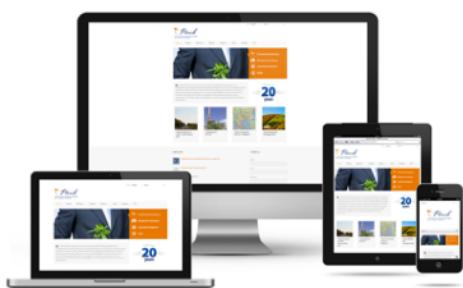


**Web Server (Github)**  
HTML/CSS/JS



## **Static Website**

No Back-end  
Processing



```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="utf-8">
5  <title> A Tiny HTML Document </title>
6  <link href = "styles.css" rel="stylesheet">
7  <script src="scripts.js"></script>
8  </head>
9
10 <body>
11 <p>Let's rock the browser, HTML5 style.</p>
12 </body>
13 </html>
```

## Browser

Screen Display



## Web Server

HTML/CSS/JS



## Database Server

Data



## Dynamic Website

With Back-end  
Processing



```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title> A Tiny HTML Document </title>
6     <link href = "styles.css" rel="stylesheet">
7     <script src="scripts.js"></script>
8   </head>
9
10  <body>
11    <p>Let's rock the browser, HTML5 style.</p>
12  </body>
13  </html>
```

ID	PHONE	POPULARNAME	PREFERREDNAME	LATITUDE	LONGITUDE
1194820	00994614	popular_name_00994614	preferred_name_00994614	23.789875	88.897985
1194821	00994616	popular_name_00994616	preferred_name_00994616	23.789875	88.897985
1194822	00994618	popular_name_00994618	preferred_name_00994618	23.789875	88.897985
1194823	00994617	popular_name_00994617	preferred_name_00994617	23.789875	88.897985
1194824	00994619	popular_name_00994619	preferred_name_00994619	23.789875	88.897985
1194825	00994619	popular_name_00994619	preferred_name_00994619	23.789875	88.897985
1194826	00994620	popular_name_00994620	preferred_name_00994620	23.789875	88.897985
1194827	00994621	popular_name_00994621	preferred_name_00994621	23.789875	88.897985
1194828	00994622	popular_name_00994622	preferred_name_00994622	23.789875	88.897985
1194829	00994623	popular_name_00994623	preferred_name_00994623	23.789875	88.897985
1194830	00994624	popular_name_00994624	preferred_name_00994624	23.789875	88.897985
1194831	00994625	popular_name_00994625	preferred_name_00994625	23.789875	88.897985

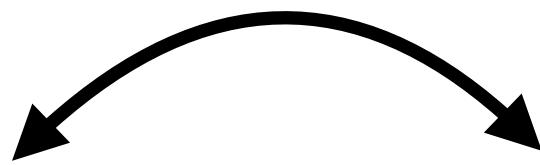
## FRONT-END PROCESSING

HTML/CSS/JS



React/Angular/Vue

## DATA FLOW



## WEB APP FRAMEWORKS For Front-end & Back-end

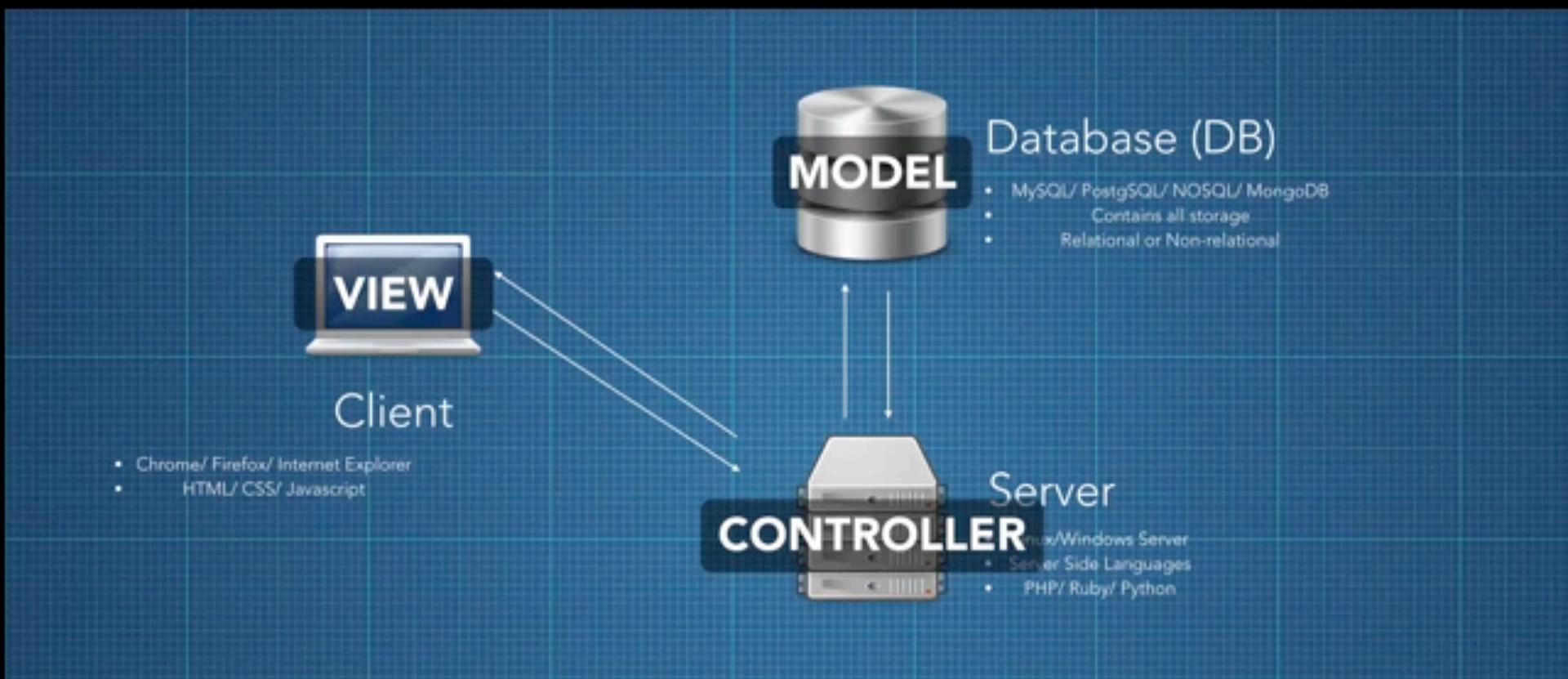
## BACK-END PROCESSING

PYTHON/PHP,  
NODE.JS/C# etc



FLASK/Django  
Laravel  
EXPRESS.JS/.NET

## HOW DOES A WEBSITE WORK? THE FLOW



## API (e.g. RESTful)



## BACK-END TECHNOLOGY

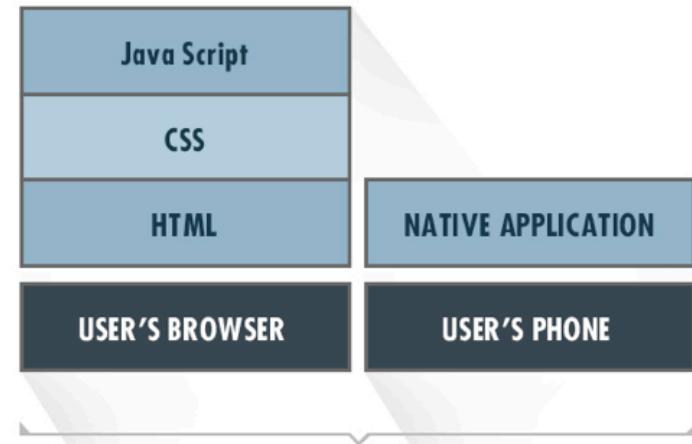
后台

## WHAT IS YOUR “CLOUD” AND “STACK” STRATEGY?

云架构及前端和后端的全栈策略

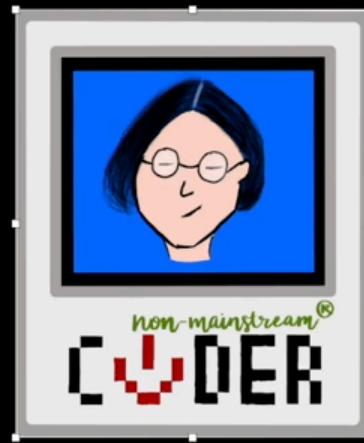
{ JSON }

THE INTERNET



## FRONT-END TECHNOLOGY

前台



Non-mainstream Coder

@非主流程序员

# **WHY PYTHON ?**

- Simple syntax designed as a teaching language for learning Computer Science.
- Extensive support of programming libraries, especially for doing AI and data science related works.
- Commercial adoptions have increased rapidly, providing plenty of job opportunities.
- Used by YouTube, Dropbox, Facebook, Netflix, Google, Instagram, Spotify, Reddit, Yahoo Maps, Pinterest, Washington Post, NASA, etc.
- Like PHP, Ruby, Node.JS, Java and C#, Python is a very powerful back-end language for web programming.

# **DEVELOPMENT ENVIRONMENT**

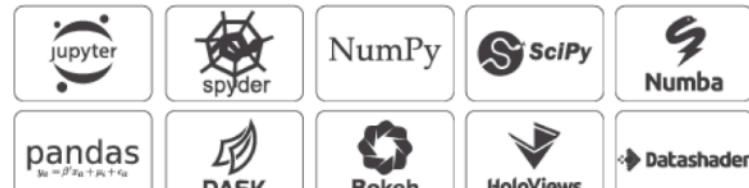
[Products](#)[Why Anaconda?](#)[Solutions](#)[Resources](#)[Company](#)[Contact Us](#)[Download](#)

# Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

[Download](#)

The open-source [Anaconda Distribution](#) is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:



This website uses cookies to ensure you get the best experience on our website. [Privacy Policy](#)

[ACCEPT](#)

Waiting for staticcontent.com...

<https://www.anaconda.com/distribution/>

# PYTHON DEVELOPMENT WITH ANACONDA AND JUPYTER NOTEBOOK

- Why Anaconda?
- Most of the popular Python libraries such as BeautifulSoup, NumPy, Pandas, Mathplotlib are preinstalled
- Other additional software tools such as R Studio, JupyterLab, Jupyter Notebook are also preinstalled
- Cross-platform support on Mac, Windows, and Linux
- Jupyter Notebook is a great tool for learning Python
- Jupyter Notebook supports HTML/CSS markup
- Github support for Jupyter notebook document

# **DATA TYPES**

1. Integer
2. Float
3. String
4. Boolean

# **VARIABLES & OPERATIONS**

1. Variable assignment (=)
2. Addition (+)
3. Subtraction (-)
4. Multiplication (\*)
5. Division (/)
6. Modulus (%)
7. Exponent (\*\*)

# **CONDITIONS & CONTROL FLOW**

1. == (equal)
2. != (not equal)
3. >= (greater than and equal)
4. <= (less than and equal)
5. not
6. if ... elif ... else

# **LISTS AND DICTIONARIES**

# **Basic Data Structures in Python**

- List and Tuple
- Dictionary and Set

## **Tuple and List**

- Tuples are arrays enclosed with round brackets for storing multiple variables
- Variables with data types can be stored into a Tuple
- A Tuple operates like a string and therefore can be indexed from the beginning (positive) and the end (negative)
- Tuples are immutable
- Inorder to manipulate a tuple, a new one has to be created
- Tuples can be nested e.g. Tuple2 = (1,2,(3,4),5)
- Due to its rigidity, List is more commonly used than Tuple.
- Lists are like Tables but are mutable and enclosed with square brackets
- List, similar to a Tuple, operates like a string when it comes to access individual element within the data structure

# **Using Dictionary to Store Data**

## **Set and Dictionary**

- Similar to lists and tuples, sets support different Python types
- Sets use {} (braces) to embed values/elements
- Sets do not allow duplicates
- Lists can be converted into sets with the set function
- Dictionary store data in an array of key-value pairs in braces
- For instance, here is a dictionary instance: dict = {"key1":1,"key2":2,"key3":3}
- 1st column representing the key and 2nd column representing the value
- .keys() returns all the keys
- .values() returns all the values

**Thank you for your time!**