



# NAVIGATING DATA CYCLE

Bernard Suen  
Center for Entrepreneurship  
Chinese University of Hong Kong



Center for  
Entrepreneurship

VERB

JS

JavaScript = act on a HTML tag, CSS property or respond to an event triggered by user action

Decomposition

Pattern

Abstraction

Algorithm

Automation &  
Testing

**Computational thinking is about  
system and data.**

**JavaScript provides us with the capabilities to build system and transform data.**

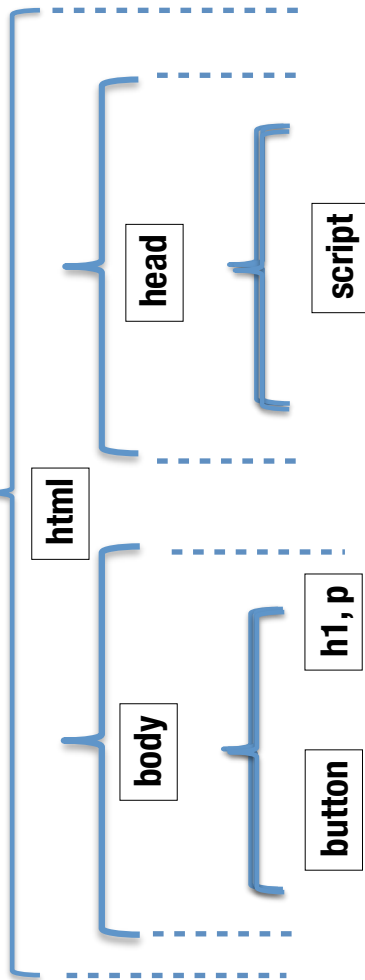
# **JavaScript Functions that Transform Input into Output**

## Basic Structure of a JavaScript Function

optional parameters  
↓

```
<head>  
<script>  
function function_name(parameter1, parameter 2...) {  
    Embed data type variables, input/output commands and  
    logical and mathematical operators in the function to  
    compute and return values.  
}  
</script>  
</head>
```

## Code View



```
<!DOCTYPE html>
<html>
<head>
<script>
function show_topic() {
    var x = document.getElementById("demo");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
</script>
</head>
<body>
<h1>Demo Page</h1>
<p id="demo">Click the button to change the color and font
style of this line.</p>
<button onclick="show_topic()">Click this button!</button>
</body>
</html>
```



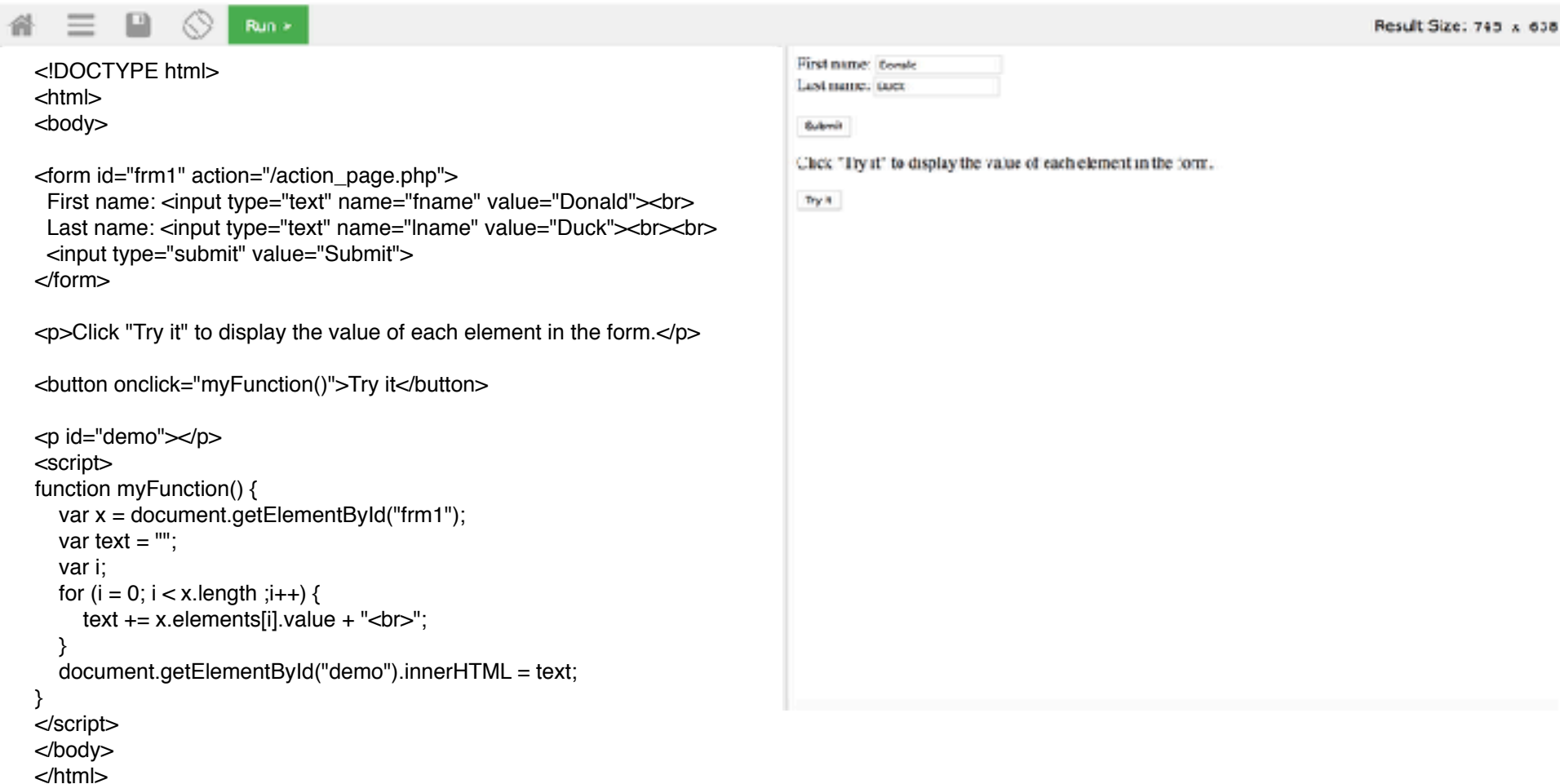
**Similar to CSS, JS can be placed in  
an External File**

# Functions in JavaScript Programming

- You can look at a function as a mini-system.
- A function is designed to transform input into output.
- You can execute a function within another function.
- A program can be viewed as a collections of functions decomposed into hierarchy of functions to get things done.
- Good programmer looks for patterns in job to be done and abstract common parameters, algorithms, and outcomes to be placed inside a function for code reuse (automation).

# **The HTML Form and Input Objects(Tags)**

[https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_form\\_elements](https://www.w3schools.com/js/tryit.asp?filename=tryjs_form_elements)



The screenshot shows the W3Schools JavaScript TryIt editor interface. The top bar includes navigation icons and a 'Run' button. The right side of the interface displays the 'Result Size: 745 x 638'.

The code editor on the left contains the following HTML and JavaScript code:

```
<!DOCTYPE html>
<html>
<body>

<form id="frm1" action="/action_page.php">
  First name: <input type="text" name="fname" value="Donald"><br>
  Last name: <input type="text" name="lname" value="Duck"><br><br>
  <input type="submit" value="Submit">
</form>





<p>Click "Try it" to display the value of each element in the form.</p>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>
<script>
function myFunction() {
  var x = document.getElementById("frm1");
  var text = "";
  var i;
  for (i = 0; i < x.length ;i++) {
    text += x.elements[i].value + "<br>";
  }
  document.getElementById("demo").innerHTML = text;
}
</script>
</body>
</html>
```

The rendered output on the right shows a form with two text input fields labeled 'First name:' and 'Last name:', both containing the values 'Donald' and 'Duck' respectively. Below the inputs is a 'Submit' button. A message below the form reads: 'Click "Try it" to display the value of each element in the form.' Below this message is a 'Try it' button. The bottom of the rendered area shows a paragraph with the ID 'demo', which is currently empty.

[https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_form\\_elements](https://www.w3schools.com/js/tryit.asp?filename=tryjs_form_elements)

    Run ▶

Result Size: 745 x 638

First name:

Last name:

Click "Try it" to display the value of each element in the form.

```
<!DOCTYPE html>
<html>
<body>

<form id="frm1" action="/action_page.php">
  First name: <input type="text" name="fname" value="Donald"><br>
  Last name: <input type="text" name="lname" value="Duck"><br><br>
  <input type="submit" value="Submit">
</form>

<p>Click "Try it" to display the value of each element in the form.</p>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>
<script>
function myFunction() {
  var x = document.getElementById("frm1");
  var text = "";
  var i;
  for (i = 0; i < x.length ;i++) {
    text += x.elements[i].value + "<br>";
  }
  document.getElementById("demo").innerHTML = text;
}
</script>
</body>
</html>
```

<https://getbootstrap.com/>

# Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

[Get started](#)

[Download](#)

Currently v4.1.2



# **Using Bootstrap Form and Button to Make Browser UI Look Better**

Source: <https://getbootstrap.com/docs/4.0/components/forms/>

Email address

We'll never share your email with anyone else.

Password

☐ Check me out



# Buttons

Use Bootstrap's custom button styles for actions in forms, dialogs, and more with support for multiple sizes, states, and more.



Try DigitalOcean today  
and get a free \$100  
credit.  
ads via Carbon

## Examples

<https://getbootstrap.com/docs/4.0/components/buttons/>

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control.



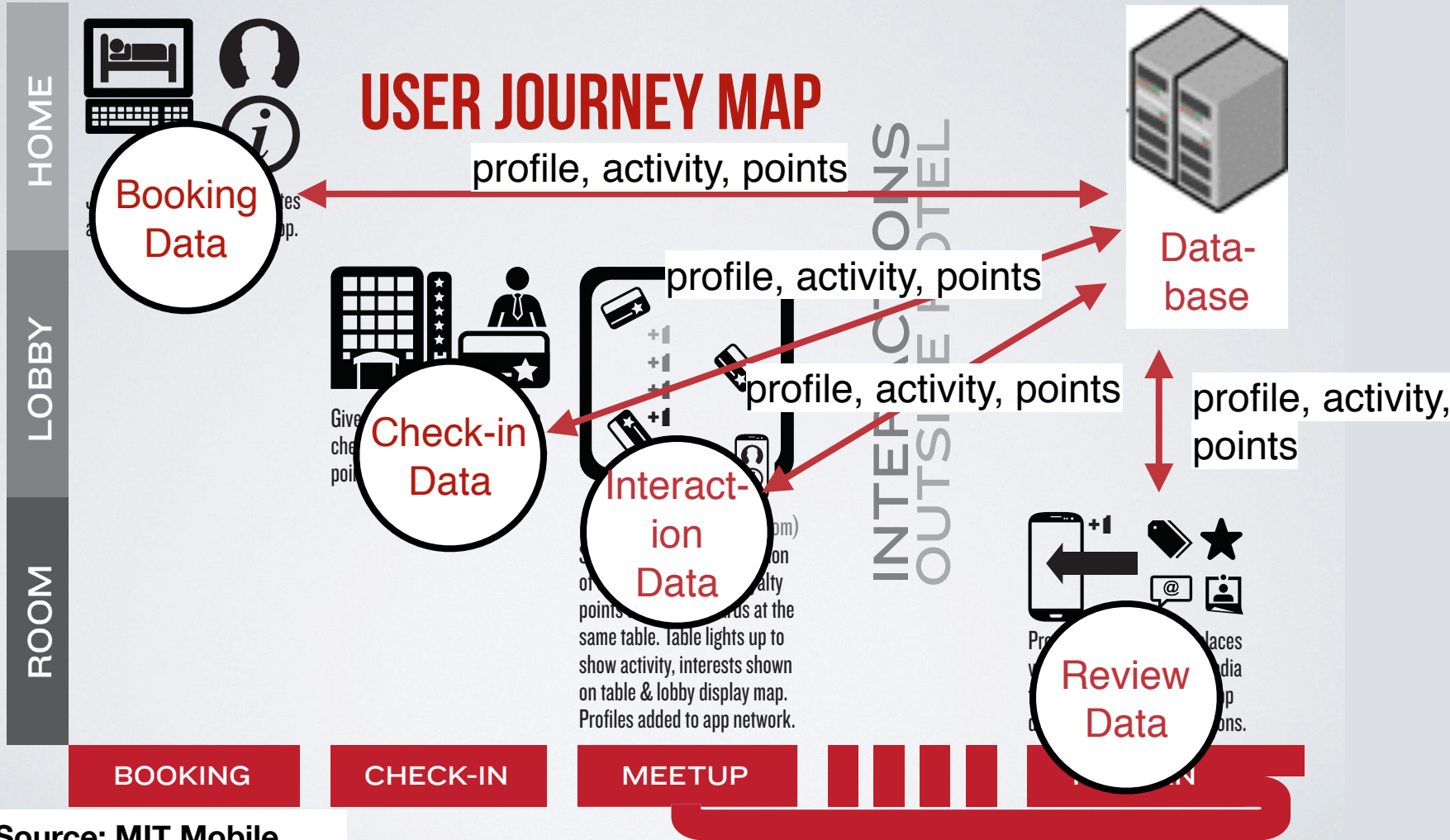
```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

# The Grammar of JavaScript

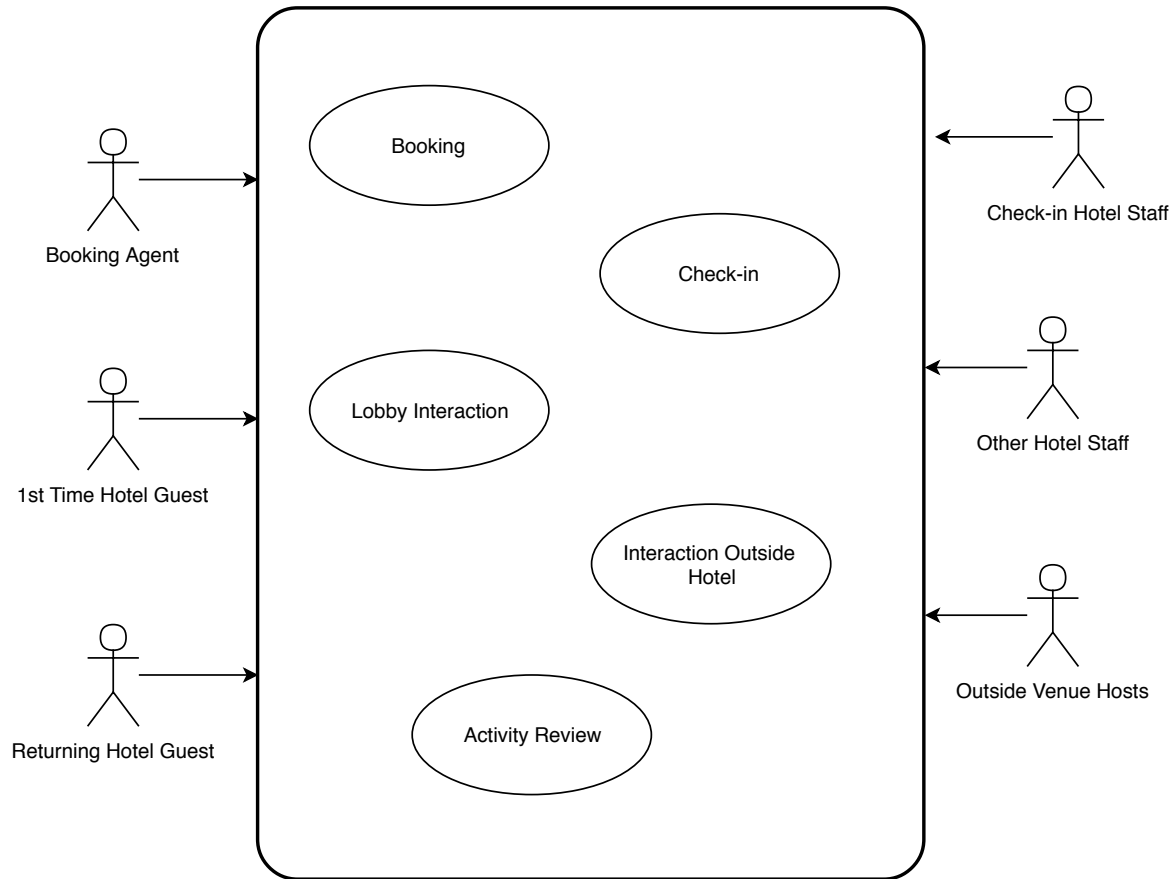
- JavaScript is a programming language that can be used to write functions placed inside html or an external file.
- JavaScript can be placed between the <script> and </script> tags inside the <head> section or link to an external file through the script src link.
- JavaScript codes can be understood as a collection of functions that respond to events triggered by internal browser activities and external user interactions.
- JavaScript can be used to manipulate HTML elements and CSS styles.

# **DATA MODELLING CYCLE**



Source: MIT Mobile Experience Lab

**To understand and construct our world  
through data models.**



## Use Cases of a Hotel Loyalty App

**Derive data entities from use cases.**

## Data Entities to be Tracked

Guests

- e.g.
  - 1st time guests
  - Returning guests

Venues

- e.g.
  - Starbuck
  - Bar inside hotel
  - Boston Metropolitan Museum

Social Interactions

- e.g.
  - Add friends
  - Take pictures
  - Share rides
  - Split bills

Activity Types

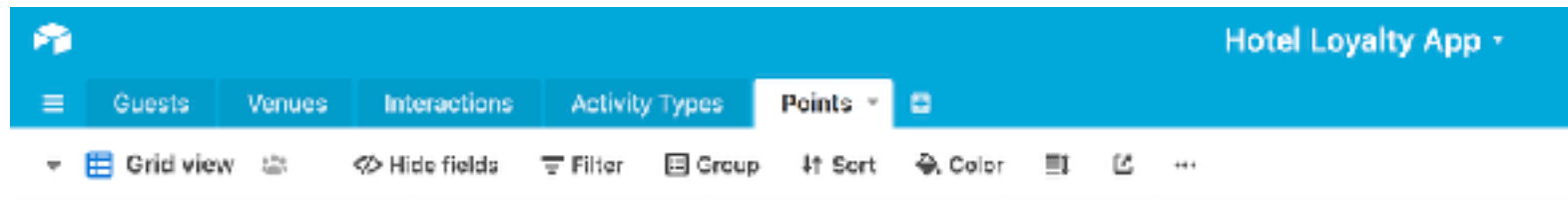
- e.g.
  - Museum tour
  - Bar hopping
  - Scenic picture taking
  - Business meeting

Points

- e.g.
  - Revisit
  - Dine in hotel
  - Shop in hotel
  - Initiate contact
  - Give reviews



# Tracking Data Entities



**Where does the data come from?**

- Open Data
- Scraping (e.g. ParseHub, Scrappy, BeautifulSoup)
- API
- Web Log
- IoT
- Data Science - R
- Python based tools e.g. Pandas, ScikitLearn, TensorFlow
- Knime/Airtable filter, formula and roll-up
- SQL Queries, Data Visualization
- Charts & Maps (e.g. Tableau)
- JS Mapping/Charting Tool (e.g. Leaflet, C3D3)

acquire — parse — filter — mine — represent — refine — interact

- Cleaning (e.g. Open Refine)
- Grouping
- Transformation

- Charts & Maps (e.g. Tableau)
- JS Mapping/Charting Tool (e.g. Leaflet, C3D3)
- Bootstrap Layout

source: Ben Fry's Visualizing Data



**INTERNAL  
DATABASE**



**OPEN DATA FILES**



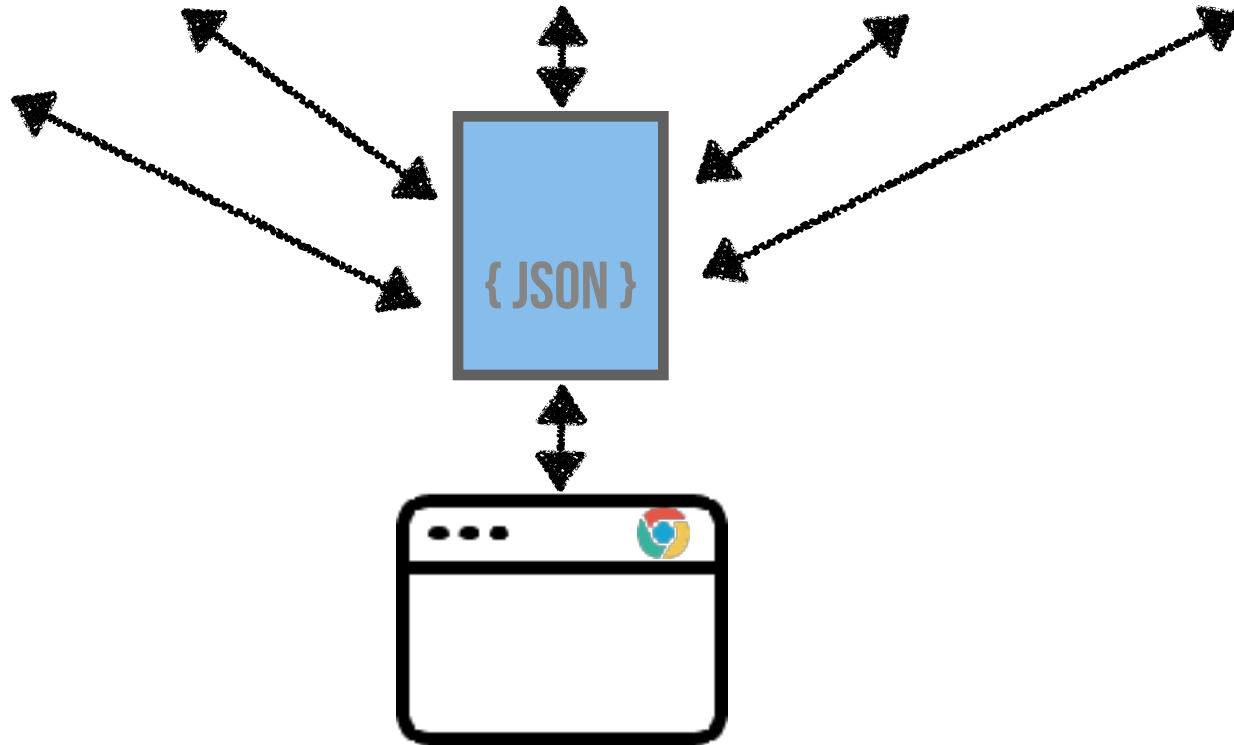
**SCRAPING**



**API**



**IOT**



source:  
freepik and  
madeyoliver

# **RISE OF THE SMART CITIES & OPEN DATA**

# 香港智慧城市

Hong Kong Smart City 藍  
Blueprint 圖



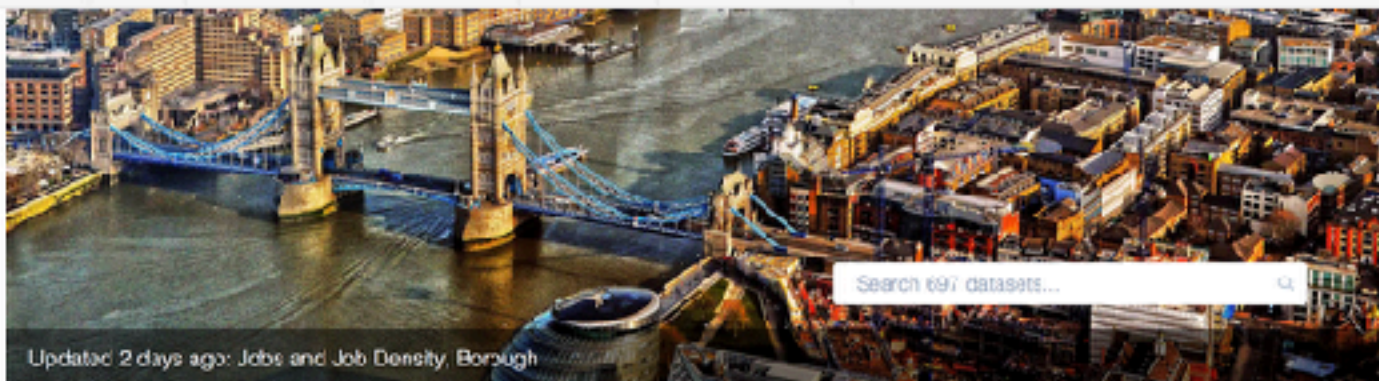
# Open Data for All New Yorkers

Where can you find public Wi-Fi in your neighborhood? What kind of tree is in front of your office? Learn about where you live, work, eat, shop and play using NYC Open Data.

Search Open Data for things like 311, Buildings, Crime



## LONDON DATASTORE

[Register an Account](#)[Login](#)[Blog](#)[Data](#)[Apps & Analysis](#)[Developers](#)[Boroughs](#)[City Data Strategy](#)[More](#)JOBS AND  
ECONOMY

TRANSPORT

ENVIRONMENT

COMMUNITY  
SAFETY

HOUSING

COMMUNITIES

HEALTH

LONDON AS A  
WORLD CITYGLA  
PERFORMANCE[Click on a circle to see more...](#)

## Welcome to the Datastore

The London Datastore is a free and open data sharing portal where anyone can access data relating to the capital. Whether you're a citizen, business owner, researcher or developer, the site provides over 700 datasets to help you understand the city and develop solutions to London's problems. Please do have a look around and [tell us](#).





**[ Hong Kong ]**  
Smart City Blueprint

[Home](#) [Vision & Mission](#) [Development Plans](#) [Open e-Blueprint](#) [Download Blueprint](#)

整期 A A A

# Open Data

[Home](#) > [Open Data](#)

## Smart Mobility

## Smart Living

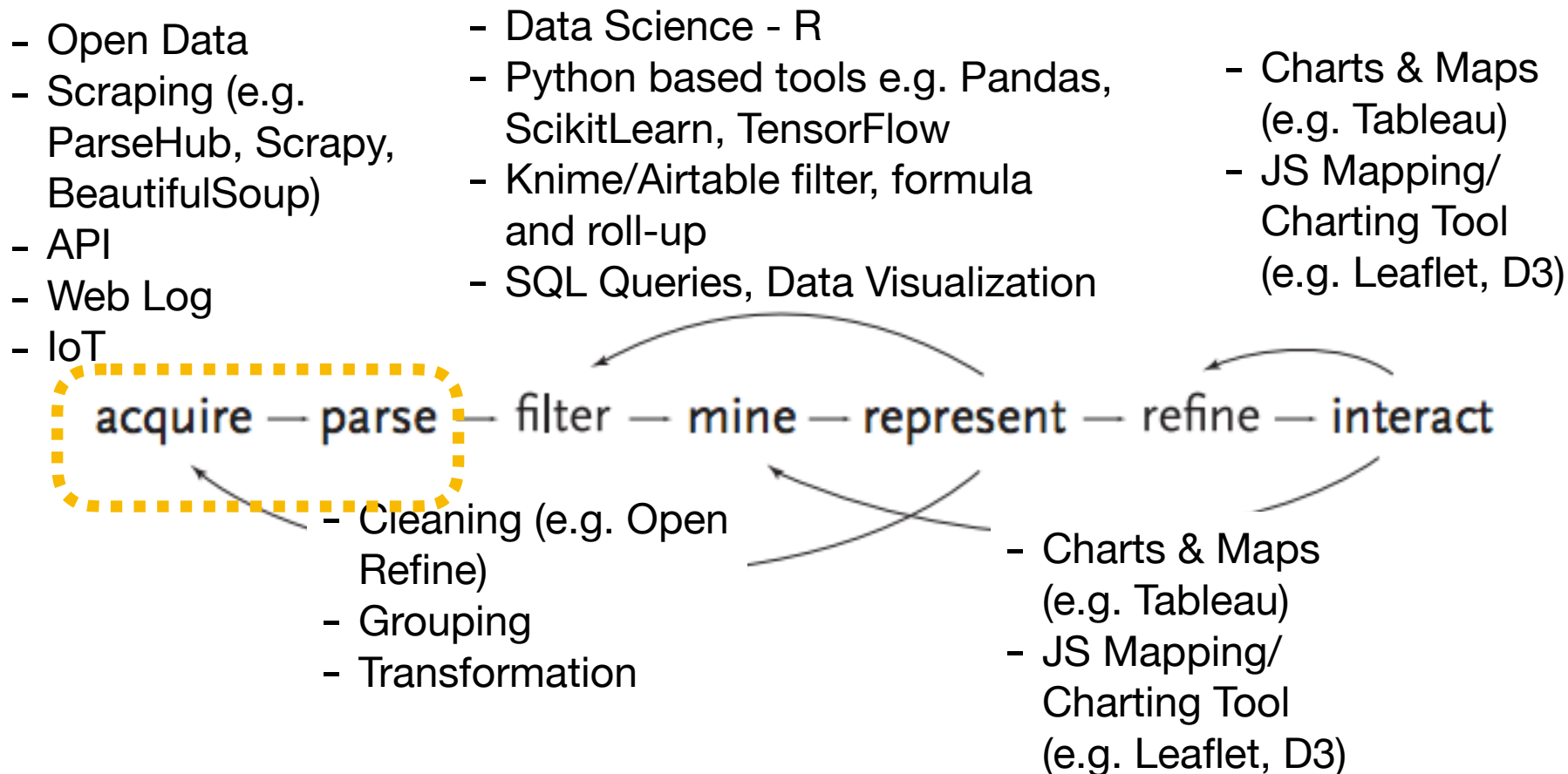
Smart  
Environment

## Smart People

Smart  
Government

## Smart Economy

## Open Data



source: Ben Fry's Visualizing Data

## THE REVAMPED PORTAL



POPULAR DATA  
GEOSPATIAL DATA  
FEATURED DATASETS

New to The Site? [START HERE](#)

Search Data (eg. population)



### BROWSE DATASETS



City  
Management



Climate and  
Weather



Commerce  
and Industry



Development



Education



Employment  
and Labour



<https://www.luxonsoftware.com/converter/xmltocsv>

<https://json-csv.com/>



**INTERNAL  
DATABASE**



**OPEN DATA FILES**



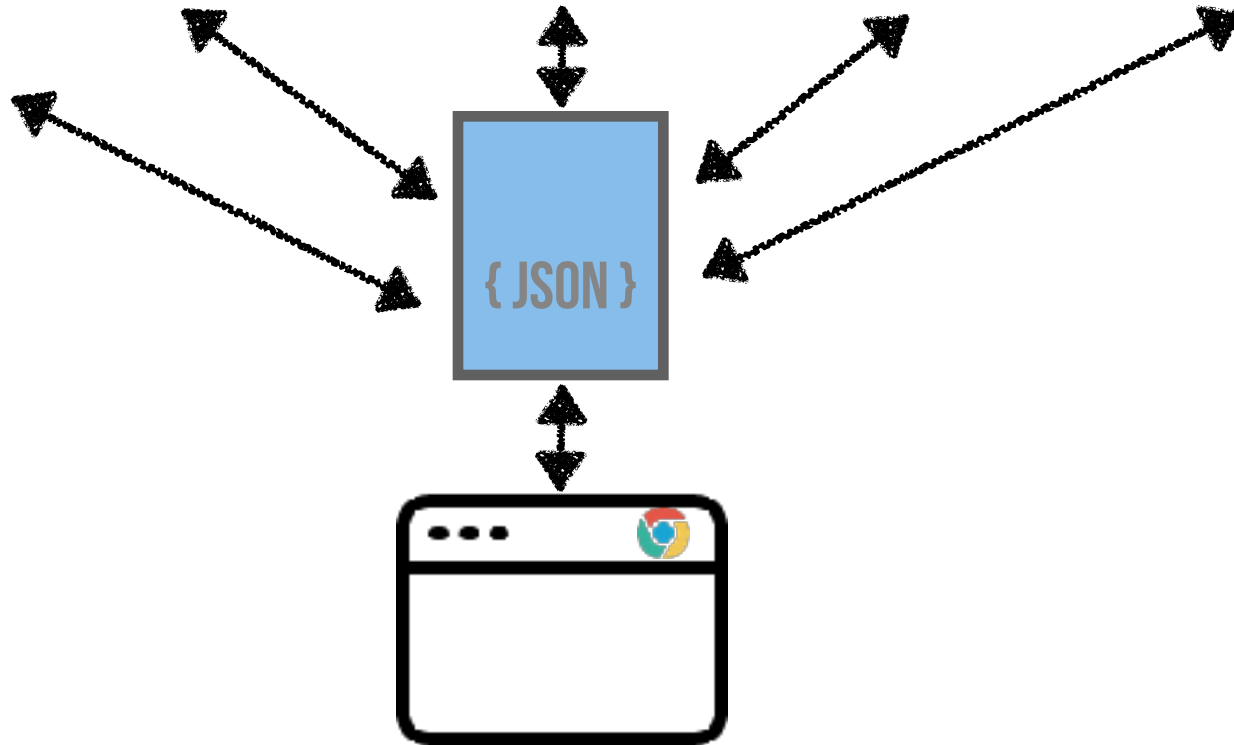
**SCRAPING**



**API**



**IOT**



source:  
freepik and  
madeyoliver

## A web scraping tool that is easy to use

ParseHub is a free web scraping tool. With our advanced web scraper, extracting data is as easy as clicking the data you need.

Download our free app



### Open a website

Download our [desktop app](#). Choose a site to scrape data from.



### Click to select data


Get data from multiple pages. Interact with AJAX, forms, dropdowns, etc.



### Download results

Access data via JSON, Excel and [API](#). Data is collected by our servers.



Double tap  to view all Parselhub holkeys.



Connecting to the Parselhub server.

If this persists for more than a few seconds, update your Parselhub extension and restart Firefox.



OSVDev | **JSON** | OSVDev Wiki | [Feedback](#)

A preview of your data will appear here





## Community

Download

[Contact Us](#)

Blog

Enhanced with Java profiler



OpenRefine (formerly Google Refine) is a powerful tool for working with messy data: cleaning it; transforming it from one format into another; and extending it with web services and external data.

OperaRefine is available in English, Chinese, Spanish, French, Russian, Portuguese (Brazil), German, Japanese, Italian, Hungarian, Hebrew, Filipino, Ogebuano, Tetzelos

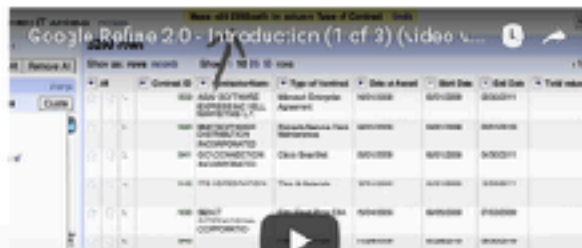
OpenRefine is supported by:

Google News Initiative

## Introduction to OpenRefine

## 1. Explore Data

OpenRefine can help you explore large data sets with ease. You can find out more about this functionality by watching the video below and going through [these articles](#).



<http://d3-media.blogspot.hk/2013/11/how-to-refine-your-data.html>

**END OF MORNING SESSION**