axess academy

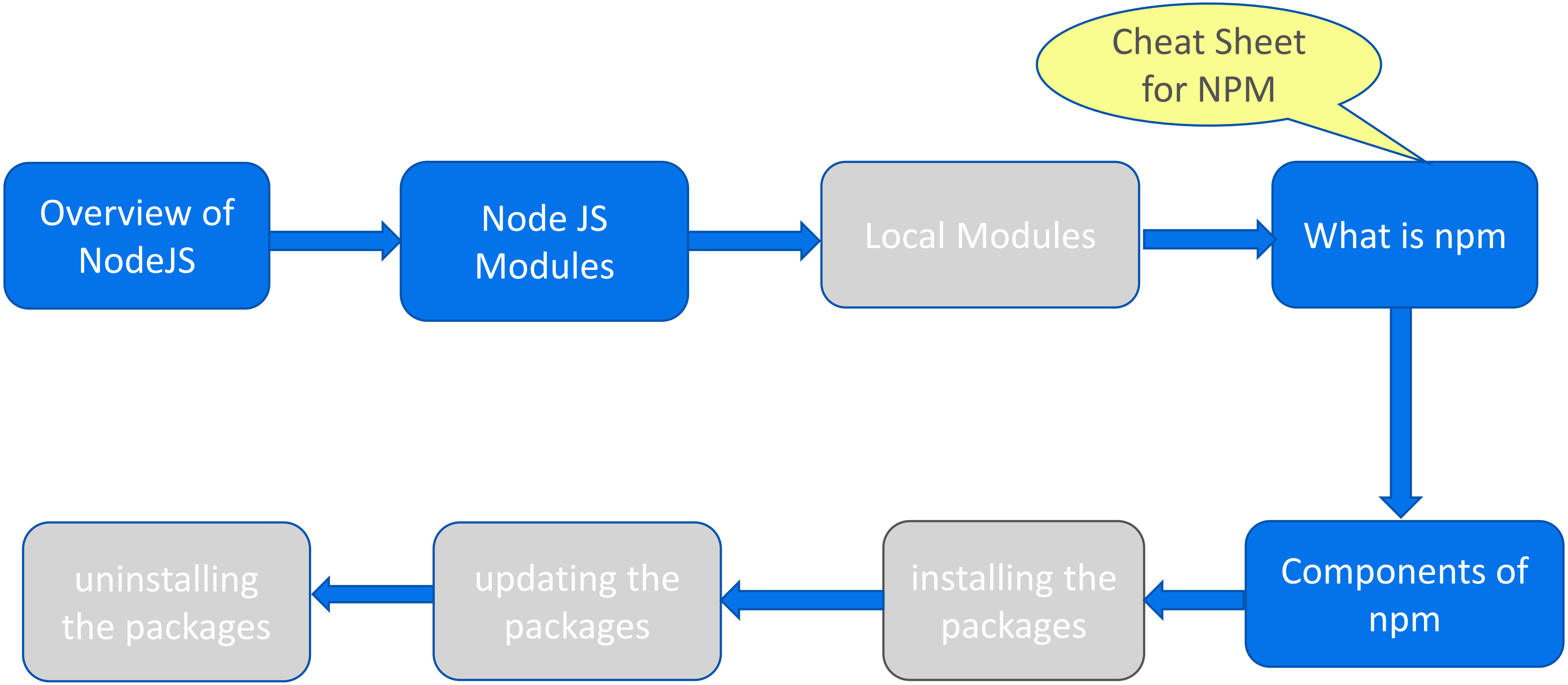
NodeJS, npm & JSON

standard chartered

27/05/2024

NodeJS/NPM module design

axess academy



S/W Requirements
Installer : NodeJS
Browser: Google Chrome

Overview of NodeJS

Node JS

- **Node.js** is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser.
- Node.js is used for development of server-side web applications.
- Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.
- Initial version was released in May 2009. The current version is 21.7.3 released April 2024.
- The official website is http://nodejs.org

NodeJS Modules

NodeJS Modules

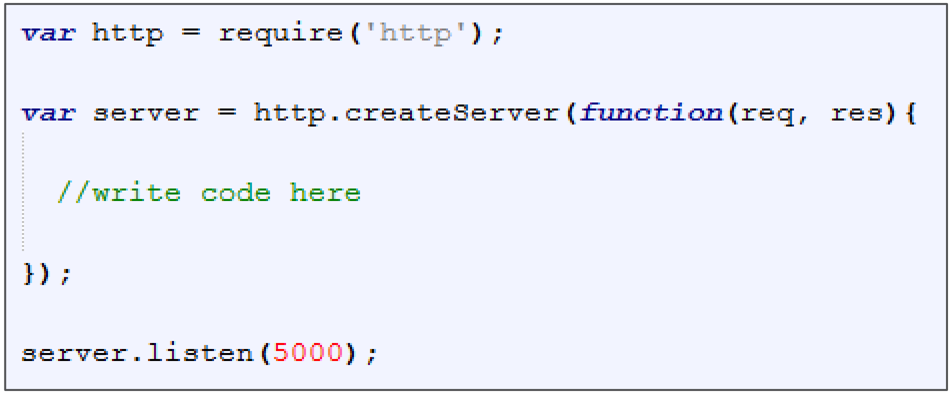
- Module in Node.js is a simple or complex functionality organized in single or multiple JavaScript files which can be reused throughout the Node.js application.
- Node.js includes three types of modules:

1. Core Modules (http, url, fs etc.)
2. Local Modules (User defined module)
3. Third Party Modules (React, Bootstrap, jquery etc.)

Core Modules

- NodeJS comes with core modules. Below is the list of in-built modules.

S No	Module Name	Description
1	http	http module includes classes, methods and events to create Node.js http server
2	url	url module includes methods for URL resolution and parsing
3	querystring	querystring module includes methods to deal with querystring
4	path	path module includes methods to deal with file paths
5	fs	fs module includes classes, methods and events to work with file I/O
6	util	util module includes utility functions for the programmers



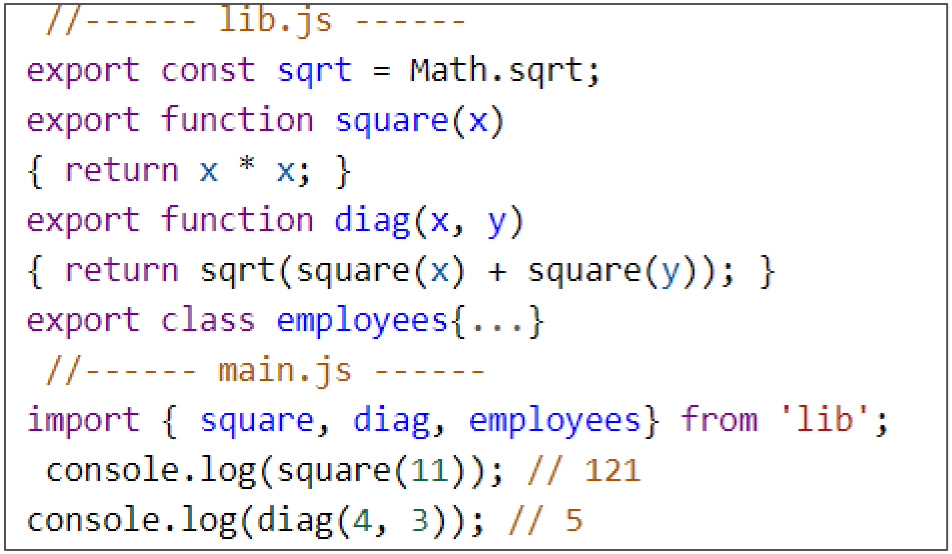
```
var http = require('http');

var server = http.createServer(function(req, res){

  //write code here

});

server.listen(5000);
```

Local Modules

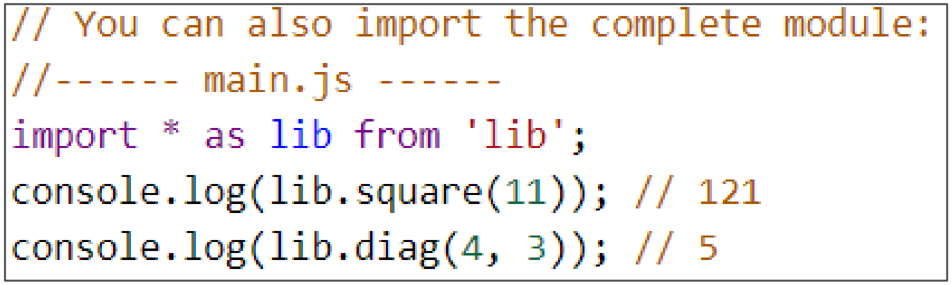
- Local modules are modules created locally in your Node.js application. These modules include different functionalities of your application in separate files and folders.
- The export statement is used when creating JavaScript modules to export classes, functions, objects or primitive values from the module so they can be used by other programs with the import statement.
- A module can import things from other modules using module specifiers like:

  Relative paths ('../model/user‘)

  Absolute paths ('/lib/js/helpers')
- Modules are singletons. Even if a module is imported multiple times, only a single “instance” of it exists.



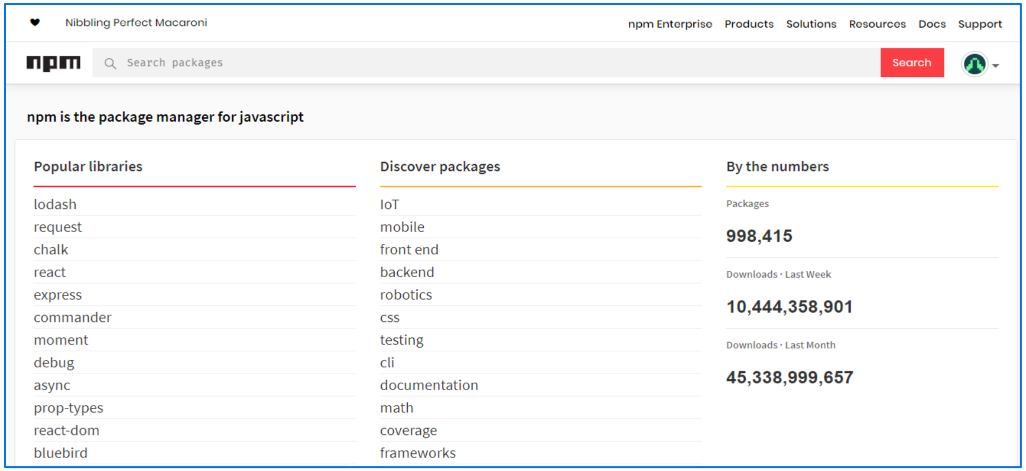
```
 //------ lib.js ------
export const sqrt = Math.sqrt;
export function square(x)
{ return x * x; }
export function diag(x, y)
{ return sqrt(square(x) + square(y)); }
export class employees{...}
 //------ main.js ------
import { square, diag, employees} from 'lib';
 console.log(square(11)); // 121
console.log(diag(4, 3)); // 5
```



```
// You can also import the complete module:
//------ main.js ------
import * as lib from 'lib';
console.log(lib.square(11)); // 121
console.log(lib.diag(4, 3)); // 5
```

Third Party Modules

- Third party modules are nothing but available in npm database like React, Bootstrap, jquery , webpack etc. and can be installed or uninstalled using npm.



npm is the package manager for javascript

Popular libraries

lodash
request
chalk
react
express
commander
moment
debug
async
prop-types
react-dom
bluebird

Discover packages

IoT
mobile
front end
backend
robotics
css
testing
cli
documentation
math
coverage
frameworks

By the numbers

Packages
998,415

Downloads · Last Week
10,444,358,901

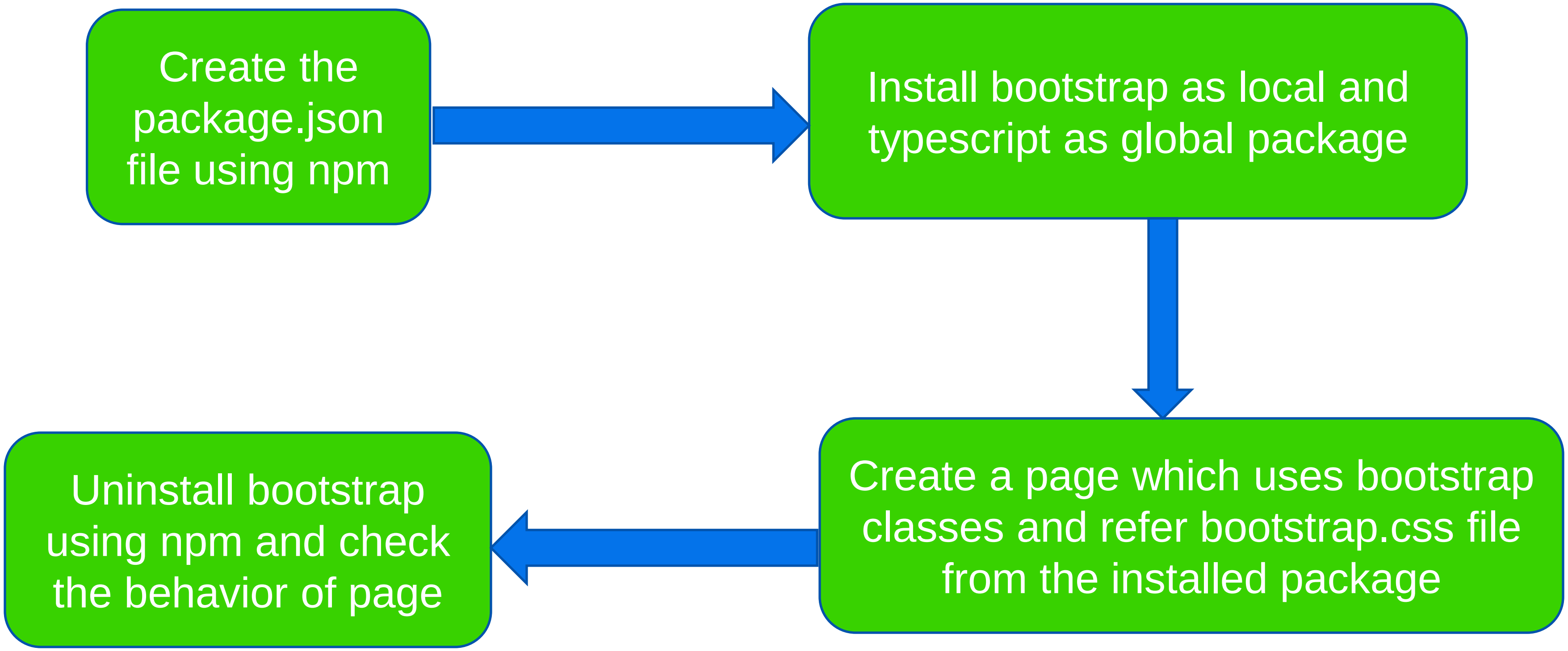
Downloads · Last Month
45,338,999,657

npm

Presenting the use case

Use case on npm

axess academy


Create the package.json file using npm

Install bootstrap as local and typescript as global package

Create a page which uses bootstrap classes and refer bootstrap.css file from the installed package

Uninstall bootstrap using npm and check the behavior of page

What is npm

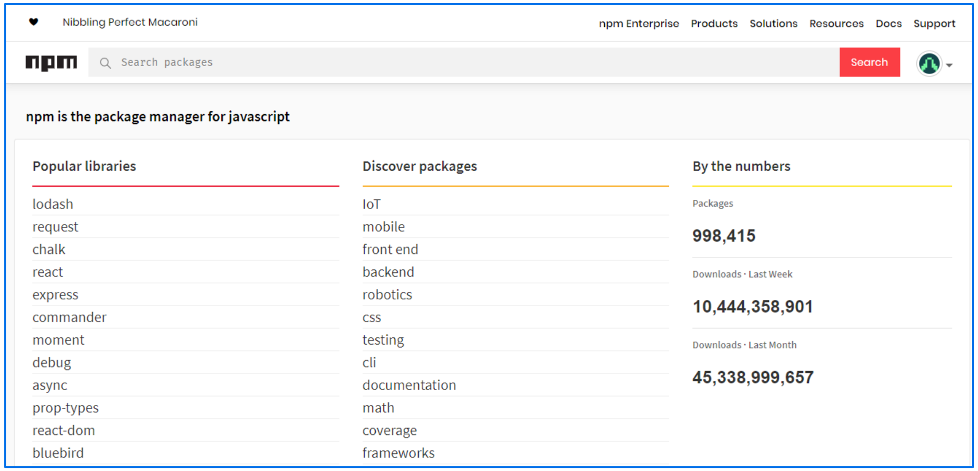
- **npm** (originally short for **Node Package Manager**) is a package manager for the JavaScript programming language.
- It allows users to consume and distribute JavaScript modules that are available on the registry.
- The registry is an online database of public and paid-for private packages.
- The registry is accessed via the npm command tool and the available packages can be browsed and searched via the npm website.
- First version was released in January 2010. The current version is 10.5.2 released in April 2024.
- Official website is www.npmjs.com

Components of npm

npm consists of three distinct components:

- the website : the primary way developers discover packages, set up profiles, and manage other aspects of their npm experience.
- the registry : a large database of information about packages
- the Command Line Interface (CLI): runs from a terminal. This is how most developers interact with npm.

npm website and registry



❤ Nibbling Perfect Macaroni

npm is the package manager for javascript

Popular libraries

- lodash
- request
- chalk
- react
- express
- commander
- moment
- debug
- async
- prop-types
- react-dom
- bluebird

Discover packages

- IoT
- mobile
- front end
- backend
- robotics
- css
- testing
- cli
- documentation
- math
- coverage
- frameworks

By the numbers

Packages

998,415

Downloads · Last Week

10,444,358,901

Downloads · Last Month

45,338,999,657

npm Installation

- npm is automatically installed with node.js installation. However, npm gets updated more frequently than Node.js, so be sure that you have the latest version.

S No.	Command	Description
1	npm -v	To test current version
2	npm install npm@latest -g	If the current version does not match the latest version. It will download the latest version

Install, Uninstall and Update of packages

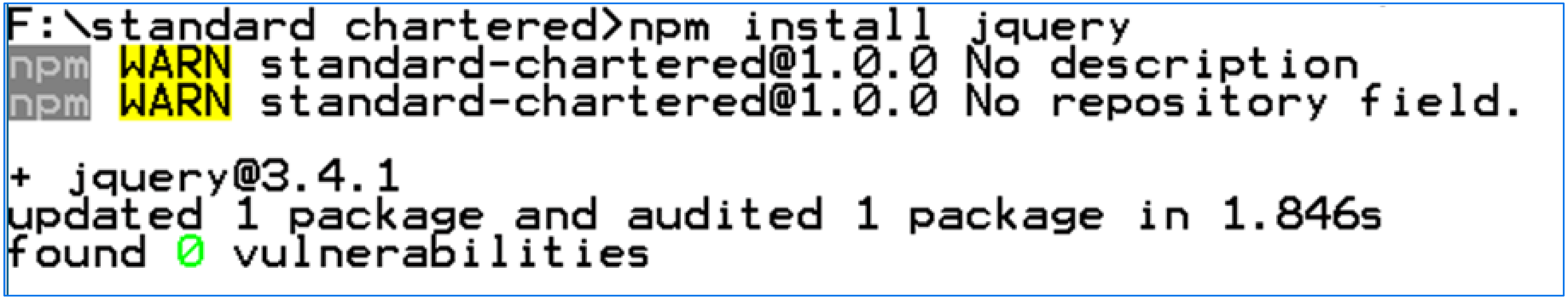
Installing the packages

Packages may contain multiple modules or a single module. Packages are used to group together files offering relevant tools.

- There are two ways to install npm packages: locally or globally.
- If you want to depend on the package from your own module, then install it locally. This is the choice you would use if you are using require statements, for example. jQuery, Bootstrap etc.

npm install <packagename>

This will download module at the location from where you are running the command.



```
F:\standard chartered>npm install jquery
npm WARN standard-chartered@1.0.0 No description
npm WARN standard-chartered@1.0.0 No repository field.

+ jquery@3.4.1
updated 1 package and audited 1 package in 1.846s
found 0 vulnerabilities
```

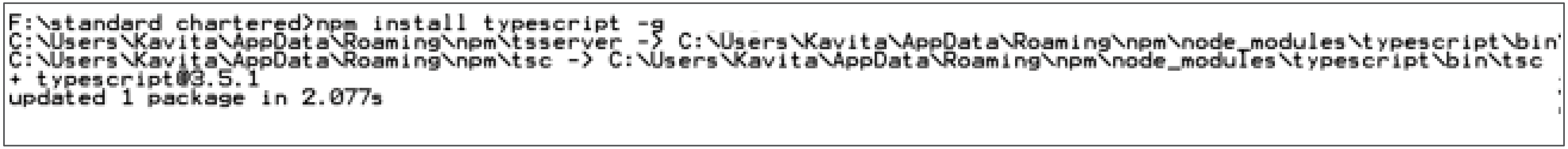
Installing the packages cont…

- Global Installation: If you want to use a package as a command line tool, then install it globally. This way, it works no matter which directory is current. For example grunt, typescript etc.

  npm install <packagename> -g

  This will download module at below location:

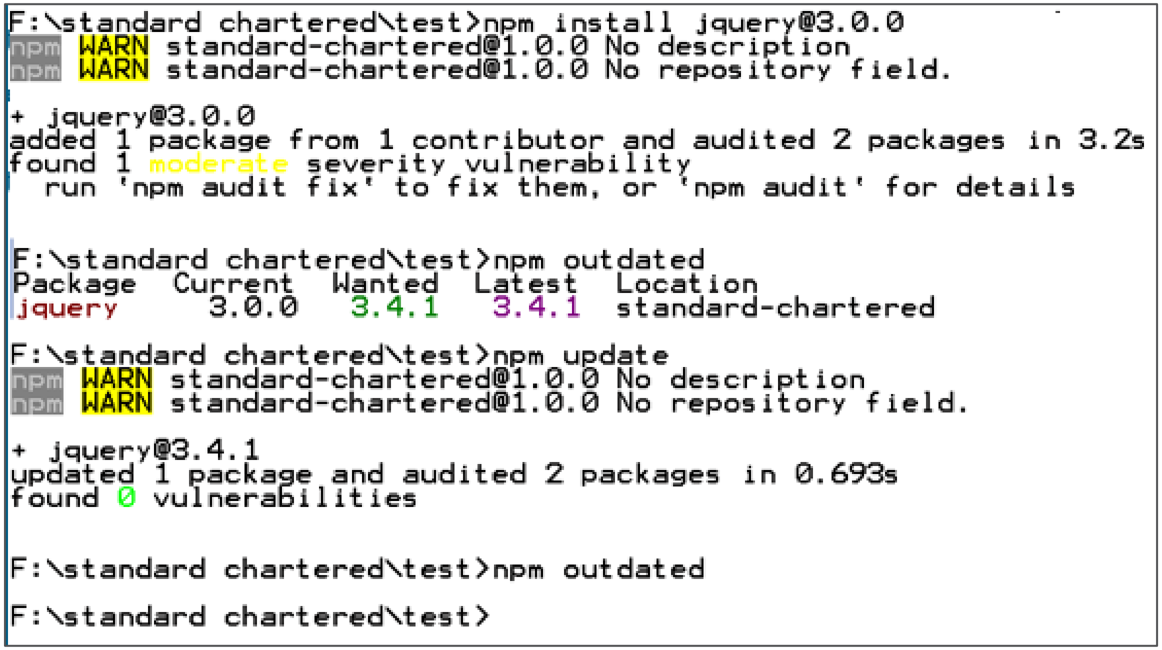
  C:\Users\<username>\AppData\Roaming\npm\node_modules



```
F:\standard chartered>npm install typescript -g
C:\Users\Kavita\AppData\Roaming\npm\tsserver -> C:\Users\Kavita\AppData\Roaming\npm\node_modules\typescript\bin
C:\Users\Kavita\AppData\Roaming\npm\tsc -> C:\Users\Kavita\AppData\Roaming\npm\node_modules\typescript\bin\tsc
+ typescript@3.5.1
updated 1 package in 2.077s
```

Updating the packages

- It's a good practice to periodically update the packages your application depends on. Then, if the original developers have improved their code, your code will be improved as well.
- npm outdated. There should be few outdated packages.
- npm update in the same directory as the package.json file of the application that you want to update.
- npm outdated. There should not be any results.

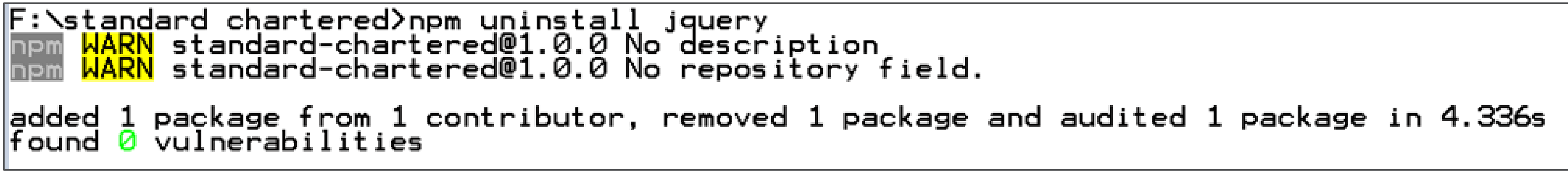


```
F:\standard chartered\test>npm install jquery@3.0.0
npm WARN standard-chartered@1.0.0 No description
npm WARN standard-chartered@1.0.0 No repository field.

+ jquery@3.0.0
added 1 package from 1 contributor and audited 2 packages in 3.2s
found 1 moderate severity vulnerability
  run 'npm audit fix' to fix them, or 'npm audit' for details


F:\standard chartered\test>npm outdated
Package  Current  Wanted  Latest  Location
jquery     3.0.0   3.4.1   3.4.1  standard-chartered

F:\standard chartered\test>npm update
npm WARN standard-chartered@1.0.0 No description
npm WARN standard-chartered@1.0.0 No repository field.

+ jquery@3.4.1
updated 1 package and audited 2 packages in 0.693s
found 0 vulnerabilities


F:\standard chartered\test>npm outdated

F:\standard chartered\test>
```

Uninstall the packages

- To remove a package from your node_modules directory, use:

  npm uninstall <package>



```
F:\standard chartered>npm uninstall jquery
npm WARN standard-chartered@1.0.0 No description
npm WARN standard-chartered@1.0.0 No repository field.

added 1 package from 1 contributor, removed 1 package and audited 1 package in 4.336s
found 0 vulnerabilities
```

Cheat Sheet references

Cheat Sheet for npm:

axess academy

JSON

standard chartered

27/05/2024

JSON

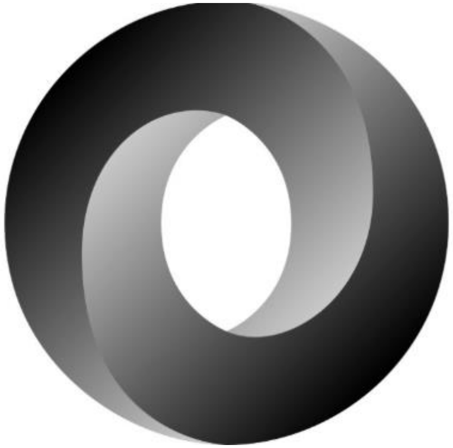
- **JSON** (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generat that is the reason it works as the replacement of XML
- JSON is a language-independent data format. It was derived from JavaScript, but as of 2017 many programming languages include code to generate and parse JSON-format data.
- The JSON syntax is a subset of the JavaScript syntax.
- The JSON format is almost identical to JavaScript objects.
- The file type for JSON files is ".json"
- In JSON, *keys* must be strings, written with double quotes.
- Syntax:

  JSON Object:

  {"name":"John", "age":31, "city":"New York"}

  JSON array of objects:

  {“arr”: [{"name":"John","age":30},{"name":"Jack","age":25}] }



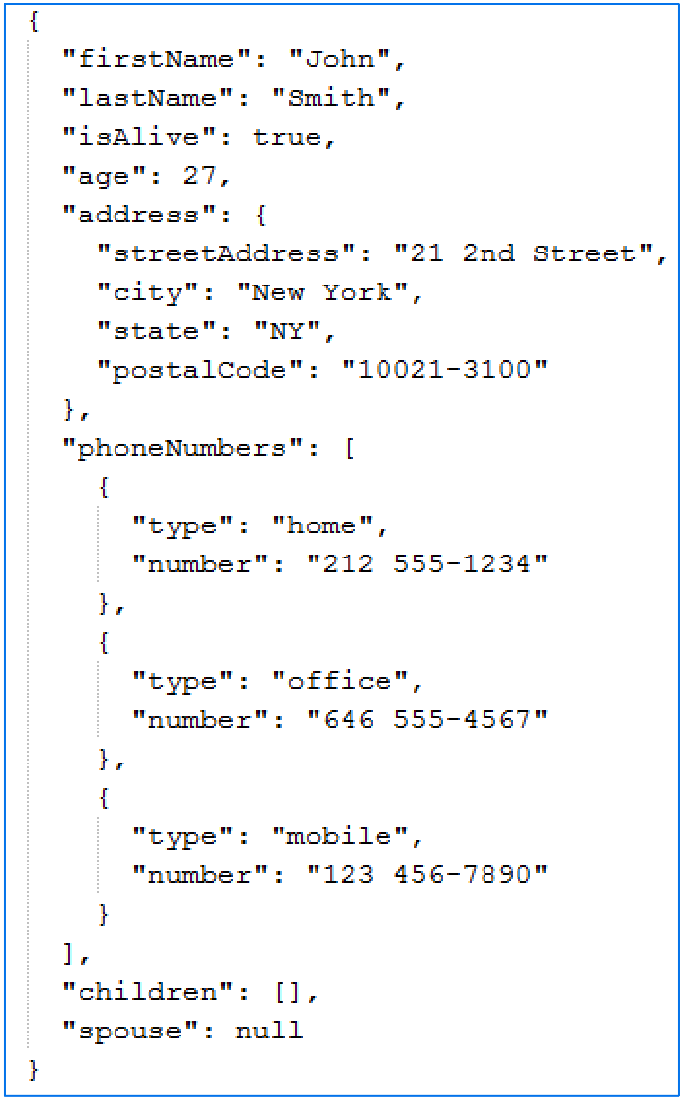
JSON value data types

In JSON, values must be one of the following data types:

- a string
- a number
- an object (JSON object)
- an array
- a boolean
- *null*

JSON values **cannot** be one of the following data types:

- a function
- a date
- *undefined*

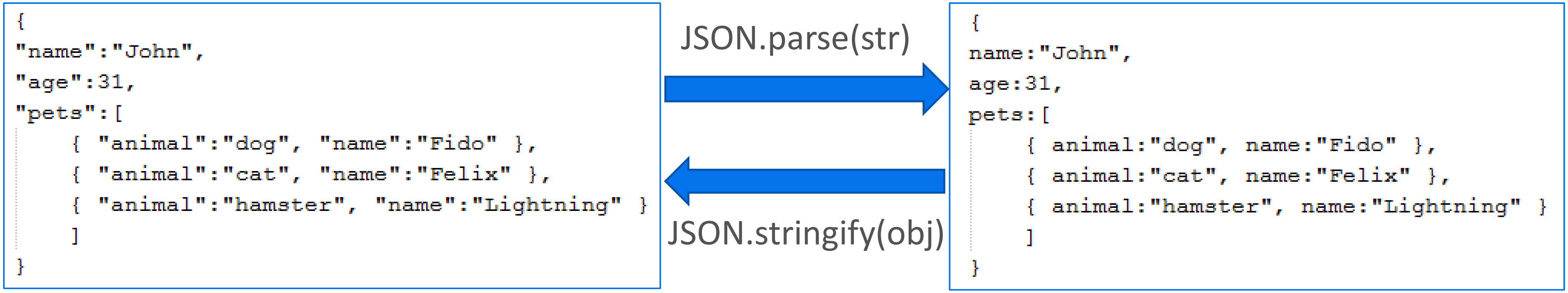


```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```

sample.json

JSON Methods

- A common use of JSON is to exchange data to/from a web server.
- When receiving data from a web server, the data is always a string.
  - Parse the data with JSON.parse() and the data becomes a JavaScript object.
- When sending data to a web server, the data has to be a string.
  - Convert a JavaScript object into a string with JSON.stringify() method.



```
{
"name":"John",
"age":31,
"pets":[
    { "animal":"dog", "name":"Fido" },
    { "animal":"cat", "name":"Felix" },
    { "animal":"hamster", "name":"Lightning" }
    ]
}
```

JSON.parse(str)

JSON.stringify(obj)

```
{
name:"John",
age:31,
pets:[
    { animal:"dog", name:"Fido" },
    { animal:"cat", name:"Felix" },
    { animal:"hamster", name:"Lightning" }
    ]
}
```

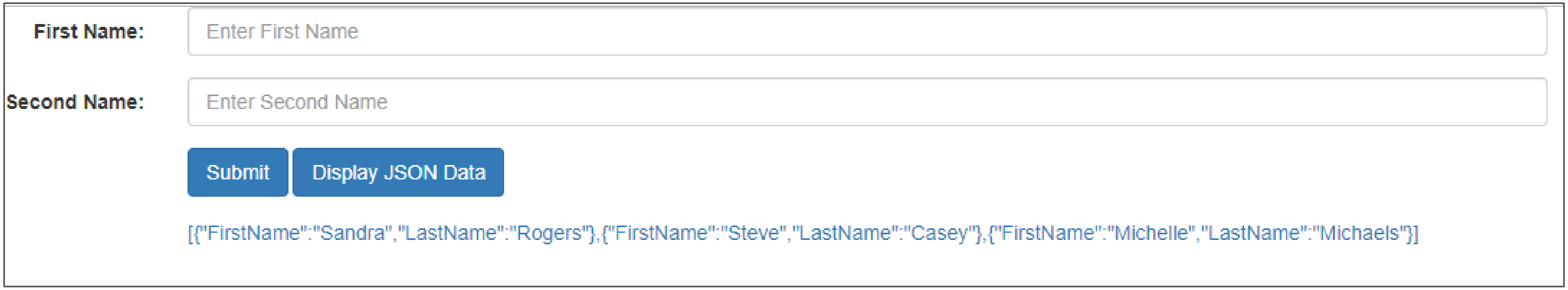
JSON data (str)

Java Script Object (obj)

Let us work on use case on JSON

Presenting the Use case on JSON

- Design the below page using bootstrap. Once the user clicks on submit button the entries get saved into JSON array and on click of Display JSON data , it should display the JSON array down.



First Name: Enter First Name

Second Name: Enter Second Name

```
[{"FirstName":"Sandra","LastName":"Rogers"},{"FirstName":"Steve","LastName":"Casey"},{"FirstName":"Michelle","LastName":"Michaels"}]
```

Cheat Sheet references

Cheat Sheet for JSON : JSON Cheat Sheet

Thank You