**Node.js Modules**

In Node.js, **Modules** are the blocks of**encapsulated code** that communicate with an **external application on the basis of their related functionality**. Modules can be a single file or a collection of multiple files/folders. The reason programmers are heavily reliant on modules is because of their reusability as well as the ability to break down a complex piece of code into manageable chunks.

**We will discuss the different types of Node.js Modules:**

**Table of Content**

* [Core Modules](https://www.geeksforgeeks.org/node-js-modules/?ref=lbp#core-modules)
* [Local Modules](https://www.geeksforgeeks.org/node-js-modules/?ref=lbp#local-modules)
* [Third-party modules](https://www.geeksforgeeks.org/node-js-modules/?ref=lbp#thirdparty-modules)

**Core Modules**

Node.js has many built-in modules that are part of the platform and come with [Node.js](https://www.geeksforgeeks.org/nodejs/)installation. These modules can be loaded into the program by using the **required** function.

**Syntax:**

const module = require('module\_name');

The require() function will return a JavaScript type depending on what the particular module returns. The following example demonstrates how to use the Node.js http module to create a web server.

JavaScript

**const** http = require('http');

http.createServer(**function** (req, res) {

res.writeHead(200, { 'Content-Type': 'text/html' });

res.write('Welcome to this page!');

res.end();

}).listen(3000);

In the above example, the require() function returns an object because the Http module returns its functionality as an object. The function http.createServer() method will be executed when someone tries to access the computer on port 3000. The res.writeHead() method is the status code where 200 means it is OK, while the second argument is an object containing the response headers. The following list contains some of the important core modules in Node.js:

| **Core Modules** | **Description** |
| --- | --- |
| http | creates an HTTP server in Node.js. |
| assert | set of assertion functions useful for testing. |
| fs | used to handle file system. |
| path | includes methods to deal with file paths. |
| process | provides information and control about the current Node.js process. |
| os | provides information about the operating system. |
| querystring | utility used for parsing and formatting URL query strings. |
| url | module provides utilities for URL resolution and parsing. |