Botlhale Village

Working together for ICT innovation and growth in Africa

BELGIUM CAMPUS iTversity

It's the way we're wired

WPR252

Modules

Modules

- Node.js treats each JavaScript file as a separate module
- For instance, if you have a file containing some code and this file is named xyz.js, , then this file is treated as a *module* in Node, and you can say that you've created a module named xzy.
- Let's take an example to understand this better.
- You have a file named circle.js which consists of the logic for calculating the area & the circumference of a circle of a given radius
- Before executing the code written inside a module, Node takes the entire code and encloses it within a function wrapper.

Modules

BELGIUM CAMPUS iTversity
It's the way we're wired

```
(function(exports, require, module, __filename, __dirname) {
  // Module code actually lives in here
});
```

- The five parameters-: exports, require, module, __filename, __dirname are available inside each module in Node.
- These parameters provide valuable information related to a module.
- Node modules allow you to select what functions and variables from the included file are exposed to the application.
- If the module is returning more than one function or variable, the module can specify these by setting the properties of an object called exports.

- If the module is returning a single function or variable, the property module .exports can instead be set.
- Modules can then be published to the npm (Node Package Manager) repository, an online collection of ready-to-use Node modules, and shared with the Node community.
- Modules can either be single files or directories containing one or more files.
- If a module is a directory, the file in the module directory that will be evaluated is normally named index.js

Types of Module

1. Core module: Modules that come shipped with Node.js, e.g. https, os, fs, net, etc.
2. Third-party module: Modules that you install from any package manager. We use these modules to accomplish or simplify any existing task. For example, to simplify our web API development we use express, or to deal with date and time we use moment.
3. Local module: These are the modules that we create for our own use. These modules basically consist of core business logic of our code.

__filename

This is a variable that contains the absolute path of the current module.

Given two modules: a and b, where b is a dependency of a and there is a directory structure of:

- /User/home/node_blog/a.js
- /User/home/node_blog/node_modules/b/b.js

So, if we do console.log(__filename)within b.js, we will get /User/home/node_blog/node_modules/b/b.js. If we do console.log(__filename) within a.js, we will get /User/home/node_blog/a.js.

__dirname

BELGIUM CAMPUS iTversity
It's the way we're wired

- The directory name of the current module. This is the same as the path.dirname() of the __filename.
- So, for the above modules, a.js and b.js.
- If we do console.log(__dirname) within b.js, we will get /User/home/node_blog/node_modules/b/ and in a.js, we will get /User/home/node_blog/.

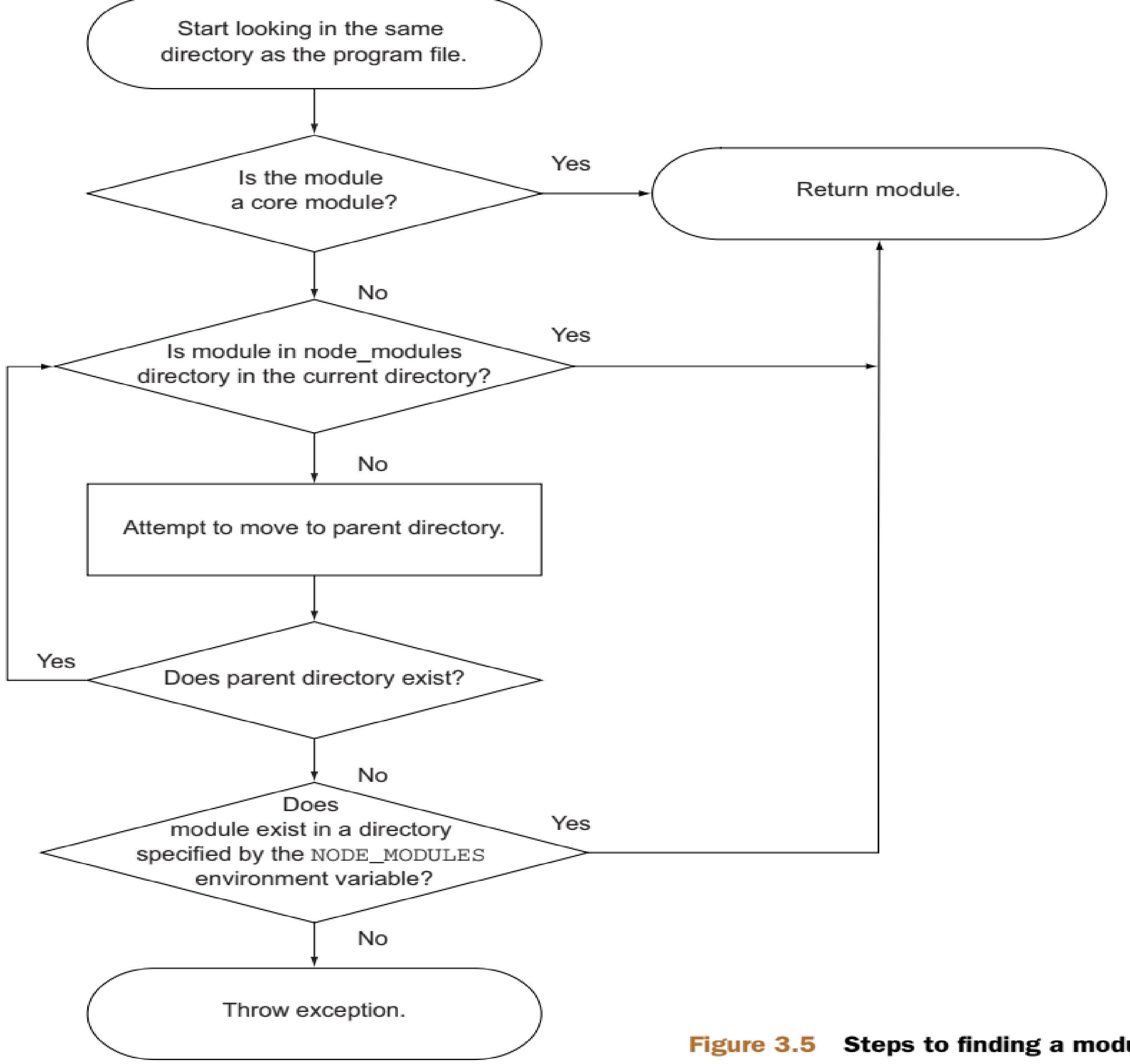
Start looking in the same directory as the program file.
Is the module a core module?
Yes
Return module.
No
Is module in node_modules directory in the current directory?
Yes
No
Attempt to move to parent directory.
Yes
Does parent directory exist?
No
Does module exist in a directory specified by the NODE_MODULES environment variable?
Yes
No
Throw exception.

Figure 3.5 **Steps to finding a module**

BELGIUM CAMPUS iTversity

It's the way we're wired