greatlearning
Power Ahead

Node JS

Agenda

- Prerequisites
- What is Node JS?
- Features of Node JS
- Installing Node JS
- Editors
- Our First Program
- REPL
- Global Objects
- Functions
- Modules

Prerequisites

What is HTML?

- Stands for HyperText Markup Language
- HyperText: Link between web pages
- Markup Language: Text between tags which defines structure
- It is a language to create web pages
- HTML defines how the web page looks and how to display content with the help of elements
- It forms or defines the structure of our Web Page
- Need to save your file with .html extension

Features Of HTML

- Learning curve is very easy (easy to modify)
- Creating effective presentations
- Adding Links wherein we can add references
- Can display documents on platforms like Mac , Windows, Linux etc
- Adding videos, graphics and audios making it more attractive
- Case insensitive language

What is JavaScript?

- It is a verb of the web page that defines all the actions to be performed on a webpage
- Its an object oriented programming language that uses JIT compiler
- It is everywhere and all web browsers are installed with it
- JS application ranges from web development, mobile development etc
- JS is easy, simple and very compatible with HTML-CSS
- It is must to have skill for any software engineer role

What is Node JS?

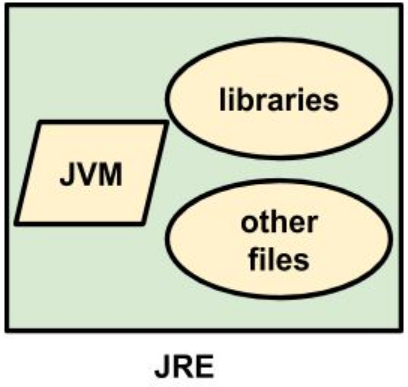
What Node JS is not?

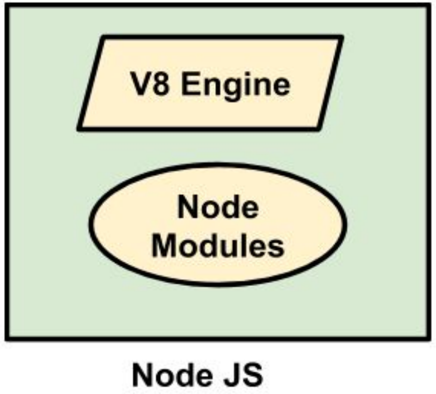
- It is not a programming language
- It is not a framework
- It is not a library

What is Node JS?

- Platform to run JavaScript on the server i.e outside the browser
- It is a runtime environment which includes everything to run/execute a JavaScript program
- It is a JavaScript runtime built on Chrome's V8 JavaScript engine

Comparison of RunTime Environment

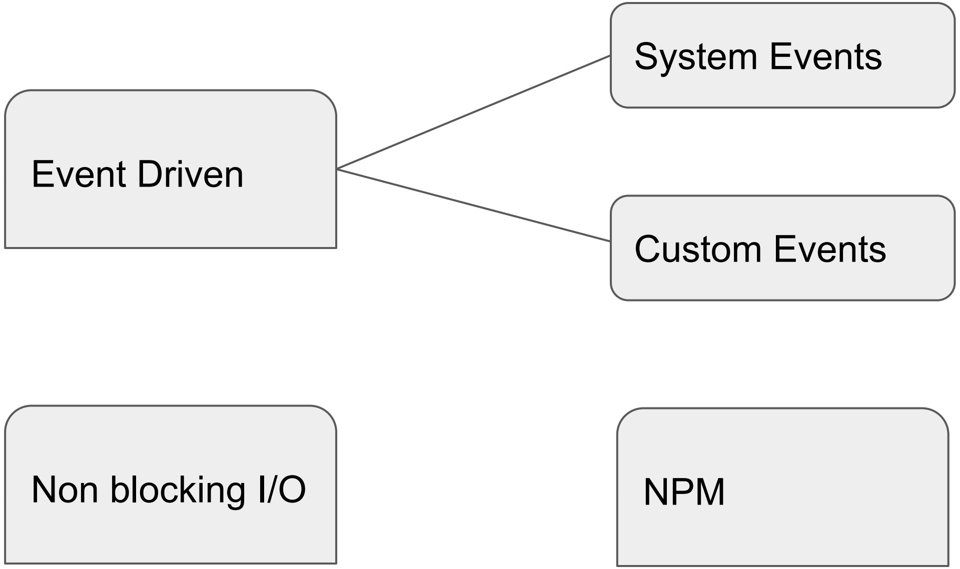




What else is there?

- We can run JS programs on our own machines like standalone applications
- Node Js acts like other scripting languages, for example- Python, Java, Php etc
- Node Js can read, write, delete, etc files on the server
- Node Js can add, delete, modify data in database

Why Node JS?



Event Driven

System Events

Custom Events

Non blocking I/O

NPM

Features Node JS

Features

- Faster
- Easier
- Efficient
- Lightweight
- Scalable

Installing Node JS

How to install Node JS?

- https://nodejs.org/en/
- Click and install
- To check
  1. Open command prompt
  2. node -v
     - Give the version of node js installed

Editors

Which editor should you choose?

- Visual Studio Code
- Sublime Text
- Atom
- CodeMix
- Brackets

Our First Program

Writing Our First Program

- Open Sublime
    - Write: console.log(“Hello World”);
    - Save with .js extension

- Open cmd
    - Run command node name_of_file.js

- console.log(): display message on console.

REPL

What is REPL?

- Stands for Read, Eval, Print and Loop
- Press node without file_name
  - R: read the input, parse in DS in JS, then store it
  - E: evaluates the DS
  - P: prints the result
  - L: loops the above statements

- Simple expression
- Use variables
- _ is used get previous result
- To exit ctrl_c twice

Variables

- Variable is a name of a memory location where the data is stored.
- **Syntax** : var varname = val;
- In JS, variable is defined using var keyword
    - var name =”Faizan”;
    - var roll_no=27;

Naming Convention

- Variable names should begin with a letter, $ or an underscore(_).
- First character can be then followed by any combination of letters or digits.
- A variable name cannot be the same as any keyword as they are reserved for special purposes.
- Variable names case sensitive.
    - Eg: valid- apple, _name, $address1
    - Eg: invalid- 123apple, *roll, @email

Data Types

- JS is a dynamically typed language and does not need to specify data type explicitly of the variable.
- There are 5 primitive data types
  - Number, String, Boolean, Undefined, Null
    - **Example**:

```
var num=23;
var string="hello";
var boolean=true;
var undef;
var nullValue=null;
```

```
num
23

string
"hello"

boolean
true

undef
undefined

nullValue
null
```

Global Objects

What are Global Objects?

- console.log()
- setTimeout(function (){

  ```
      Console.log(“Prints after one sec”);
  }, 1000);
  ```

- setInterval(function (){

  ```
      Console.log(“Prints after every one sec”);
  }, 1000);    // ctrl+c
  ```

- __dirname: name of directory in which our file is located
- __filename : includes file name as well

Functions

What is a Function?

- It is referred to a code snippet that generally perform some operation
- It helps in modularization of code
- Enables reusability of it as and when required
- Make implementation and debugging efficient and easy
- Function can be 2 ways in general to define functions
  - Using function keyword
  - Function expression

Using Function Keyword

- Using function keyword
- Starts will function keyword
- Example

```
function happy(){
    console.log("I am grateful");
     }
```

- Calling:
  - happy();

Using Function Expression

- Starts with a var and assignment operator
- Example

```
var faith=function(){
console.log("hope");
}
```

- Calling: faith()

Modules

How to write Modules?

External file:

```
        var name = "Vicky";
        module.exports = name;
```

Using it in a different file:

```
     var name = require('./everything.js');
    console.log(name);
```

Thank You