# 为什么要学习Node.js？

具有服务端开发经验更好

后端（back-end）

目的：打开服务端的黑盒子，只有了解服务端才可以更加好的配合服务端开发人员进行协同开发

基本的网站开发能力（服务器，前端，运维部署）

打开服务端的黑盒子,服务端入口：java,php,python,.Net,Ruby,node.js,c##

# Node.js是什么？

Node.js是javaScript的运行环境，可以解析和执行javaScript代码，也就是说现在的javaScript可以完全脱离浏览器来运行，一切归功于Node.js

Node.js不是语言，不是框架，不是库，是一个运行平台，等同于浏览器

* 浏览器中的javaScript：

1. EcmaScript

* 基本语法
* If
* Var
* Function
* Object
* Array

1. BOM(浏览器对象模型)

* Widow

1. DOM（文档对象模型）

* Document
* Node.js中的javaScript:（web服务器开发）

1. **没有BOM,DOM**
2. EcmaScript
3. 在Node.js这个javaScript执行环境中为JavaScript提供了一些服务器级别的API

* 文件的读写
* 网络文件的构建
* 网络通信
* HTTP服务器

特点：事件驱动，非阻塞IO模型（异步），轻量和高效

# Node.js可以做什么？

1. web服务器后台
2. 命令行工具

* Npm(node开发)
* Git（C语言开发）
* Hexo（node开发）
* 。。。。

1. 对于前端开发而言，接触Node最多的是他的命令行工具，主要使用第三方工具

简单的命令行操作：

* Cd（切换到某一级目录下）
* Dir(列出目录)
* Ls(列出目录)
* Mkdir
* Rm(删文件)

参考资源

Node中文文档：<http://nodejs.cn>

Node社区：<https://cnodejs.org/>

# 这门课可以学什么？

## B/S编程模型

* Browser-Server
* Back-end
* 任何服务端技术这种BS编程模型都是一样，和语言无关
* Node只是作为我们学习BS编程模型的一个工具而已

## 模块化编程

* RequireJS
* SeaJS
* @import(‘文件路径’)

## Node常用API

## 异步编程

* 回调函数
* Promise
* Async
* generator

## Express开发框架

## Ecmascript6

## .......

# BOM中的对象

## Window对象：

*是整个BOM的核心，所有对象和集合都以某种方式回接到window对象。Window对象表示整个浏览器窗口，但不必表示其中包含的内容。*

## Document对象：

*实际上是window对象的属性。这个对象的独特之处是唯一一个既属于BOM又属于DOM的对象。从BOM角度看，document对象由一系列集合构成，这些集合可以访问文档的各个部分。*

## Location对象：

*它是window对象和document对象的属性。Location对象表示载入窗口的URL，此外它还可以解析URI.*

## Navigator对象：

*Navigator包含大量Web浏览器相关的信息。各种浏览器支持该对象的属性和方法不尽相同。*

## Screen对象：

*通过其可以获取用户屏幕相关的信息*

BOM是浏览器对象模型，用来获取或设置浏览器的属性、行为，例如：新建窗口、获取屏幕分辨率、浏览器版本号等。 DOM是文档对象模型，用来获取或设置文档中标签的属性，例如获取或者设置input表单的value值。 BOM的内容不多，主要还是DOM。 由于DOM的操作对象是文档（Document），所以dom和浏览器没有直接关系。

BOM的核心是window，而window对象又具有双重角色，它既是通过js访问浏览器窗口的一个接口，又是一个Global（全局）对象。这意味着在网页中定义的任何对象，变量和函数，都以window作为其global对象。

document对象

document对象：实际上是window对象的属性，<br data-filtered="filtered">document == window.document为true，是唯一一个既属于BOM又属于DOM的对象

<br data-filtered="filtered">document.lastModified //获取最后一次修改页面的日期的字符串表示

<br data-filtered="filtered">document.referrer //用于跟踪用户从哪里链接过来的

<br data-filtered="filtered">document.title //获取当前页面的标题，可读写

<br data-filtered="filtered">document.URL //获取当前页面的URL，可读写

<br data-filtered="filtered">document.anchors[0]或document.anchors["anchName"] //访问页面中所有的锚

<br data-filtered="filtered">document.forms[0]或document.forms["formName"] //访问页面中所有的表单

<br data-filtered="filtered">document.images[0]或document.images["imgName"] // 访问页面中所有的图像

<br data-filtered="filtered">document.links [0]或document.links["linkName"] //访问页面中所有的链接

<br data-filtered="filtered">document.applets [0]或document.applets["appletName"] //访问页面中所有的Applet

<br data-filtered="filtered">document.embeds [0]或document.embeds["embedName"] //访问页面中所有的嵌入式对象

<br data-filtered="filtered">document.write(); 或document.writeln(); //将字符串插入到调用它们的位置