今天学习的内容

1:复习bootstrap重点知识

2:作业

3:今天的目标

3.1:(准备工作)NodeJS 下载并且安装

3.2:一道面试题 用户在浏览器中输入 www.taobao.com 直到看到页面之间发生了什么?

(1)操作系统访问网络上的DNS服务器，把域名转换IP地址

(2)浏览器获取返回ip地址并且向ip地址发送HTTP请求

(3)web服务器接收并解析请求消息,查找指定资源,可能访问数据库,构造并返回http响应消息

(4)浏览器接收并解析响应消息

(5)浏览器缓存接收响应消息，并解析和渲染响应的内容

3.3:静态网页和动态网页

静态网页:网页内容任何人在任何时间访问都是不变的.

HTML/CSS/JS/视频...

动态网页:网页内容不同人在不同时间访问可能不同的

JSP/PHP/ASP.NET/Node.JS

3.4:NodeJS 概述(重点&难点)

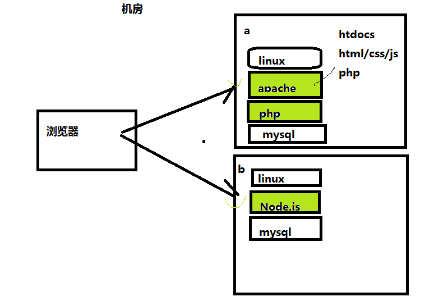
JSP=HTML+java :功能强可靠，适合大型企业级项目

PHP=HTML+PHP :简单易于，适合互联网项目

ASP.NET=HTML+C# :易用适合windows 平台

Node.JS=HTML+"JS" :性能好,适合于服务器端IO密集型项目，不适合CPU密集项目.

|  |  |
| --- | --- |
| CPU密集型项目 | 滴滴打车,天气预报 |
| IO密集型项目 | 输入输出(学子商城) |



概念:Node.JS不是JS,是一种服务器开发平台,它的竞争对手是PHP/JSP/ASP. 历史上第一次有一种语言可以通吃前后台.

官网:www.nodejs.org

版本:(16初)0.12 (16年中)4.x (16底)6.x

LTS:Long Term Support 长期稳定支持版本

Current:最新版本

安装时注意事项:安装目录(不要有中文,不要有空格)

3.5:NodeJS 运行模式

(1)交互模式--一般用于临时测试

REPL:Read Evaluate Print loop,输入一行代码执行一行

注意:交互模式自带输出功能,不必写 console.log();

node 回车 #进入交互模式

.exit #退出交互模式

(2)脚本模式--正式项目中使用的方式

把要执行的所有语句，编写在一个文本文件中(后缀任意，没有也行),一次性提交给node解释器.

node 完整路径名/x.js 回车

提示:只要安装完成Node.js重启一个WebStorm,ws可以自动检测node.js ,右键运行

练习:创建01.js文件，打印九九乘法表，以脚本模式运行,

ws中运行

3.6:如何自学一门语言 Node.JS

(1)了解背景

---百度百科 Node.JS (IO非阻塞/事件驱动)

(2)搭建开发环境

下载并安装nodejs.msi

(3)数据类型--(多)

|  |
| --- |
| 前端JS数据类型  (1)基本/原生/值类型  number/string/boolean/null/undefined  (2)引用类型  ES对象;String;Math;Date;Object;Function;RegExp;Error  *BOM:window;document;history;screen;location;event*  *DOM:Node/Element/Attr/*  用户自定义:{} |
| 后端Node.JS数据类型  (1)基本/原生/值类型  string/number/boolean/null/undefined  (2)引用/对象类型  ES:String/Number/Boolean/Math/Data/RegExp/Object..  用户自定义对象:{}  NodeJS原生第三方对象库:几百 十多万 |

(4)变量和常量

var age = 20;

const PI = 3.14;

(5)运算符

算术运算符 + - \* / % ++ --

比较运算符 < > <= >= == === !==

逻辑运算符 && || !

位运算符 << >> ^= !

三目运算符 ? :;

赋值运算符 += -= \*= /+ %=

特殊运算符 typeof .

(6)逻辑结构

循环: while do..while for(;;); for(..in..) for(..of..)

练习:03.js 声明一个保存5个学生成绩的数组,使用三种

循环依次打印出每个成绩值

选择结构:if..else switch..case

练习:创建一个变量 var path = "/path";使用二种选择结构,判断path的值为哪种(/index,/search/,/login)，调用不同的函数

(7)通用小程序

九九乘法表,100以内质数,数组排序...

练习:打印出100以内质数 14:34--14:36

(8)函数和对象

(9)常用组件,第三方工具

(10)实际小项目

3.7:Nodejs中特有的概念--模块(重点&难点)

|  |
| --- |
| Modal:模态框 Model:模型 Module:模块 |

一个web项目功能可以分为很不同的"模块":如：商品管理模块，用户管理模块，支付模块，促销模块...

Node.JS按照不同的功能，可以把函数，对象分别保存不同文件或目录下“文件或目录称为模块”

Node.js为每个模块创建独立构造函数，解释会(自动)添加到每个.js文件中.

05.js

(function (exports, require, module, \_\_filename, \_\_dirname){

exports:{} 用于声明向外部出自己成员

require:fn 用于导入其它的模块，并且创建指定模块对象

module:当前模块对象

\_\_filename

\_\_dirname

var i = 10;

console.log(document);

//自己编写程序内容

});

每个模块都可以使用自己的require()函数引入另一个模块--底层本质就是创建指定模块的一个对象实例.

var m = require("./模块文件名");

每个模块可以使用exports对象向外导出/公开一些自己的成员供其它模块使用

exports.成员名 = 成员值.

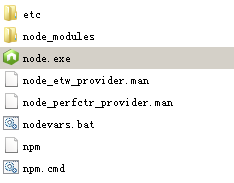
|  |
| --- |
| Node.JS中有二个对象可以向外公共数据;  exports和module.exports  Node.js底层代码: exports = module.exports;  所以：  若只是给exports对象添加新成员，则等于给module.exports添加新成员,但是若改了exports指向，则不同产生实质作用.  exports.age = 19;  module.exports.age = 19;  但是如果:  exports = {name:"tom",age:19} |

练习:创建一个文件模块:08\_circle.js 其中定义常量PI，声明一个方法getSize(r),根据传入圆形的半径，返回其面积，以及一个方法 getPerimiter(r);根据传入圆形半径，返回周长，再创建一个应用程序 09\_app.js 引入08\_circle.js模块，调用其公开的两个成员方法.

练习:创建一个模块:10\_ArrayUtil.js包含一个方法sum(arr),返回指定数组中所有数值的和，一个方法avg(arr)返回指定数组中的所有数组的平均值，使用module.exports导出上述两个成员

再写11\_app.js 模块，调用其公开的方法

3.8:Node.js模块分类



(1)Node.JS官方提供的模块--安装在解释器内器

require("模块名");

(2)第三方编写模块

(3)用户自定义编写模块

exports.x = require("./模块文件名");

3.9:Node.js 预定义模块---Global

该模块提供可以直接使用，而无需require(global);

console

setInterval();

setTimeout();

setImmediate(fb); 等价于setTimeout(0,fn);

3.10:练习:仿写Node.js提供一模块:URL

创建模块文件 MyUrl,向外导出一个方法 resolve(url);

该方法接收参数:

形如: http://www.jd.com:8080/ad/index?uname=tom&pno=3

返回对象:形如

{

protocol:"http",

server:"www.jd.com",

path:"/ad/index",

query:"uname=tom&pno=3"

}

再编写一个主模块:引入上述模块并调用

运算出错: nodejs路径不正确

(1)File->Setting->Lanaguages->Node.js and NPM

(2)Node interpreter:[c:\program Files/nodejs/node.exe][...]

作业1:授权己有拥角色/未拥有角色

作业2:url模块: query {uname:"tom",pno:3}