今天学习的内容

1:复习总结上周重点

京东项目【登录模块/产品列表/添加至购物车/购物车/查/修改】

2:cookie

cookie:小甜饼

作用:保存(指定保存时间)用户添加数据.默认情况**会话**结束数据就删除了.

会话:是一个**操作过程**:当用户打开浏览器请求指定页面(会话开始),下一步操作,关闭浏览器(会话结束).

操作方式

(1)保存数据至cookie document.cookie='uid=10';

(2)从cookie中获取数据 var rs = document.cookie;

#注意事项:cookie数据保存在客户端浏览器，不要将**安全性高**数据保存cookie

#cookie保存数据通常是:用户编号/用户喜欢产品编号/昵称..

3:NodeJS

1:阿里面试题:用户在浏览器中输入 www.taobao.com 直到看面之间发生了什么?

(1)操作系统(window)访问网络上DNS服务器，把或域名转换IP址地

DNS [www.taobao.com 60.28.242.250]

(2)浏览器发起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，构建并返HTTP响应消息.

(4)浏览器接收并解析响应消息

(5)浏览器(**缓存**)接收到解析内容，并解析和渲染响应内容.

2: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/Flash/音频/视频..

动态网页:网页内容任何人在任何时间访问可能是不同的.

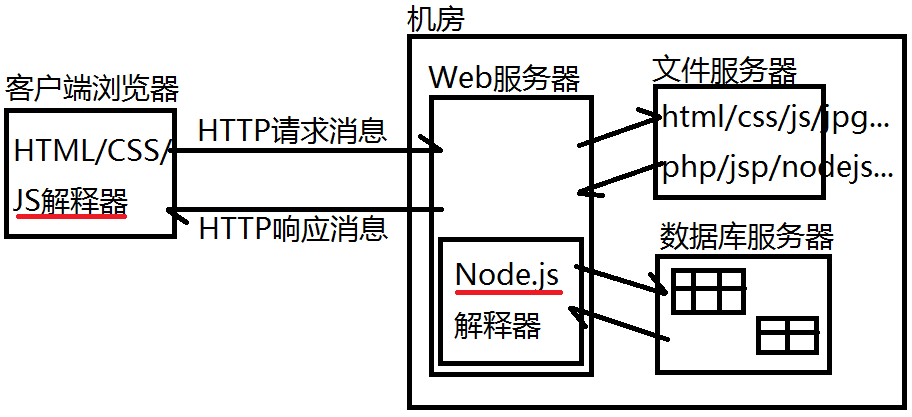
JSP/PHP/NodeJS...

JSP =HTML+JAVA 功能强大可靠，适合大型企业级项目

PHP =HTML+PHP 简单易用，适合互联网项目

ASP =HTML+C# 易用，适合windows平台

NodeJS =HTML+"JS" 性能好，适合服务器IO密集(微博)型项目，不适合CPU密集(天气预报)型项目.



3:Nodejs概述

NodeJS不是JS，就一个**软件开发平台**(安装服务器端),它的竞争对手，PHP/JSP/ASP ! 历史上第一次有一种语言可以通吃前后端.

官网:www.nodejs.org

版本: 0.12(2016年初) 4.x(2016年中) 6.x(2016年底)

LTS: Long Term Support 长期稳定支持版本

Current:最新版本

4:nodejs两种运行模式

(1)交互模式--一般用于临时测试

REPL:输入一行代码执行一行

注意:交互模式自带输出功能，不必定 console.log(age)

node 回车 进入交互模式

.exit 退出交互模式

(2)脚本模式--正式项目中使用方式

把要执行所有语句编写在一个文本文件中(后缀名任意,没有都行)

一次性提交node解释器执行.

node 完整路径名/x.js 回车

提示:只要安装node.js 重启 webstorm, ws可以自动发现nodejs解析器程序。记得新建项目一开视频修改文件编码UTF-8.

练习:创建3.js文件，打印九九乘法表,以脚本模式运行.

5:面试题-如何掌握一门新语言

如果自学一门新语言--NodeJS

(1)了解背景 --百度百科

(2)搭建开环境，编写HelloWorld

(3)数据类型--重点

|  |
| --- |
| 前端JS的数据类型  1:基本/原生/值类型/原始..  number/string/boolean/null/undefined  2:引用/对象类型  ES对象:  String/Number/Boolean/Math/Date/RegExp/Object/Function.  BOM:window/document/screen/location/navigator/event  DOM:Node/Element/Att...  3:用户自定义对象 |
| 后端NodeJS数据类型  1: 基本/原生/值类型/原始..  number/string/boolean/null/undefine  2:引用  ES 对象/Math/RegExp/Date/Object/Function/Boolean/...  3:NodeJS官方与**第三方**对象 https://www.npmjs.com/  4:用户自定义对象 |
|  |

(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

(6)逻辑结构

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

练习:创建4.js 声明一个保存5个学生成绩数组，使用三种for循环获取每个学生成绩.

选择结构 if..else switch...case

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

(7)通用小程序 九九乘法表 100以内的质数 数组排序..

(8)函数和对象

(9)常用的组件、第三方工具、框架

(10)实际小项目

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

6:模块(NodeJS中特有概念)--**重点&难点**

一个web项目功能可以分为很多个“模块”：如商品管理模块，支付模块，促销模块，商家管理....

NodeJS按照不同的功能，把函数，对象分别保存在不同文件，目录称为"Module".

***快速入门***:不同文件模块之间调用...

NodeJS中每个模块都是一个独立构造函数，解释器会为每个模块添加如下代码.

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

size(); //自己编写内容

});

exports:{} 用于声明向外部导出自己的成员

require:fn 用于导入其它的模块，**创建指定模块对象 new**

module: 代表当前模块

\_\_filename 当前文件绝对路径;

\_\_dirname 当前文件所属目录绝对路径;

|  |
| --- |
| NodeJS的模块中exports和module.exports对象区别是什么?  二者都可以用于向外界导出自己的内部成员  但是  module指代当前模块对象，直正导出的是moulde.exports  NodeJS底层有代码 exports = module.exports  所以:若只是给exports对象添加新的成员, 则等价于给module.exports添加新成员，*但是若修改了exports的指向，则不公产生实质作用*. |

练习: 创建14\_app.js

每个模块都可以使用自己的require()函数引入加一个模块--底层本质就是创建指定模块的一个对象实例.

require("./模块文件名");

每个模块可以使用exports对象向外导出/公开一些自己内部的成员供其它的模块使用.

exports.成员名=值

NodeJS 模块的分类

(1)nodejs官方提供的模块--安装在解释器内部

var 模块名 = require("模块名");

(2)第三方编写的模块

(3)用户自定义的模块

exports.x= y + require("./模块文件名");

NodeJS预定义--Global

该模块提供的可以直接使用,而无需 require("global");

exports 用于向外部导出当前模块的内部成员

module 用于指代当前模块

\_\_filename 返回当前模块的绝对路径

\_\_dirname 返回当前模块所属目录绝对路径

console 控制对象,注意该以象与chorme中console

setInterval(time,fn);

setTimeout(time,fn);

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

作业一:15\_app.js

作业二:仿写NodeJS提供的一个模块 URL

创建模块文件 myURL,向外出现一个方法 resolve(url),该方法接收的参数形如: http://www.jd.com:8080/index?uname=tom&age=10

返回对象:形如

{

protocol:'http',

sever:"www.jd.com",

port:80,

path:"/index?uname=tom&age=10"

}

再编写一个主模块:引入上述模块并调用