今天学习的内容

NODE.JS

NODE.JS 运行于服务器端开发平台,该平台的开发语言使用js

下载:淘宝镜像

静态网页

HTML/CSS/JS

动态网页

asp/php/jsp/node.js

node.js 数据类型

前端数据类型

原始

引用

ES

dom/bom

{}

node.js 数据类型

原始

引用

ES

{}

第三方对象

模块

用户自定义模块

文件模块

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

导出内部成员

exports.属性名=值

module.exports.属性名=值

module.exports = {};

导入其它模块

var m = require("./模块名');

目录模块

1:复习昨天重点

#面试题:用户在浏览器输入www.taobao.com直到看到页面

之间发生了什么

静态网页和动态网页

静态:网页内容任何人在任何时间访问都是不变的

HTML/CSS/JS/Flash/视频

动态:网页内容任何人在任何时间访问可能是不同的

JSP/PHP/ASP/Node.js

JSP = HTML+JAVA 功能强大，稳定适合大型项目(卡车)

PHP = HTML + PHP 简单易用，适合互联网(家用轿车)

ASP.NET = HTML+C# 易用，适合windows平台

(轿车/价格车卡贵)

Node.JS = HTML + "JS" 性能好，适合服务器IO密集型项目（输入输出频繁），

不适合CPU（计算复杂的情况不使用）密集型项目(超跑)

3.3:Node.js 概述

Node.JS 是一种运行于服务器端**开发平台**，该平台开发**语言js**.

Node.js不是JS,是一种服务器技术，它竞争对手是PHP/JSP/ASP

历史上一次有一种语言可以通吃前后台.

IO密集型项目(Node.JS) CPU密集型项目(java/c++)

滴滴打车,美团,天气预报

数据读取，数据保存

3.4:下载与配置(阿里镜像)

https://npm.taobao.org/mirrors/node

安装:下一步

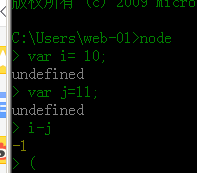
3.5:Node.JS的两种运行模式

(1)交互模式--一般用于测试

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

node 回车

注意:交互模式自带输出功能，不必写console.log();

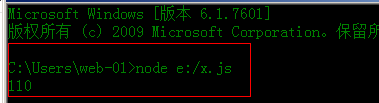


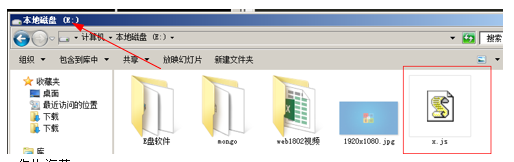
(2)脚本模式--正式项目开发（ws）

把要执行所有语句编写到一个文本文件中,(后缀任意,没有可)

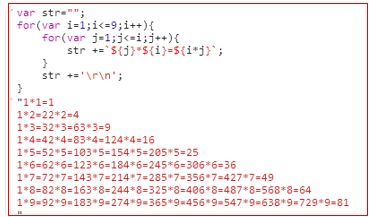
一次性提交nodejs解析执行

node 完整路径/x.js





练习：控制台输出：99乘法



双引号怎么去掉？

2:作业:海葵

海葵摆动效果(头部摆动)

贝赛尔曲线{起始点,控制点,终点}

正弦函数

3:今天的目标

3.1:node.js 基本概念

面试题--如何自学一门语言

(1)了解背景 百度百科 nodejs

阻塞IO(php/java/c#)、非阻塞IO(node.js)

阻塞:效率差，易于控制(死心眼-服务员)

非阻塞:效率高，控制有一定难度(机灵--服务员)

(2)搭建环境，编写HelloWorld

(3)数据类型

红色的是对方没有的:

|  |
| --- |
| **前端js中数据类型**  1:基本/原生/值类型  string/number/boolean/undefined/null  2:引用/对象类型  ES对象类型:String/Number/Boolean/Math/Date/  RegExp/Object/Function/Error/...  BOM对象类型:window/document/location/history/..  DOM对象类型:Node/Element/Attr....  3:用户自定义:{} |
| 后端nodejs中数据类型  1:基本/原生/值类型  string/number/boolean/undefined/null  2:引用/对象类型  ES对象类型:String/Number/Boolean/Math/Date/  RegExp/Object/Function/Error/...  3:用户自定义:{}  4:Node.js 十几万个第三方对象[https://www.npmjs.com/] |
|  |

(4)变量和常量

var age = 10;

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( .= +=...)

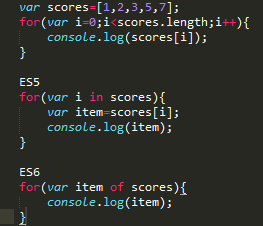
特殊运算符 . instanceof typeof

(6)逻辑结构

循环结构: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for循环依次打印出每个成绩值



选择结构:if ...else switch..case..

练习04: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),调用不同执行函数.(准备三个函数)



(7)通用小程序

练习05:打印100以内所有质数 11:48~11:53

(8)函数和对象

(9)常用组件，第三方工具，框架

(10)实际小项目

3.2:node.js 模块(重点)

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

3.3:Node.js的特有概念---模块

一个web项目功能可以分为很多不同“模块”,如商品管理,用户管理,订单,支付模块....

node.js按照功能不同，把函数，对象分别保存不同文件或目录下,

这些文件或者目录node.js就称为"Module"

Node.js中每个模块,自动添加构造函数，由解释器为每一个

.js文件**自动添两样以下代码**：

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

exports:{} 用于**声明向外部导出**的自己的成员

require:fn 用于**导入其它的模块**,并且创建指定模块对象

module:表示当前模块对象

\_\_filename:文件路径

\_\_dirname:文件所处目录路径（如下）

var age = 10; //自己代码内容

})

语法：导出exports.属性=值； module.exports.属性=值；

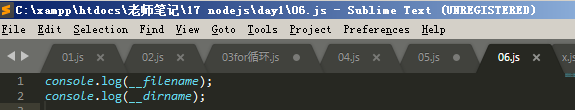
每个模块都可以使用自己require引入另一个模块--本质就是创建一个指定模块的对象实例.

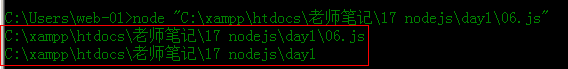
require("./模块名称");

每个模块可以使用exports对象向外导出/公开一些内部的成员,其它其模块使用.

exports.car = "半辆奔驰"

示例1：1.输出路径：06

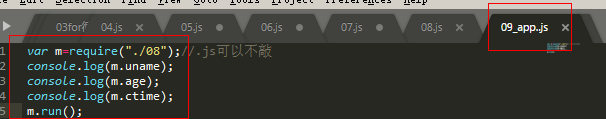


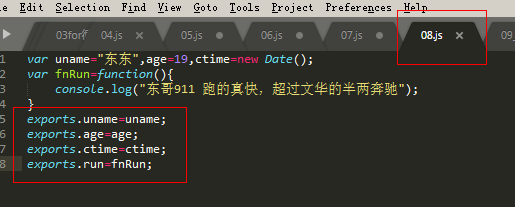


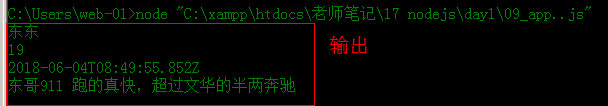
2.导出：别人就可以用了06

3.导入06模块对象，调用属性和方法

示例2:09







|  |
| --- |
| Node.js模块中exports和module.exports的区别是什么？（图解） |
| ........ |

练习:nodejs 导出导入引用

创建一个文件模块 10\_circle.js 其中定义常量PI

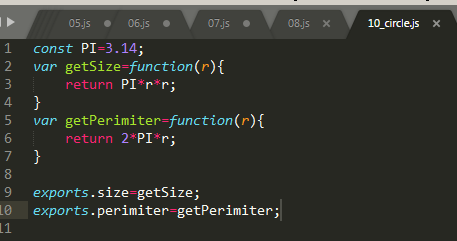
声明一个方法 getSize(r) 根据传入圆形的半径

返回其面积；声明一个方法getPerimiter(r),

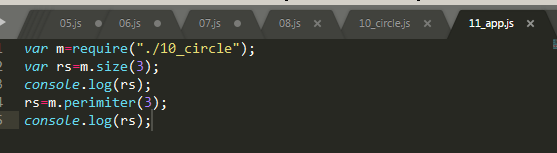
根据传入圆形的半径，返回其周长,再创建一个应用11\_app.js

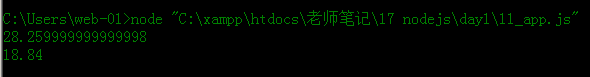
引入10\_circle模块,调用其公开的两成员方法

输出



引入

结果



练习:编写一个模块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)

返回指定数组中所有数值的平均值，

使module.exports导出上述两个成员,再创建 app.js引入

ArrayUtil.js模块，调用其公开的方法

3.4:Node.js中模块分类

(1)Node.JS官方提供的模块--安装解析器

require("模块名");

(2)第三方编写模块

(3)用户自定义模块(文件+目录)

exports.x = require("./模块名称");

3.5:Node.JS 官方模块--(预定义模块--Global)

该模块提供的可以直接使用，而无需require("global")

练习:

仿写 Node.JS 提供一个模块:URL

创建一个MyUrl向外导出一个resolve(url)该方法接收的参数形如

http://www.jd.com:8080/ad/index?uname=tom&no=3

返回对象,形如

{

protocol:"http",

server:"www.jd.com",

port:8080,

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

3.6:用户自定义模块

(1)文件模块

创建一个js文件,如m3.js,导出需要公开数据,其它模块可以

require("./m3");模块

(2)目录模块

方式一:

方式二:

方式三:创建目录,必须名 node\_modules ,其中再创建目录模块

假设名 m6,其中创建package.json描述文件,

其中声明main属性指定默认执行启动js文件,

如 6.js ,其中导出需要公开的数据

其它模块可以 require("m6");



17:30~17:40

练习:使用方式3创建二个目录模块,

circle/rectangle都对外公开两个方法size() perimeter()

返回指定图形面积和周长最后在最外层的主模块中引入上述两个模块circle.size(r); circle.perimeter(r);

r.size(r) r.perimiter(r)