# 第一阶段面试题

## 一、MySQL数据库

### 1. MySQL中varchar与char的区别以及varchar(50)中的50代表的涵义？

**答案：**Varchar是变长字符串，最多存储的大小是65535字节，查询速度相对较慢；Char 是定常字符串，最多存储的大小是255字节，查询速度相对较快；

varchar(50)表示：Mysql 4.0 表示存放50个字节，5.0存放50个字符。

### 2. 简述浏览器打开www.codeboy.com显示页面，中间都经过哪些过程？

**答案：**

向DNS服务器获取域名对应的IP地址并返回浏览器，通过IP地址向web服务器发请求，web服务器从文件服务器获取网页中所需要的文件，从数据库服务器获取网页中所需要的数据，最后讲完整的网页相应给浏览器。

### 3. 如何解决MySQL中存储中文乱码问题？

**答案：**

脚本文件采用UTF8编码

客户端连接数据库使用UTF8编码

服务器端创建数据库使用UTF8编码

### 4.Float和Double的区别是什么？

**答案：**

FLOAT(单精度浮点数)类型 数据可以存储至多8位十进制数，并在内存中占4字节。

DOUBLE(双精度浮点数)类型 数据可以存储至多18位十进制数，并在内存中占8字节。

### 5. 把数据id等于1的名字oldboy更改为oldgirl？

**答案：**

update test set name='oldgirl' where id=1;

### 6.如何登录mysql数据？

**答案：**

mysql -uroot

## 二、JS基础

### 1. continue和 break有什么区别？

**答案：**

break和continue都是用来控制循环结构的：

break终止循环，跳出循环体执行循环后面的语句。

continue跳过本次循环，执行下一次循环。

### 2. i++和++i的区别？

**答案：**

i++ ：先用i值后加1,

++i ：先加1后用i值， n= 6 7 8 9

计算：var n=5; 求 var num=n++ + ++n + n++ + ++n +n; //37

5 7 7 9 9

### 3. JavaScript都有哪些数据类型？

**答案：**

原始类型：数值型/字符串型/布尔型/undefined/null

引用类型：对象

### 4. 自调用函数是什么？用于什么地方？

**答案：立即执行的函数。** (function(){ //函数体 })();

用于创建一个局部作用域。

### 5. slice和splice有什么区别？

**答案：**

slice截取子数组 ，从指定的数组中，截取几个连续的元素组成一个新数组

splice允许从指定数组中，删除一部分元素，同时再添加另一部分元素

### 6. typeof返回的类型有哪些？

**答案：**

number 、string 、boolean 、undefined 、object 、function

ex:

var arr=[1,2,3,4,5];

console.log(typeof(arr)); //object

function myDemo(){

return function test(){};

}

console.log(typeof(myDemo())); //function

### 7. 取 1~11之间的随机数 (即包括1不包括11)？

**答案：**

Math.floor(Math.random()\*10+1)

parseInt (Math.random()\*10+1)

### 8. 什么是变量声明提前？

**答案：**

使用var关键字声明的变量，会将声明提升到所在作用域的最前边。

### 9. push、pop、shift和unshift 区别？

**答案：**

这两组同为对数组的操作，并且会改变数组的本身的长度及内容。

不同的是 push()、pop() 是从数组的尾部进行增减，unshift()、shift() 是从数组的头部进行增减。

### 10. 例举4种强制类型转换和3种隐式类型转换？

**答案：**

强制转换：parseInt/parseFloat/Number/toString

隐式转换：

字符串+数值 数值转字符串

数值+布尔型 布尔型中true转为1，false转为0

字符串+布尔型 布尔型转为字符串。

布尔型+布尔型 转换为数字

### 11. 函数声明与函数表达式的区别？

**答案：**

相同点：两者都可以创建函数。

不同点：函数声明可以存在函数提升（前），函数表达式不存在函数提升（前）。

//函数声明

function myFunction(){

function innerFunction() {}

}

//以下为函数表达式

var myFunc = function(){};

myFunc(function(){

return function(){};

} );

### 12. 请指出JavaScript宿主对象和原生对象的区别？

**答案：**

宿主对象：指JavaScript解释器提供的对象，由解释器厂家自定义并提供实现，不同的解释器提供的扩展对象存在较大的差异（DOM和BOM对象）。

原生对象：JavaScript语言本身预定义的对象，在ECMAScript标准中定义，由所有的解释器厂家来提供具体实现（Array,Date,Math,Number,String,Boolean等）。

### 13. js中有哪些内置函数？

**答案：**Object, Array, Boolean, Number, String, Function, Date, Math, RegExp, Error, Global

### 14.列举出获取日期相关函数

**答案：**

console.log(Date.now()); // 返回当前日期和时间'1970/01/01 00:00:00'之间的毫秒值

**var dt = new Date();**  // 获取当前时间---年 月 日 时 分 秒

console.log(dt.getTime()); //返回当前日期和时间'1970/01/01 00:00:00'之间的毫秒值

console.log(dt.getFullYear()); // 年份

console.log(dt.getMonth()+1); // 月份（0-11）

console.log(dt.getDate()); // 日期（0/1-31）

console.log(dt.getDay()); //星期（0-6）

console.log(dt.getHours()); // 小时（0-23）

console.log(dt.getMinutes()); // 分钟（0-59）

console.log(dt.getMilliseconds); // 返回毫秒（0-999）

console.log(dt.getSeconds()); // 秒（0-59）

### 15.Math相关函数

**答案：**

Math.random() - 返回 0 ~ 1 之间的随机数

Math.abs(x) - 返回数的绝对值

Math.ceil(x) - 向上取整

Math.floor(x) - 向下取整

Math.round() 四舍五入

Math.max() 和 Max.min() 获取一组数据中的最大值和最小值

Math.PI  获取圆周率π 的值  
Math.pow() 获取一个值的多少次幂   
Math.sqrt() 对数值开方

Math.pow(10，2) = 100;

Math.sqrt(100) = 10;

### 16.null和undefined的区别？

**答案：**

undefined是访问一个未初始化的变量时返回的值，

null是访问一个尚不存在的对象时所返回的值。因此，可以把undefined看作是空的变量，而null看作是空的对象。

### 17.==和===有什么不同？

**答案：**

== 抽象相等，比较时，会先进行类型转换，然后再比较值；

===严格相等，判断两个值相等同时数据类型也得相同。

### 18.setTimeout和setInterval的区别是什么？

**答案：**

二者都是用来设置定时操作的。

setTimeout: 设置一个定时器，在定时器到期后执行一次函数或代码段

setInterval: 设置一个定时器，以固定的时间间隔重复调用一个函数或者代码段

### 19. 请说出以下代码执行结果

for (var i = 0; i < 3; i++) {

setTimeout(function() {

console.log(i);

}, 0);

console.log(i);

}

**答案：**0 1 2 3 3 3，**执行过程如下**

var i = 0;

console.log(i); i++;

console.log(i); i++;

console.log(i);i++;

setTimeout(function() {

console.log(i);

}, 0);

setTimeout(function() {

console.log(i);

}, 0);

setTimeout(function() {

console.log(i);

}, 0); //输出 0 1 2 3 3 3

### 20. 请说出（true+flase）>2+true的执行结果

**答案：false**

### 21. 当前代码块输出结果是什么？

var z=10;

function foo(){console.log(z);}

(function(funArg){var z=20;funArg();})(foo);

**答案**： 10，自调用函数。

### 22. setTimeout(function(){},10)表示什么意思？

**答案**：每隔10毫秒调用一次函数。

### 23. 程序中捕获异常的方法？

**答案**：try{

}catch(e){

}finally{

}

### 24. 以下代码执行结果？

**var uname = 'jack'**

**function change() {**

**alert(uname) // ?** undefined

**var uname = 'lily'**

**alert(uname) //? 'lily'**

**}**

**change()-+**

**答案**：undefined lily

### 25. 如何使用npm下载express模块？

**答案**：npm install express

### 26. split和join的区别？

**答案**：split()将字符串按照指定的字符分割成一个数组，并返回

join()将数组用指定的字符连接成一个字符串，并返回

### 27. 看下列代码会有什么样的输出？

**var foo = "11"+2-"1";**

**console.log(foo);**

**console.log(typeof foo);**

**答案**：111 number

### 28. foo = foo||bar ，这行代码是什么意思？为什么要这样写？

**答案**：这种写法称之为短路表达式

相当于：

var foo;

if(foo){

foo=foo;

}else{

foo=bar;

}

常用于函数参数的空判断

### 29. 用js实现随机选取10–100之间的10个数字，存入一个数组，并排序

**答案**：function getRandom(istart, iend){

var iChoice = iend - istart +1;

return Math.floor(Math.random() \* iChoice+ istart);

}

var iArray = [];

for(var i=0; i<10; i++){

var result= getRandom(10,100);

iArray.push(result);

}

iArray.sort();

### 30. 如何获取javascript三个数中的最大值和最小值？

**答案**：Math.max(a,b,c);//最大值

Math.min(a,b,c)//最小值

### 31. 实现冒泡排序？

var array = [5, 4, 3, 2, 1];

var temp = 0;

for (var i = 0; i <array.length; i++){

for (var j = 0; j <array.length - i; j++){

if (array[j] > array[j + 1]){

temp = array[j + 1];

array[j + 1] = array[j];

array[j] = temp;

}

}

## 三、NODEJS

### 1. 同步和异步有何区别？

**答案：**

同步：指发送一个请求,需要等待返回,然后才能够发送下一个请求，有等待过程（在一个任务进行中时不能开启其他的任务）。

异步：指发送一个请求,不需要等待返回,随时可以再发送下一个请求，即不需要等待（在一个任务进行中时可以开启其他的任务）。

### 2. NodeJS中有哪些类型模块，文件操作用哪一个模块？

**答案：**

模块类型： 核心模块、自定义模块、第三方模块。

文件操作：fs模块

### 3. 对NodeJS的优点和缺点提出自己的看法？

**答案：**CPU密集型任务的特点是进行大量的计算，消耗CPU资源，比如计算圆周率(上千位)、对视频进行编码等, 全靠CPU的运算能力 (一般用C语言，java)

IO（Input / Output）密集型任务，这类任务的特点是CPU消耗很少，大部分时间都在等待IO操作。常见的大部分任务都是IO密集型任务，比如Web应用（一般用脚本语言：python/Nodejs）。

Nodejs设计思想中以事件驱动、异步、非堵塞I/O密集型为核心，他提供的大多数api都是基于事件的、异步的风格。所以非常适合处理高并发请求。此外，与Node服务器交互的客户端代码是由js语言编写的，因此客户端和服务器端都用同一种语言编写，减少了成本。

### 4. 使用NodeJS完成登录功能(编写HTML页面和路由接口中的代码)？

**答案：**

**HTML页面：**

<form method="post" action="/user/login">

用户：<input type="text" name="uname"> <br>

密码：<input type="text" name="upwd"> <br>

<input type="submit" >

</form>

**user.js路由**

const express=require('express');

const pool=require('../pool.js');

var router=express.Router();

router.post('/login',(req,res)=>{

var obj=req.body;

var $uname=obj.uname;

if(!$uname){

res.send({code:401,msg:'uname required'});

return;

}

var $upwd=obj.upwd;

if(!$upwd){

res.send({code:402,msg:'upwd required'});

return;

}

var sql='SELECT \* FROM xz\_user WHERE uname=? AND upwd=?';

pool.query(sql,[$uname,$upwd],(err,result)=>{

if(err) throw err;

if(result.length>0){

res.send({code:200,msg:'login success'});

}else{

res.send({code:301,msg:'login error'}); }

});

});

//路由器导出

module.exports=router;

### 5. 什么是Nodejs？

Nodejs是一个JavaScript的运行环境，是一个服务器端的“JavaScript解释器”，用于方便高效地搭建一些响应速度快、易于扩展的网络应用。它采用事件驱动、异步编程，为网络服务而设计

### 6. nodejs适用于哪些地方？

高并发、聊天、实时消息推送

### 7. npm是什么？

### npm是nodejs包管理和分发的工具，用于管理node包。如安装、卸载、发布、查看等.

### 8. npm的好处是什么

通过npm，可以安装和管理项目的依赖，且可以指明依赖项的具体版本号。

### 9. Node.js中导入模块和导入js文件写法上有什么区别？

nodejs引入模块，直接使用名字导入即可

const express = require("express");

导入js文件，需要使用文件的路径，如：

const student = require("./pool.js");

### 10. console有哪些常用方法？

console.log/info/error/warn/time/timeEnd

### 11. express response有哪些常用方法？

### res.download() 弹出文件下载

### res.end() 结束response

### res.json() 返回json

### res.jsonp() 返回jsonp

### res.redirect() 重定向请求

### res.render() 渲染模板

### res.send() 返回多种形式数据

### res.sendFile 返回文件

### res.sendStatus() 返回状

### 12. node中的Buffer如何应用?？

Buffer是用来处理二进制数据的，比如图片，mp3,数据库文件等.Buffer支持各种编码解码，二进制字符串互转

### 13. 实现一个简单的http服务器？

var http = require('http'); //加载http模块

var server=http.createServer(function(req, res) {

res.writeHead(200, {'Content-Type': 'text/html'});

res.write('hello world');

res.end(); //结束输出流

});

server.listen(3000);