一、单选题

1.D

2.B

3.B

JavaScript语言是不是JAVA语言的一种扩展语言，而是基于JAVA语言设计的。

JavaScript中字符在声明的时候使用单引号和双引号

D.JavaScript是一门面向过程的语言

4.C

逻辑与 (&&) expr1 && expr2 如果expr1 能转换成false则返回expr1,否则返回expr2. 因此, 在Boolean环境中使用时, 两个操作结果都为true时返回true,否则返回false.

逻辑或 (||) expr1 || expr2 如果expr1能转换成true则返回expr1,否则返回expr2. 因此,在boolean环境(在if的条件判断中)中使用时, 二者操作结果中只要有一个为true,返回true;二者操作结果都为false时返回false.

逻辑非(!) !expr 如果单个表达式能转换为true的话返回false,否则返回true.

能够转换为false的表达式有:null,0,""和undefined.

5.D

6.D

7.B

window.onload是在页面所有的元素都加载完成后才触发$(function(){})是在页面的dom结构加载完毕后就触发 dom里的内容不一定都已经加载完成比如说一个 页面有好多图片 而加载这些图片需要一定的时间window.onload必须得等到全部的图片都加载完成后才能触发而$(function(){})只要在dom加载完毕之后就会执行 图片不一定已经加载完

8.C

==用于一般比较，===用于严格比较，==在比较的时候可以转换数据类型，===严格比较，只要类型不匹配就返回flase。

拓展：

"1" == true

类型不同，"=="将先做类型转换，把true转换为1，即为 "1" == 1；

此时，类型仍不同，继续进行类型转换，把"1"转换为1，即为 1 == 1；

此时，"==" 左右两边的类型都为数值型

如果比较："1" === true 左侧为字符型，右侧为bool布尔型，左右两侧类型不同，结果为false；

总之"==" 只要求值相等; "===" 要求值和类型都相等

9.B

var a=10，b=20，c=4; ++b+c+a++

++b先自加再运算，a++先运算再自加

++b + c + a++

21 + 4 + 10 = 35

拓展：

var a = 1; var b = a++ + ++a; // 4

1 + 3

10.D

匹配前面一个表达式1次或者多次。等价于 {1,}。

二、填空题

1.任意类型

2.Undefined Null Boolean Number String

3.undefined

4.引用，指向同一个对象，值相等

5.对应基本类型的一个包装对象

三、程序阅读题

1、输出：4 4 4

原因：

for(var i=1;i<=3;i++){

setTimeout(function(){

console.log(i);

},0);

};

执行顺序： for循环---> setTimeout执行，（但是其里面的函数不执行，虽然设定的延迟时间是 0 ，里面的函数只是setTimeout的一个任务，函数执行时，要等到上面的代码执行完毕后才能执行 ）；

--->最后 一次for循环 此时i=4,不满足i<=3 条件，执行setTimeout执行，最后执行 function(){console.log(i)}。

拓展：

for (var i = 1; i <= 3; i++) {

setTimeout(function (a) {

console.log(a);

}(i),0);

};

输出： 1 2 3

setTimeout里面的函数是一个立即执行的函数，当setTimeout执行之后，其里面的函数也会立即执行，括号中的i作为实参传到形参a中，每次for循环都会输出。

2.

var foo =1;

function fn(){

foo=10;

return;

function foo(){

}

}

fn();

console.log(foo);

输出结果： 1

原因：

首先先进行预解析

var foo =1;

function fn(){

foo=10;

function foo(){

}

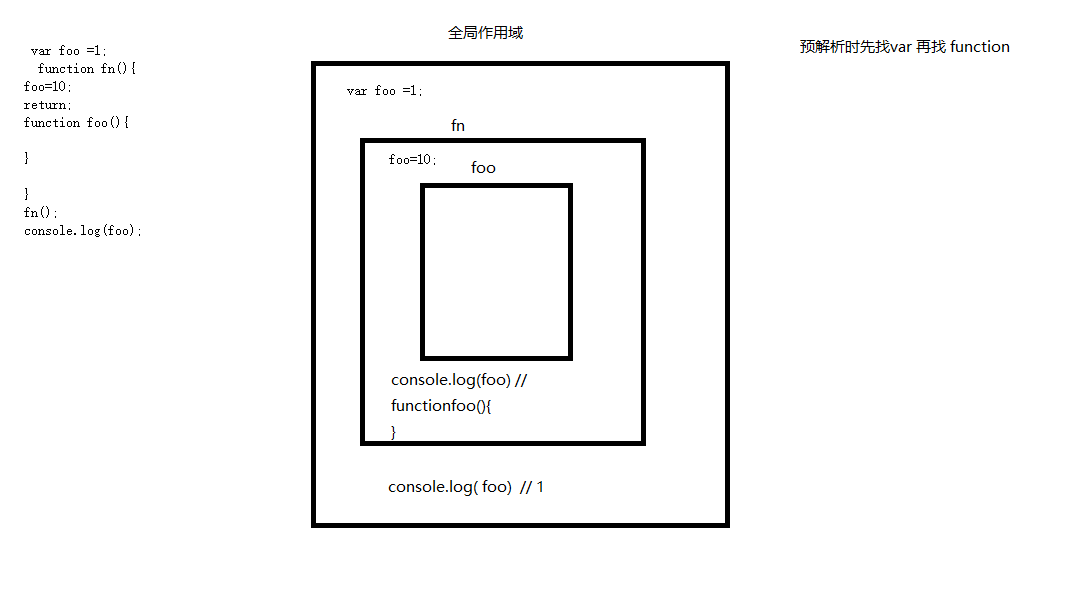
return;

}

fn();

console.log(foo);

由于是在全局作用域中查找foo所以是 1



3.

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

console.log(foo);

console.log(typeof foo);

"11"+2 --> "112"

+ 号在字符串和数值之间或者字符串和字符串之间有“连接”作用 ， 数值之间起到 “相加” 的作用。

"112" -"1" --> 111 -->number

- 号起到 隐式转换作用

4.

function aa()

{

return function(){

var aa=10;

var bb=20;

};

}

console.log(aa);

console.log(bb);

console.log(aa); // function aa()

{

return function(){

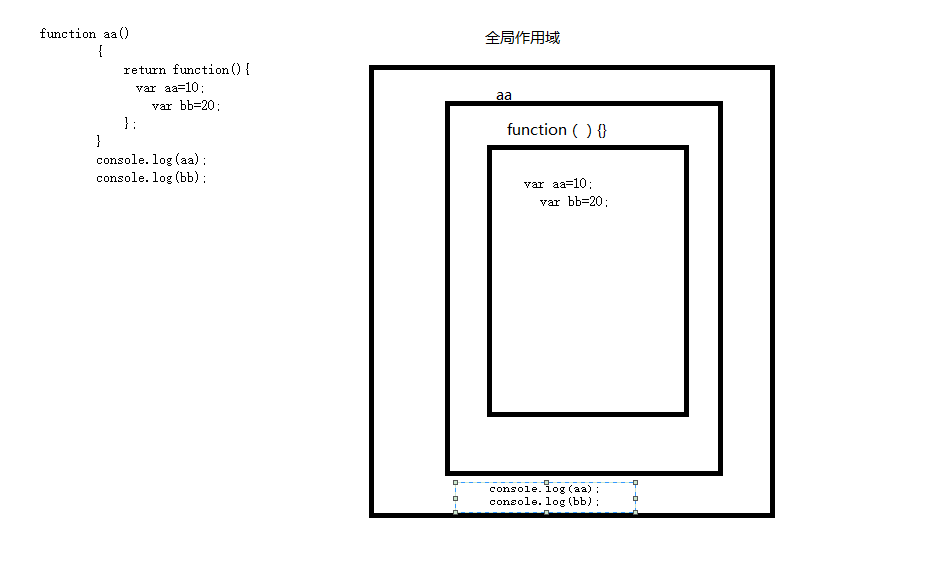
var aa=10;

var bb=20;

};

}

console.log(bb);// 报错（在全局作用域种没有找到）



5.

输出： false

function Fn(){return Fn};

fn =new Fn()

console.log(fn instanceof Fn);

instanceof 用于判断一个变量是否是某个对象的实例（面向对象那会讲）

new Fn() 相当于创建了一个调用后得Fn

Fn函数中是有返回值的所以new Fn（）就相当于 return Fn === function Fn(){return Fn}

如果是这样

function Fn(){ };

fn =new Fn()

console.log(fn instanceof Fn);

输出： ture

四、简答题

1.

//谷歌和火狐浏览器中阻止事件冒泡的写法

e.stopPropagation()

//IE8中阻止事件冒泡的写法

window.event.cancelBubble=true;

//阻止默认事件(W3C)

e.preventDefault();

//IE中阻止默认事件

window.event.returnValue = false;

2.

JavaScript是一种客户端的脚本语言，不需要编译，直接运行时边解析边执行的语言，主要用来向HTML（标准通用标记语言下的一个应用）页面添加交互行为，有了JavaScript，可使网页变得生动，它可以通过嵌入或调入在标准的HTML语言中实现，单独的js文件有利于结构和行为的分离。

特点

简单 JavaScript语言中采用的是弱类型的变量类型,对使用的数据类型未做出严格的要求,是基于Java基本语句和控制的脚本语言,其设计简单紧凑

动态性　 JavaScript是动态的，它可以直接对用户或客户输入做出响应，无须经过Web服务程序。

跨平台性 　JavaScript是依赖于浏览器本身，与操作环境无关，只要能运行浏览器的计算机，并支持JavaScript的浏览器就可以正确执行。

3.

事件捕获阶段

事件目标阶段

事件冒泡阶段

4.

如果 foo为真，则赋值给foo ，否则 bar赋值给 foo 。

精简了代码的，优化了代码。

5.

var a =null;

console.log(typeof a);

结果是 object

原因是null值表示一个空对象指针，所以使用typeof操作符检测null值时会返回object

五、程序题

1.

var array1 = ['a','b','c','e'];

var array2 = ['d','e','f','f'];

var array = array1.concat(array2);

function unique(arr) {

var result = [];

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

var item = arr[i]

// 如果新的数组中找不到该元素，就放入新的数组中。

if (result.indexOf(item) === -1) {

result.push(item);

}

}

return result;

}

console.log(unique(array));

2.

function getRandom(){

// 定义一个空数组,把符合的随机数放在里面

var num = [];

for(var i=0;i<5;i++){

// 生成一个随机数

var rdom = Math.ceil(Math.random()\*10);

// 假设，生成的随机数都不相等

var flag = false;

for(var index in num ){

// 判断如果生成的随机数相等

if(num[index]==rdom){

flag = true;

break;

}

}

//生成的随机数相等,就让i--，重新进行循环

if(flag){

i--;

}else{ // 生成的随机数不相等，放到数组中去

num[num.length] = rdom;

}

}

return num;

}

console.log(getRandom());

3.

// 1

// 1 3

// 1 3 5

// 1 3 5 7

// 1 3 5 7 9 2n-1

// getSum(num-1) + 2n-1

function getSum(num){

if(num==1){

return 1;

}

return 2\*num-1+getSum(num-1);

}

console.log(getSum(4));

4.

var arr = [65, 97, 76, 13, 27, 49, 58];

//比较了多少趟 arr.length - 1

//for (var i = 1; i <= arr.length - 1; i++) {

for (var i = 0; i < arr.length - 1; i++) {

var sorted = true;//假设已经排好了

//每一躺中 两两比较的次数

for (var j = 0; j < arr.length - 1 - i; j++) {

//两两比较

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

var temp = arr[j + 1];

arr[j + 1] = arr[j];

arr[j] = temp;

sorted = false;

}

}

//如果没有任何人告诉我说没排好 那就说明真的排好了

if (sorted) {

break;

}

}

console.log(arr);

5.

function getColor(){

var colorElements = "0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f";

var colorArray = colorElements.split(',');

var color = "#";

for(var i=0;i<6;i++){

color+=colorArray[Math.floor(Math.random()\*16)]

}

return color;

}

var result = getColor();

console.log(result);