**JS一期一阶段第二次测试（5-7章）**

您的姓名： [填空题]

填空题数据请通过下载详细数据获取

1.下面的等式成立的是() [单选题]

|  |  |  |
| --- | --- | --- |
| 选项 | 小计 | 比例 |
| A， parseInt(12.5) == parseFloat(12.5) |  |  |
| B， Number('') == parseFloat('') |  |  |
| C， isNaN('abc') == NaN |  |  |
| D， typeof NaN === 'number' |  |  |

正确率：63.64%

解析：

1. 取整后 12 取浮点数后 12.5 12==12.5 （false）
2. Number后 是 0 取浮点数后 是NaN 0==NaN （false）
3. isNaN(‘abc’) 返回一个布尔值 true 但NaN和任何值都不相等（包括它自己） isNaN(‘abc’) ==NaN （false）
4. NaN属于number类型 所以 typeof NaN 结果是字符串”number” typeof NaN == ‘number’ （true）

2.以下代码运行后，结果为()

var i = 0;

alert( ++i == i++ ); [单选题]

|  |  |  |
| --- | --- | --- |
| 选项 | 小计 | 比例 |
| A： true |  |  |
| B：false |  |  |
| C：undefined |  |  |
| D：NaN |  |  |

正确率：52.73%

解析： ++i 和 i++的区别

++i 先加1 ，i会立即变成+1后的值

I++ 后加1，i会在i++之后 才得到+1后的值 当时的i还是未+1的值

I 的初始值为 0

大家可以试着打印 i++ 和 ++i 就明白了

++i 之后 i已经变成了 1 相当于等号左侧 是1 ，等号右面的i++ ，没有立即改变i的值，所以还是1

3.JavaScript 定义 var a="40",var b=7,则执行 a%b 会得到() [单选题]

|  |  |  |
| --- | --- | --- |
| 选项 | 小计 | 比例 |
| A.5 |  |  |
| B."5" |  |  |
| C.undefined |  |  |
| D.null |  |  |

正确率：90.91%

解析： 取模运算 可以进行隐式转换

4.看下列代码输出什么？

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

console.log(foo);

[单选题]

|  |  |  |
| --- | --- | --- |
| 选项 | 小计 | 比例 |
| A: 1121 | 13 |  |
| B: 131 | 0 |  |
| C: 111 | 90 |  |
| D: 112 | 7 |  |

正确率：81.82%

解析： “11”+2-”1” ; 从左往右依次执行

字符串“11”和数字2 用+ 隐式转换成字符串拼接，形成字符串“112”

“112”-“1” 用减号连接，减号只有运算功能，所以会将字符串隐式转换成数字，所以 112-1 =111

5.如下代码输出的结果是什么：

console.log(1+ "2"+"2");

console.log(1+ +"2"+"2");

console.log("A"- "B"+"2");

console.log("A"- "B"+2); [单选题]

|  |  |  |
| --- | --- | --- |
| 选项 | 小计 | 比例 |
| A.122 122 NaN NaN |  |  |
| B.122 32 NaN NaN2 |  |  |
| C.122 32 NaN2 NaN |  |  |
| D.122 32 NaN2 NaN2 |  |  |

正确率：61.47%

解析： 第一个 console.log(1+ "2"+"2"); 隐式转换字符串拼接 “122”

第二个 console.log(1+ +"2"+"2"); +”2” 在字符串前面如果有-+号,会被识别为正负号，所以会转成number 1+2+“2”

第三个 console.log("A"- "B"+"2"); 字符串相减非数字会产生NaN，所以是NaN+”2”

第四个 console.log("A"- "B"+2); 字符串相减非数字会产生NaN，NaN+2 还是NaN

6.下面代码的运行结果是： 第一次弹()第二次弹()

function fn1() { alert(1); }

alert( fn1() ) [多选题]

|  |  |  |
| --- | --- | --- |
| 选项 | 小计 | 比例 |
| A：1 |  |  |
| B：alert(1); |  |  |
| C：function fn1() { alert(1); } |  |  |
| D：undefined |  |  |

解析：第一次 先执行 fn1() ,所以会弹 1

第二次会执行 alert(fn1()) ，因为 fn1函数没有定义返回值(return) 所以fn1() 在执行之后 是undefined

相当于 alert(undefined) ,所以会弹出 undefined

7.以下代码运行后，打印出b的值为()

(function(){

var a = 5，b=10;

})();

console.log(b); [单选题]

|  |  |  |
| --- | --- | --- |
| 选项 | 小计 | 比例 |
| A: 5 |  |  |
| B: 程序报错 |  |  |
| C: undefined |  |  |
| D. null |  |  |

正确率：57.8%

解析：var a=5,b=10; 相当于 var a=5; var b=10; 所以b是布局变量，在外部访问不到；

在js当中如果使用未定义的变量或者函数，就会报错。Undefined 只在定义过但未赋值的情况下出现；例 var a;

Console.log(a)

8.请给出这段代码的运行结果( )

var bb = 1;

function aa(bb) {

bb = 2;

alert(bb);

};

aa(bb);

alert(bb); [单选题]

|  |  |  |
| --- | --- | --- |
| 选项 | 小计 | 比例 |
| A.1 1 |  |  |
| B.1 2 |  |  |
| C.2 1 |  |  |
| D.2 2 |  |  |

正确率：73.39%

解析：首先会执行aa(bb) , bb当前是传入的是一个全局变量值1，相当于 aa(1);

再函数aa中 bb=1; 此时，bb是函数aa的参数，可理解成局部变量，那在函数aa范围内的bb修改的就只是局部的bb，

与全局变量bb无关。 所以在函数内的bb=2；只是改了aa内的局部变量bb的值。alert(bb); 当前作用域下的bb是2 所以会弹出2；

在全局的alert(bb) 调用的bb是全局变量bb，所以会弹出1

9.阅读下列程序，写出 x，y，z 最后的值。

var x = 1, y = z = 0;

function add(n)

{

return n = n + 1;

}

y = add(x);

function add(n)

{

return n = n + 3;

}

z = add(x); [单选题]

|  |  |  |
| --- | --- | --- |
| 选项 | 小计 | 比例 |
| A. 1 2 4 |  |  |
| B. 1 4 4 |  |  |
| C. 4 2 4 |  |  |
| D. 1 1 1 |  |  |

正确率：30.56%

解析：1. 函数声明会预解析，所以 俩个同名函数add后面会覆盖前面的，并且提升上去;所以js中只保留了下面的add函数（即n+3的函数）

1. 全局变量x值是1 相当于 y=add(1);
2. 执行add函数后 它会返回一个新的值 即 n=n+3;这里的n传进来的是1，所以n新的值=1+3；那n就等于4，并且把4赋值给了y；
3. 执行z = add(x); x还是1 ，和上面一样 传入1，并返回 1+3的结果4；然后赋值给z
4. 所以最后 x值没有变过，是1 ； y和z都是4；因为他俩调用的是同一个函数

10.以下代码的运行结果是

(function() {

var a = b = 5;

})();

console.log(b);

console.log(a); [单选题]

|  |  |  |
| --- | --- | --- |
| 选项 | 小计 | 比例 |
| A. 5 5 |  |  |
| B. undefined，undefined |  |  |
| C. 5，undefined |  |  |
| D. 5，Uncaught ReferenceError: a is not defined |  |  |

正确率：59.63%

解析：这里注意，var a=b=5; 和 var a=5,b=5; 意思不一样；

var a=b=5; 相当于 var a=5 ; b=5; （这里的b前面没有带var）

var a=5,b=5; 相当于 var a=5; var b=5; （这里的b是带var的）

在题目中，var a=b=5; 这种写法下，b就不是局部变量，那它就可以在全局被调用；

a是加了var 的， 所以是局部变量，外部无法访问。