# web前端 第二周 周考题目

选择题

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| C | D | C | C | D | A | A | A | C | C | b/D | C | D | B | B |

判断题 🗸 🗴

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| T | F | F | F | T | T | T | T | F | F | F | F | F | T | F |

一: 选择题

1. 以下关于函数的描述,错误的是: C
2. 函数体中使用到参数列表的参数, 必须在定义时声明
3. 函数可以复用, 可以给同一个函数传递不同的参数
4. 自调用函数的函数体必须是匿名函数
5. 自调用函数可以不需要函数名
6. 关于DOM节点,以下描述错误的是 D
7. <html>节点没有父节点
8. <body>节点和<head>节点是兄弟节点
9. 一个节点可以是父节点,也可以是子节点
10. html中的文本不是节点
11. 以下哪个选项不是 HTML DOM 属性 D
12. innerHTML
13. parentNode
14. appendChild()
15. attributes
16. 以下哪个选项不是针对子节点的操作 C
17. removeChild()
18. appendChild() // 末尾添加节点
19. createElement()
20. insertBefore() // 开头
21. innerHTML 和innerText的区别是 D
22. 一个是节点的方法,一个是节点的属性
23. 一个操作节点内部文本,一个操作节点的外部文本
24. 没有区别
25. innerText不处理文本中的html标签字符串
26. 以下节点查找的方法中, 不会返回集合的是 A
27. getElementById
28. getElementsByClassName
29. getElementsByTagName
30. documen.images
31. ×以下描述正确的是 C
32. document.createElement方法会在页面上创建一个div
33. appendChild方法会在页面上创建一个节点
34. document.createAttribute方法会在页面上生成一个属性
35. setAttributeNode方法
36. 以下关于说法,正确的是 D
37. appendChild ()方法可以在任意指定位置添加节点
38. removeChild()不需要父节点也能操作
39. 节点的remove()方法任何时候都生效
40. document.getElementById("fang01").remove()可以删除节点
41. 以下关于代码规范的说法,正确的是 D
42. 代码规范是为了让解释器更加高效的运行
43. 过多的注释也会影响程序的执行效率
44. 代码规范不能替代语法, 所以代码规范的约束力不是强制的
45. 代码规范很严格,必须每行代码都遵循
46. 有语句var divE = document.createElement('div'); 以下代码执行不会生效的是 A
47. divE.ID = "000";
48. divE.style.display = "";
49. divE.class = "red";
50. divE.setAttribute( "innerText", "red")
51. 以下关于setInterval()函数的说法,正确的是 D
52. 这个函数不需要参数
53. 这个函数用来周期性的执行一个函数
54. 这个函数没有返回值 // 句柄
55. 这个函数接收两个参数, 分别为函数和毫秒值
56. 关于setTimeout()函数,说法正确的是: C
57. 这个函数没有返回值
58. 这个函数一旦开始, 可以执行多次
59. 这个函数延迟执行
60. 这个函数有两个参数,一旦执行,就不能停止
61. 以下哪个事件是鼠标事件 D
62. onchange
63. onkeypress
64. onkeypress
65. onclick
66. 以下关于事件的说法正确的是 B
67. 事件对象必须要声明才能时候
68. 事件对象在事件调用的时候隐式的传递给了事件函数
69. 事件对象中包含了事件源信息, 事件源就是鼠标,键盘等触发事件者
70. 事件的对象内容总是不变的,跟浏览器无关
71. 以下说法错误的是 B
72. document.body上的事件必须有事件源
73. 浏览器窗口window 上不能注册事件
74. 事件源可以是浏览器窗口
75. 事件源包含了其自身的坐标信息

二: 判断题

1. document.getElementById()通过id去查找节点, 参数必须不为空 T
2. onmousedown事件其实就是onclick事件,只是触发顺序不同 F
3. setInterval()函数执行的周期可以随时调整 F
4. document.getElementById("fang01").innerText = "我<br>是<br>小猪<br>佩奇";这条语句会 报错,因为innerText不能包含html标签符号 F
5. 注册一个事件函数的时候, 可以不调用事件源 T
6. 页面的onLoad事件总是会发生, 不管有没有绑定事件函数 T
7. document是 window的一个属性 T
8. 所谓的全局变量, 都是window对象的直属属性 T
9. window是一个庞大的对象, 是一个标准, 跟浏览器无关 F
10. 一个全局函数可以被可以通过window访问, 所以它内部声明的变量也可以 F
11. 每一个通过new创建的对象, 对于window而言都是可见的 F
12. document.body的高度等于页面的高度 F
13. 一旦声明了一个事件函数, 就不能销毁, 但是可以重写以覆盖旧的事件函数 F 绑定事件 = NULL
14. 解释器会校验语法错误, 但逻辑错误不属于解释器校验的范围 T
15. 页面上所有可见的元素都可以被选中, 都可以触发事件 F

三: 程序分析题

1. 以下代码,计时器每秒都会在控制台上打印currAngle, 请描述它的规律(前3次输出内容)

var timerPan = setInterval(function() {

var currAngle = 0 ;

currAngle += 5; // 每次转动的角度

if (currAngle > 360) {

currAngle = 0;

}

console.info(currAngle );

}, 1000);

5

5

5

2. 假设页面上有一系列class= " student "的div标签, 请指出以下代码的意图, 并指出这个方法是否可行.

var allguys = document.getElementsByClassName("student");

while (allguys.length > 0) {

allguys[0].remove();

}

意图:清除所有属性为student的div

可行

四:程序设计题

1. 已知页面上有两个按钮, 写一段代码, 当点击bt1的时候,禁用bt2 , 并且按钮上的文字改成"已经禁用"

提示 : 禁用按钮可以修改它的disabled属性 = true;

<input type="button" id="bt1" value="禁用你" />

<input type="button" id="bt2" value="不关我的事啊" />

var button1 = document.getElementById("bt1");

button1.onclick = function() {

document.getElementById("bt2").disabled = "true";

document.getElementById("bt2").value = "已经禁用";

}

1. 已知页面上有个按钮,

<input type="button" id="bt3" value="停止报时" />

<input type="button" id="bt4" value="继续报时" />

设定一个计时器, 每隔1秒在控制台输出一次当前时间的毫秒数,

当按钮 bt3被点击的时候, 停止控制台的输出.

当按钮bt4 被点击的时候, 继续控制台的输出

var timer = setInterval(function() {

var now = new Date().getTime();

console.info(now);

}, 1000)

document.getElementById("bt3").onclick = function() {

clearInterval(timer);

}

document.getElementById("bt4").onclick = function() {

timer = setInterval(function() {

var now = new Date().getTime();

console.info(now);

}, 1000)

}