1.说说以下代码的执行结果:

var result = 123 + true + "456";

console.log(result);

结果：124456

2.说出如下代码的执行结果

var num = 10;

var result = num++ + ++num + num

console.log(result);

结果：34

3.写出下列代码的执行结果

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

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

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

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

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

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

结果：

122

32

02

112

NaN2

NaN

4.写出下列代码的执行结果

console.log("0 || 1 = "+(0 || 1));

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

console.log("0 && 1 = "+(0 && 1));

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

结果：

1

1

0

2

5. 从键盘输入小明的期末成绩:

当成绩为100时，'奖励一辆BMW'

当成绩为[80-99]时，'奖励一台iphone15s'

当成绩为[60-80]时，'奖励一本参考书'

其他时，什么奖励也没有

var score = prompt("请输入小明的期末成绩");

if(score >= 0 && score <=100){

if(score == 100){

alert("奖励一辆BMW");

}else if(score >= 80 && score < 100){

alert("奖励一台iphone15s");

}else if(score < 80 && score > 60){

alert("奖励一本参考书");

}

}else{

alert("什么奖励也没有");

}

6.大家都知道，男大当婚，女大当嫁。那么女方家长要嫁女儿，当然要提出一定的条件：

高：180cm以上; 富:1000万以上; 帅:500以上;

如果这三个条件同时满足，则:'我一定要嫁给他'

如果三个条件有为真的情况，则:'嫁吧，比上不足，比下有余。'

如果三个条件都不满足，则:'不嫁！'

var height = prompt("请输入男方的身高");

var money = prompt("请输入男方财产");

var face = prompt("请输入男方帅气程度");

if(height >= 180 && face >= 1000 && money >= 500){

alert("我一定要嫁给他");

}else if(height >= 180 || face >= 1000 || money >= 500){

alert("嫁吧，比上不足，比下有余。");

}else{

alert("不嫁！");

}

1. 编写程序，由键盘输入三个数分别存入变量num1、num2、num3，对他们进行排序，并且从小到大输出。

var num1 = +prompt("请输入num1");

var num2 = +prompt("请输入num2");

var num3 = +prompt("请输入num3");

if(num1 <= num2 && num1 <= num3){

if(num2 < num3){

alert(num1 + ','+ num2 + ',' + num3);

}else{

alert(num1 + ','+ num3 + ',' + num2);

if(num2 <= num1 && num2 <= num3){

if(num1 < num3){

alert(num2 + ','+ num1 + ',' + num3);

}else{

alert(num2 + ','+ num3 + ',' + num2);

}

if(num3 <= num2 && num3 <= num1){

if(num1 < num2){

alert(num3 + ','+ num1 + ',' + num2);

}else{

alert(num3 + ','+ num2 + ',' + num1);

}

1. 假如投资的年利率为5%，试求从1000块增长到5000块，需要花费多少年

1000 1000\*1.05

1050 1050\*1.05

var a = 1000;

var count = 0;

while(money < 5000){

money \*= 0.05;

count ++;

}

alert("一共需要"+count+"年");

1. 打印1-100之间所有奇数之和

for(var i = 1,sum = 0;i <100;i++){

if(i % 2 != 0){

sum += i;

}

}

console.log("1-100所有的奇数之和为："+ sum);

1. 打印1-100之间所有7的倍数的个数及总和

var count = 0;

for(var i = 1,sum = 0;i < 100;i ++){

if(i % 7 == 0){

sum += i;

count ++;

}

}

console.log("1-100的7的倍数个数："+ count);

console.log("1-100的7的倍数它们的和："+ sum);

1. 从键盘接收整数参数，如果该数为1-7，打印对应的星期，否则打印非法参数。

var a = +prompt("请输入一个整数");

// a = Number(a);

switch(a){

case 1:

console.log("你输入的是星期一");

break;

case 2:

console.log("你输入的是星期二");

break;

case 3:

console.log("你输入的是星期三");

break;

case 4:

console.log("你输入的是星期四");

break;

case 5:

console.log("你输入的是星期五");

break;

case 6:

console.log("你输入的是星期六");

break;

case 7:

console.log("你输入的是星期日");

break;

default:

console.log("你输入的是非法字符");

}

1. 水仙花数是指一个3位数，它的每个位上的数字的3 次幂之和等于它本身。

（例如：1^3 + 5^3 + 3^3 = 153）,请打印所有的水仙花数。

for(var i = 100;i < 1000;i ++){

var bai = parseInt(i / 100);

var shi = parseInt((i - bai\* 100)/10);

var ge = i % 10;

if(bai\*bai\*bai + shi\*shi\*shi + ge\*ge\*ge == i){

console.log(i);

}

}

1. 在页面中接收一个用户输入的数字，并判断该数是否是质数。

质数：只能被1和它自身整除的数，1不是质数也不是合数，质数必须是大于1的自然数。

var num = prompt("请输入一个大于1的整数");

if(num < 1){

alert("你输入的字符不合法");

}else{

var flag = true;

for(i = 2;i < num;i ++){

if(num % i == 0){

flag = false;

}

}

}

if(flag){

alert("你输入的是质数");

}else{

alert("你输入的不是质数");

}

1. 打印\*

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

//for(var j=0 ; j<i+1 ; j++)

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

document.write("\*&nbsp");

}

document.write("<br/>");

}

1. 打印99乘法表

<style type="text/css">

body{

width: 2000px;

}

span{

display: inline-block;

width: 80px;

}

</style>

<script type="text/javascript">

for(var i=1 ; i<=9 ; i++){

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

document.write("<span>" + j + "\*" + i + "=" + i\*j +"</span>" );

}

document.write("<br/>")

</span>

1. 打印100内的质数

for(var i=2 ; i<100 ; i++){

var flag = true;

for(var j=2 ; j<i ; j++){

if(i%j == 0){

flag = false;

}

}

if(flag){

console.log(i);

}

}

1. 习题

输出以下代码的执行结果

=========================================

var a = 123;

function fun(){

alert(a);

}

fun();

结果：123

=========================================

var a = 123;

function fun(){

alert(a);

var a = 456;

}

fun();

alert(a);

结果：undefined

123

=========================================

var a = 123;

function fun(){

alert(a);

a = 456;

}

fun();

alert(a);

结果：123

456

=========================================

var a = 123;

function fun(a){

alert(a);

a = 456;

}

fun();

alert(a);

结果：undefined

123

=========================================

var a = 123;

function fun(a){

alert(a);

a = 456;

}

fun(123);

alert(a);

结果：123

123

18.说出代码的输出结果

var a = 1

function c(a, b) {

console.log(a)

a = 2

console.log(a)

}

c();

结果：undefined

2

1. 写出下列代码的输出值：

function A() {} function B(a) {this.a = a;} function C(a) {if (a) {this.a = a;}}

A.prototype.a = 1;

B.prototype.a = 1;

C.prototype.a = 1;

console.log(new A().a);

console.log(new B().a);

console.log(new C(2).a);

结果：1

undefined

2

19.写出下列代码的输出值：

var a = 1;

function b() {

var a = 2;

function c() {

console.log(a);

}

return c;

}

b()();

结果：2

1. 定义一个Person类，类中包含name、age两个属性，并修改该类原型中的toString()方法，使其可以输出对象的具体信息。信息格式如下：Person[name=孙悟空, age=18]

function Person(name,age){

this.name = name;

this.age = age;

}

Person.prototype.toString = function(){

return "Person[name" + this.name + " , age = " + this.age + "]";

}

var per1 = new Person("宋江",28);

console.log(per1);

1. var arr = [1,2,1,3,4,2,2,2]; 将数组中重复的内容去除（for循环嵌套）

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

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

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

if(arr[i] == arr[j]){

arr.splice(j,1);

j--;

}

}

}

console.log(arr);

1. 去掉数组中未成年的元素

function Person(name,age){

this.name = name;

this.age = age;

}

Person.prototype.toString = function(){

return "Person = " + this.name + "age = " + this.age + "]";

};

var per1 = new Person("宋江",28);

var per2 = new Person("李逵",38);

var per3 = new Person("鲁智深",17);

var per4 = new Person("林冲",16);

var arr = [per1,per2,per3,per4];

var newArr = [];

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

if(arr[i].age >= 18){

newArr.push(arr[i]);

}

}

console.log(newArr);

1. 排序

var arr = [10,3,4,8,9,5,6,4,7,2,1];

arr.sort(function(a,b){

//从小到大

if(a > b){

return 1;

}else if(a == b){

return 0;

}else{

// a-b;

})

return -1;

}

console.log(arr);

arr.sort(function(a,b){

//从大到小

if(a > b){

return -1;

}else if(a == b){

return 0;

}else{

return 1;

}

// b-a;

})

console.log(arr);

1. 说说函数中的this到底是谁

this是函数的上下文对象，函数在调用时，由浏览器传递进来。根据函数调用的形式不同，this的值也不同：

1. 以函数的形式调用，this是window；
2. 以方法的形式调用，this是调用方法的对象；
3. 以构造函数的形式调用，this是新建的对象；
4. 使用call和apply方法调用，this是call和apply的第一个参数。

写出下列代码的输出值：

var obj = {

a: 1,

b: function () {console.log(this.a);}

};

var a = 2;

var objb = obj.b;

obj.b();

objb();

obj.b.call(window);

结果：1

2

2

26.

1、编写代码，让一个图片（test.jpg 宽高 100px\*100px ），在宽高500px \* 300px的块元素中水平及垂直方向居中（20分）

注意：至少写出2中解决方案.

方案一：

.box{  
 width: 500px;  
 height: 300px;  
 line-height: 300px;  
 text-align: center;  
}  
img{  
 vertical-align: middle;  
}

方案二：

.box{  
 width: 500px;  
 height: 300px;  
 display: table-cell;  
 text-align: center;  
 vertical-align: middle;  
}  
  
img{  
 vertical-align: middle;  
}

2、.box所在 div 的高度是多少？

.box {

background-color: yellow;

line-height: 200px;

font: bold 20px/3 "微软雅黑 Light";

}

<div class="box">Hello</div>

答案：60px

行高覆盖

3、如果一个div设置如下样式，浏览器（用户代理）在页面中"为其分配的高度"空间多大？ （20分）

.box {

height: 200px;

background-color: yellow;

margin-bottom: -100px;

}

答案：100px

1. 绝对定位和固定定位的包含块分别是谁？

答案：

绝对定位：

1. 初始包含块
2. 离其最近的开启了定位的祖先元素的padding-box

固定定位：

视口

1. 使用img 和 background-img 都可以显示一张图片，它们各自的应用场景？（20分）

答案：img，商品图片或者网站图标，承载网站重要信息。

background-img，不重要的网站信息，图标等。

27.

1、如何解决因子元素浮动造成的父元素(父元素使用了.box类选择器)高度塌陷？最低兼容IE6（20分）

备注：请在.box中完善代码

.box{

}

答案：.box{

overflow: hidden;

\*zoom: 1;

}

2、你能看到的颜色？ （20分）

<div class="wrapper">

<div class="box box1">box1</div>

<div class="box box2">box2</div>

<div class="box box3">box3</div>

</div>

.wrapper{ width: 300px; position: relative}

.box{ position: absolute; width: 100px; height: 100px;}

.box1 { background: red; z-index: 1}

.box2 { background: gold; }

.box3 { background: green; z-index: -1}

答案：红色

3、请问.box距离视口顶部的距离？（20分）

\* { margin: 0; padding: 0; }

body{

overflow: hidden;

}

.box{

margin-top: 5px;

background-color: yellow;

}

.box-sub{

width: 300px;

height: 300px;

background-color: green;

margin-top: 50px;

}

<div class="box">

<div class="box-sub"></div>

</div>

答案：50px

外边距重叠（塌陷）

解决方案：

（1）给父元素开启BFC：overflow: hidden;

（2）加padding-top: 1px;

（3）加border-top: 1px;

4、一个元素设置如下样式，请问该元素的宽度？（20分）

.box-sub{

right: 0;

top: 0;

position: fixed;

width: 100%;

height: 300px;

background-color: green;

}

答案：视口的宽度

5、以下代码的意图？（20分）

<!--[if lte IE 8]>

Welcome

<![endif]-->

答案：在IE8及以下浏览器显示Welcome

28.

一、以下代码在页面显示时存在哪些问题？如何解决？（20分）

<div class="box">

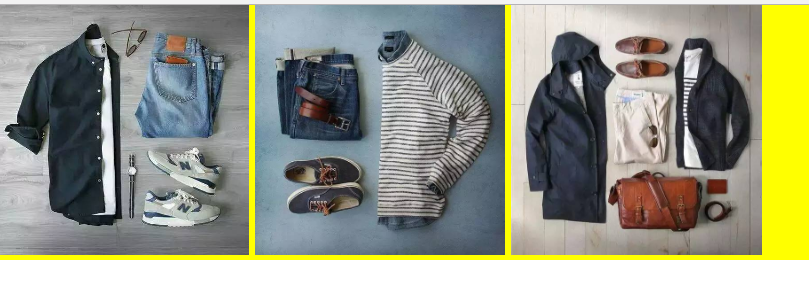
<img src="img/icon\_list.png">

<img src="img/icon\_list.png">

<img src="img/icon\_list.png">

</div>

提示：图片display 默认值是 inline-block。



图片之间有由于换行符造成的缝隙，因为父元素未指定高度，图片底部会沿着基线对齐，会留出一条缝隙。

解决换行符造成的缝隙方案：

（1）推荐使用

<img src="img/item01.jpeg" alt=""><!--  
--><img src="img/item02.jpeg" alt=""><!--  
--><img src="img/item03.jpeg" alt="">

1. 不推荐使用，如果出现文字，就会被覆盖

.box {  
 background-color: yellow;  
 font-size: 0;  
}

1. 不推荐使用，加浮动之后，需要给父元素开启BFC

img{  
 height: 200px;  
 float: left;  
}

解决基线对齐的方案：

img{  
 height: 200px;  
 vertical-align: middle;  
}

二、请画出，你 “看到的” 各矩形框(box)的大小，颜色，位置 (20分)

\* { margin: 0; padding: 0; }

.box-wrapper{ background-color: yellow; }

.box {

width: 33.3333%;

float: left;

height: 50px;

}

.box1{ background-color: green; }

.box2{ background-color: blue; }

.box3{ background-color: red; }

.box4{ background-color: deeppink; height: 100px; }

<div class="box-wrapper">

<div class="box box1"></div>

<div class="box box2"></div>

<div class="box box3"></div>

</div>

<div class="box4"></div>



三、请问Hello文所在的行高是？ (20分)

.box{

line-height: 300px;

font: bold 14px "华文宋体";

}

<div class="box">Hello</div>

答案：默认行高

行高覆盖

四、请问当鼠标点击a链接是，Hello World的颜色？（20分）

a{

font: bold 34px "华文宋体";

}

a:link{

color: orange;

}

a:visited{

color: red;

}

a:hover{

color: green;

}

a:active{

color: yellow;

}

<a href="#">Hello World</a>

答案：黄色

五、在实际开发工作中，如果一个样CSS样式没有生效，可能的原因？你如何快速排查？（20分）

备注：至少说出两种可能性。

答案：优先级不够；属性或者属性值写错；缓存

29.

一、请完善CSS代码，防止 .wrapper 高度塌陷，但是需要保证子元素内容的完整性。（20分）

\* { margin: 0; padding: 0; }

.box {

height: 80px;

width: 33.33333%;

float: left;

}

ol{ list-style: none; }

.box1{ background-color: yellow; }

.box2{ background-color: red; }

.box3{ background-color: greenyellow; }

<div class="wrapper">

<div class="box box1"></div>

<div class="box box2">

<ol>

<li>1</li>

<li>2</li>

<li>3</li>

<li>4</li>

<li>5</li>

<li>6</li>

<li>7</li>

<li>8</li>

</ol>

</div>

<div class="box box3"></div>

</div>

答案：

.wrapper:after{  
 content: "";  
 clear: both;  
 display: table;  
}

二、请问数字几显示为红色？（20分）

ol>li:nth-of-type(-5n + 8){

color: red;

}

<ol>

<li>1</li>

<li>2</li>

<li>3</li>

<li>4</li>

<li>5</li>

<li>6</li>

<li>7</li>

<li>8</li>

</ol>

答案：3 8

三、请完善CSS代码，在打印模式下，显示网页的链接？（20分）

<a href="http://www.atguigu.com">尚硅谷，让天下没有难学的技术。</a>

答案：

**@media** print {  
 a:before{  
 content: attr(href);  
 }  
}

四、让所有的input type=button 的元素显示为200\*100大小 （20分）

答案：

input[type='button']{  
 width: 200px;  
 height: 100px;  
}

1. 伪元素规范要求必须使用 ::， 但是哪两个伪元素可以使用: 为什么？

答案：:before和:after

兼容IE8

30.

一、编写代码，让一个宽高100\*100的div，缩小2倍，同时向左侧移动100px [注意：只能移动100px]（20分）

答案：

二、请仔细阅读代码，鼠标悬浮在.box时， 你看到的.box的宽高是多少？ （20分）

.box {

display: inline;

width: 100px;

height: 100px

background-color: yellow;

}

.box:hover{

transform: scale(2);

}

<div class="box">

Hello

</div>

三、当鼠标悬浮在.box时，宽度，高度，背景色的过渡时间？（30分）

.box{

width: 100px;

height: 100px;

background-color: rgba(255, 0, 0, .5);

}

.box:hover{

width: 200px;

height: 200px;

background-color: rgba(0, 0, 255, .5);

transition: width 1s -1s linear, all 3s .5s linear, background-color 1s;

}

四、触发过渡（transition）的条件？（20分）

五、设置块元素的（200\*300） transform-origin: right，如果使用像素表示，right应该是多少？（20分）

31.

一、页面中的 Hello 和 World 你能看见哪个？（20分）

.box {

transform-style: preserve-3d;

height:200px;

width: 200px;

position: relative;

transform: rotateY(180deg);

}

.front, .back{

position: absolute;

left: 0;

top: 0;

height: 100%;

width: 100%;

}

.front{

backface-visibility: hidden;

}

.back{

transform: rotateY(180deg);

}

<div class="box">

<div class="back">World</div>

<div class="front">Hello</div>

</div>

二、为什么设置景深？ 当样式中出现如下代码，该景深的计算起始点是？（20分）

perspective: 1200px;

三、页面刷新后，你看到的盒子的第一个颜色是（20分）

.box {

height:200px;

width: 200px;

animation: anim 3s backwards 1s;

background-color: blue;

}

@keyframes anim {

50%{

background-color: green;

}

to{

background-color: red;

}

}

<div class="box"></div>

四、你“看到”的盒子的大小？（20分）

.box {

height:200px;

width: 200px;

background-color: blue;

animation: anim 6s backwards -1s linear paused;

}

@keyframes anim{

to{

width: 800px;

}

}

<div class="box"></div>

<div style="height: 20px; width: 400px; background: #f00;"></div>

五、完善CSS代码，实现右半圆。（20分）

.box {

background-color: yellow;

width: 100px;

/\* 请在后面添加你的代码 \*/

}

1、

var a = new Object();

a.value = 1;

b = a;

b.value = 2;

alert(a.value);

答案：2

2、

var a = 10;

function fn1(a) {

console.log(a);

a = 20;

}

fn1();

console.log(a);

答案：undefined 10

分析：函数中传形参，相当于定义了一个局部变量，相当于var a。

3、

var a = 10;

function fn1(a) {

console.log(a);

a = 20;

}

fn1(a);

console.log(a);

答案：10 10

4、

var obj = {name:"孙悟空"};

function fn1(o) {

console.log(o.name);

o.name = "猪八戒";

}

fn1();

console.log(obj.name);

答案：报错

5、

var obj = {name:"孙悟空"};

function fn1(o) {

console.log(o.name);

o.name = "猪八戒"

}

fn(0bj);

console.log(obj.name);

答案：孙悟空 猪八戒

6、

var obj = {name:"孙悟空"};

function fn1(o) {

console.log(o.name);

o = new Object();

o.name = "猪八戒"

}

console.log(obj.name);

答案：孙悟空 孙悟空

7、

alert(a);//function a(){alert(4)};

var a = 1;

alert(a);//1

function a(){alert(2)};

alert(a);//1

var a = 3;

alert(a);//3

function a(){alert(4)};

alert(a);//3

解析：var 和 函数声明在预解析阶段已经加载好

8、

var a = 1;

function b() {

var a = 2;

function c() {

console.log(a);

}

return c;

}

b()();

答案：2

9、

for(var i=1 ; i<=10 ; i++){

setTimeout(function () {

console.log(i);

},1000\*i)

}

答案：输出10个11，隔一秒输出1个

解析：for循环在回调函数执行之前早就执行完了

10、

var a = 100;

function fn(){

alert(a);

var a = 200;

alert(a);

}

fn();//undefined 200

var a;

alert(a);//100

var a = 300;

alert(a);//300

11、

(function(){

var a = b = 3;

})();

console.log("a defined? " + (typeof a !== 'undefined'));

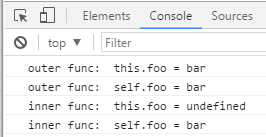
console.log("b defined? " + (typeof b !== 'undefined'));

答案：false true

解析：var a = b = 3;相当于 var a = b; b = 3;

12、

var myObject = {  
 **foo**: "bar",  
 func: function() {  
 var self = this;  
 //这时候的this指的是myObject  
 **console**.log("outer func: this.foo = " + this.**foo**);//bar  
 **console**.log("outer func: self.foo = " + self.**foo**);//bar  
 (function() {  
 //这时的this指的是window  
 **console**.log("inner func: this.foo = " + this.**foo**);//undefined  
 **console**.log("inner func: self.foo = " + self.**foo**);//bar  
 })();  
 }  
};  
myObject.func();



13、

var foo = "Hello";

(function() {

var bar = " World";

alert(foo + bar);

})();

alert(foo + bar);

答案：Hello World 报错

14、

function Foo() {

getName = function () { alert (1); };

return this;

}

Foo.getName = function () { alert (2);};

Foo.prototype.getName = function () { alert (3);};

var getName = function () { alert (4);};

function getName() { alert (5);}

//请写出以下输出结果：

Foo.getName();

getName();

Foo().getName();

getName();

new Foo.getName();

new Foo().getName();

new new Foo().getName();

function Foo() {  
 getName = function () { alert (1); };  
 return this;  
}  
//给函数对象中添加一个getName属性  
Foo.getName = function () { alert (2);};  
//给函数显示原型中添加一个getName属性  
Foo.**prototype**.getName = function () { alert (3);};  
//全局变量  
var getName = function () { alert (4);};  
//函数声明  
function getName() { alert (5);}  
  
//请写出以下输出结果：  
//调用函数的getName方法  
Foo.getName();//2  
//调用全局的getName  
getName();//4  
/\*调用Foo()返回值（window）的getName()，在调Foo()函数时，

将全局的getName改成了getName = function () { alert (1); };

\*/  
Foo().getName();//1  
  
getName();//1  
/\*以构造函数的形式调用getName()，相当于

new(Foo.getName)()

\*/  
new Foo.getName();//2  
//调用Foo实例的getName()  
new Foo().getName();//3  
  
//new(new Foo()).getName()  
new new Foo().getName();//3