**Web201215SZ（html+css+js基础）阶段考试**

## **基础**

1. 说明如下URL地址的组成部分及作用

[http://www.ceshi.com:80/test/test.html?par1=val1&par2=val2#p](http://www.ceshi.com:80/test/test.html?par1=val1&par2=val2" \l "p)

http://协议名，告诉服务器用什么约定和客户端进行通信，http://是可选的（从浏览器中输入网址再复制出来会自动加上http://），默认的协议

www.ceshi.com，服务器地址，要请求哪个服务器。

：80，连接服务器的端口号，代表连接服务器要从哪个门进去，默认为80端口，可选的

/test/test.html，文件路径，服务器上的路径定位指定的资源，可选的，可以有多层文件路径。

？par1=val1&par2=val2，查询字符串，用于给服务器端脚本语言传递参数，比如以后会将这些值传递给后端语言，查询字符串放在?以后，以 键=值 的方式来表示，多个查询字符串使用&来分隔，可选的。

#p，表示要获取/test/test.html这个资源中的子资源

1. 请解释绝对路径和相对路径(几种情况也需说明)

绝对路径：从头开始计算文件出现的路径。

D:/abc/ -- 代表根目录,绝对路径

/ -- 代表根目录,绝对路径。 如：<a href="/abc">文本</a> 或 <img src="/abc" />

相对路径：以引用文件之网页所在位置为参考基础，而建立出的目录路径。

情况：

上级 ../ -- 代表上一层目录，相对路径。 如：<a href="../abc">文本</a> 或 <img src="../abc" />

同级 ./ -- 代表目前所在的目录，相对路径。 如：<a href="./abc">文本</a> 或 <img src="./abc" />

上上级 ../../ -- 代表的是上一层目录的上一层目录，相对路径。 如：<img src="../../abc" />

## **HTML**

三、行内元素有哪些？块级元素有哪些？

行内元素：

span标签，img标签，a标签，i标签，一些表单标签（例如input，button）等

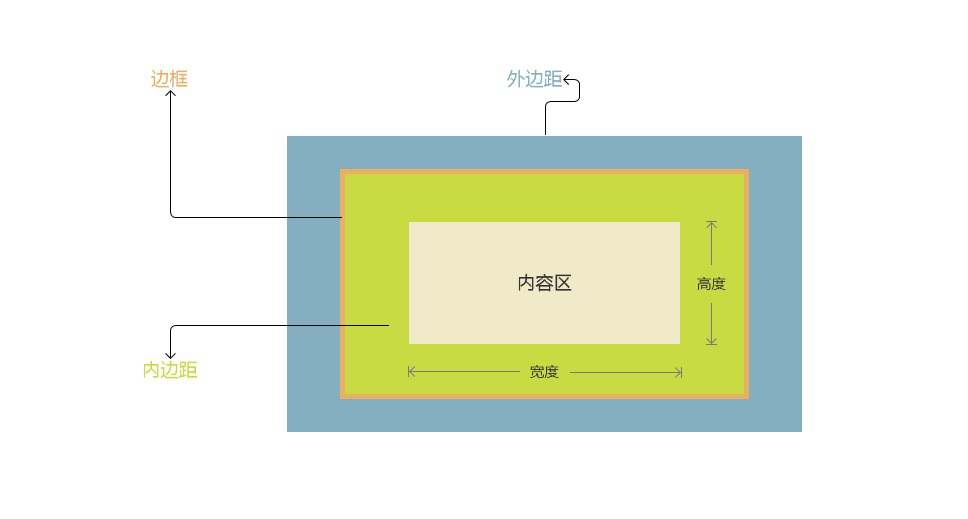
块级元素：

div标签，h标签，p标签，ul(ol)和li标签，section标签，article标签

...

四、盒子模型都由什么组成，都有什么作用？

看图：



五、HTML5新增了哪些元素？

一些语义和结构化元素：header, footer, nav, section, article, aside等

canvas元素

媒体元素：video, audio

新的表单元素，表单类型以及表单新属性（具体新增什么需要知道一下）

六、简述一下你对HTML语义化的理解

html语义化：用合适的标签去承载内容

作用：

* 语义化标签更具有可读性，便于团队的开发和维护
* 没有css的情况下，网页也能很好的呈现出内容结构和代码结构
* 关于SEO，搜索引擎更能理解到网页中各部分之间的关系，更准确更快速搜索信息

## **CSS**

七、说明下BFC，触发方式及其使用场景都是哪些？

一个块格式化上下文（block formatting context），是web页面的可视化CSS渲染出的一部分， 形成了BFC就形成了一个独立的区域，区域里面的子元素不会影响到外面的元素。

形成BFC的方法：

* 浮动元素：元素的float不是none的元素
* 绝对定位元素：position的值为absolute或fixed
* 内联块：display:inline-block
* 具有overflow并且值不是visible的块元素
* display为flow-root的值。(CSS3中新增加的)，只是单纯的触发BFC。

上面的1、2、3、4都可以形成BFC并且加上自己的特性（float形成BFC形成浮动，position形成BFC进行定位，内联块形成BFC改变元素状态，overflow形成BFC并且隐藏多出来的内容）。而flow-root只是单纯的触发BFC（形成的的是一个块元素）

使用场景：

* 解决float高度塌陷
* 解决父、子外边距折叠问题

八、写出你所知道的清除浮动的方法

1.在浮动元素后面增加空元素，并设置clear:both样式

2.使用在父元素后面使用伪元素::after，并设置：

#f::after{

content:'';

display:block;

clear:both;

}

3.利用BFC

1. 怎么做响应式设计？为屏幕宽度小于500px的设备设置样式，代码要怎么写？

使用媒体查询

@media screen and (max-width: 500px) {  
  xxx {  
    xxx:xxx  
 }  
}

十、使用弹性布局(flex)，完成下面.box相对于.main元素垂直水平居中需求（写出关键代码即可）

<html>

<head>

<style type="text/css">

.main{

width:500px;

height:500px;

border:1px solid #abc;

}

.box{

width:300px;

border:1px solid #333;

}

</style>

</head>

<body>

<div class="main">

<div class="box">未知高度上下左右居中</div>

</div>

</body>

</html>

.main{

display:flex

justify-content:center;

align-items:center

}

## **JS**

1. JS中的基本数据类型有哪些？引用类型（Object）的子类型有哪些？

基本数据类型（不包含Symbol）：

Number，String，Boolean，Undefined，Null

引用类型的子类型：

Object

包装对象（Number，String，Boolean），

Function

Date

Array

RegExp

Math

Error

JSON

备注：

引用类型的子类型也被称为内置对象或者标准库

十二、阅读代码，写出打印结果

var value = 1;

function bar() {

var value = 2;

foo();

console.log(value)

}

var foo

function foo() {

console.log(value)

}

bar(); // 1 2

十三、阅读代码，写出打印结果

var o3 = {m: 3}

var o4 = {m:4}

o4 = o3

o3 = {m: 5}

console.log(o4.m) // 3

console.log(o4===o3) // false

十四、说出你知道的数组方法（至少5个，越多越好），并说出哪些方法会改变原数组？

百度或者谷歌 搜索”Array mdn” 下拉找到左侧方法

改变原数组：

push(), pop(), shift(), unshift(), reverse(), sort(), splice()

不改变原数组：

concat(), join(), slice(),map(), filter(), forEach(), some(), every(), reduce()

1. call()，apply()怎么使用？

1.第一个参数都是this的指向（想不传的时候不填或者写undefined）

2.前面2个用于指定this指向后函数直接调用函数，不同在于他们参数的传递方式，apply()是通过数组去传递的，call则是一个个传递

例如：

1. **function** hello(name,age) {
2. console.log(name);
3. console.log(age);
4. }
5. hello.call(**this**,"tsrot",24);
6. hello.apply(**this**,["tsrot",24]);
7. 阅读代码，写出打印结果

function A() {}

A.prototype.a = function(){

console.log(1)

}

A.b = function(){

Console.log(2)

}

Object.prototype.a = function() {

console.log(3);

}

Object.prototype.b = function() {

console.log(4);

}

var a = new A();

a.a(); // 1

a.b(); // 4

console.log(a.constructor === A) // true

console.log(a instanceof A) // true

console.log(a instanceof Object) // true

1. 简述使用new实例化一个对象的时候发生了什么？

静哥笔记：

1. 创建对象，开辟内存空间。
2. 设置原型链
3. this指向该内存地址（让函数内部的this指向新生成的对象）
4. 执行函数代码
5. 将创建的对象实例返回。

华哥笔记（后面会学模拟new实现，所以补充了第四点，）：

1.创建一个空对象obj( {} )

2.将obj的[proto]属性指向构造函数的原型（即obj.[[protot]] = constrc.prototype）

3.将构造函数内部的this绑定到新建的对象obj，执行构造函数（也就是跟调用普通函数一样，只是此时函数的this为新创建的对象obj而已）

4.若构造函数没有返回非原始值（即不是引用类型），则返回该新建的对象obj（默认会添加return this）。否则，返回引用类型。

## **代码题**

十八、请写一个函数，实现打印100-999中含有2或者6的数字

1. **function** printNum(){
2. **for**(**var** n=100;n>=100&&n<=999;n++){
3. // 判断百位
4. **if**(n / 100 % 10 === 2 || n / 100 % 10 === 6){
5. console.log(n)
6. }
7. // 判断十位
8. **if**(n / 10 % 10 === 2 || n / 10 % 10 === 6){
9. console.log(n)
10. }
11. // 判断个位
12. **if**(n / 1 % 10 === 2 || n / 1 % 10 === 6){
13. console.log(n)
14. }
15. }
16. }

十九、请写函数，完成翻转字符串的功能，例如把"hello world"变成"dlrow olleh"

1. **function** reverseStr(str) {
2. **return** str.split(‘’).reverse().join(‘’)
3. }

二十、请写一个冒泡排序

1. **function** bubbleSort(arr) {
2. **var** len = arr.length;
3. **for** (**var** i = 0; i < len - 1; i++) {
4. **for** (**var** j = 0; j < len - 1 - i; j++) {
5. **if** (arr[j] > arr[j+1]) {        // 相邻元素两两对比
6. **var** temp = arr[j+1];        // 元素交换
7. arr[j+1] = arr[j];
8. arr[j] = temp;
9. }
10. }
11. }
12. **return** arr;
13. }

帮助理解网站：

https://visualgo.net/zh/sorting