

默写总结

- 总结人：杨标
- 总结时间：2022年8月23日
- 默写班级：H2204班

第一次默写

默写日期：2022年7月6日

1. 写出计数器使用的3个属性

- 你要对谁计数，那么就在这个元素的外层添加 `counter-reset` 属性
- 你要对谁计数，那么就在这个元素上面添加 `counter-increment` 属性
- 你要把统计的结果放在什么地方，那么就使用伪元素 `::before`/`::after` 来追加 `counter()` 属性

```
1  /* 你要对谁计数，那么就在这个元素的外层添加`counter-reset`属性 */
2  fieldset {
3      /* 重置一个计数器 ,也可以认为是定义一个计数器*/
4      counter-reset: aaa 0;
5  }
6
7  /* 你要对谁计数，那么就在这个元素上面添加`counter-increment`属性 */
8  fieldset>p {
9      counter-increment:aaa 1 ;
10 }
11 /* 你要把统计的结果放在什么地方，那么就使用伪元素`::before`/`::after`来追加`counter()``
    属性 */
12 fieldset>p::before{
13     content: counter(aaa);
14 }
```

根据上面的案例，我们可以得到3个属性操作

- `counter-reset`:计数器名子 初始值 初始化一个计数器，并初始化这个计数器值
- `counter-increment`:计数器名子 增量 在统计数量的时候，使用哪个计数器，并设置增量为多少
- `counter(计数器)` 从计数器里面拿出当前计数的值

2. 写出溢出处理的属性名与属性值

```
1  overflow:hidden;
2  overflow:auto;
3  overflow:visible;
4  overflow:scroll;
5  /*同时，还要注意一下overflow-x,overflow-y的使用方法*/
```

3. 边框的属性以及边框线条类型的属性值

```
1  border-style          /*边框属性*/
```

- solid 实线
- dashed 虚线
- dotted 点线
- double 双线
- groove 线槽

4. list-style-type下面任意的4个属性值

- none 取消前面的符号
- disc 实心圆
- circle 空心圆
- square 实心矩形
- upper-alpha 大写英文字母
- lower-alpha 小写英文字母
- upper-roman 大写罗马文
- lower-roman 小写的罗马文
- decimal 数值
- decimal-leading-zero 数值，但是以0开始，如 01
- cjk-heavenly-stem 采用天干记数，如 甲、乙、丙、丁、戊、己、庚、辛、壬、癸
- cjk-earthly-branch 采用地支纪年，如 子、丑、寅、卯、辰、巳、午、未、申、酉、戌、亥

5. list-style-position的属性值

- outside 符号在li这个元素的外部【默认值】
- inside 符号在li元素的内部

6. 圆角边框的属性名

```
1  border-radius          /*四个角*/
2  border-top-left-radius; /*左上*/
3  border-top-right-radius; /*右上*/
4  border-bottom-right-radius; /*右下*/
5  border-bottom-left-radius; /*左下*/
```

7. 文本居中

```
1  text-align:center;
```

8. 字体变为倾斜的状态用什么属性与属性值

- `font-style:italic` 使用任何字
- `font-style:oblique` 使字体倾斜

上面的两个方法都可以让页面上面的文字产生倾斜的效果，但是原理是不一样的，`italic` 先到系统的字体库里面去看，看有没有倾斜，如果有倾斜，则使用字体的斜体；如果系统的字体库里面没有这个字体的斜体，它会直接让文字倾斜（也就是退而求其次，使用 `oblique` 来倾斜）

9. 字体加粗的CSS属性及属性值，字体变细的属性值

`font-weight` 用于设置字体的权重

- `normal` 字体体质正常
- `bold` 字体加粗
- `bolder` 字体还粗一点
- `lighter` 字体变细

正常下完整的字体应该是有9个等级，分别是 `100~900`，如果是完整字体，我们可以设置数字来表示字体的权重，`normal` 对应的是400

10. 文字阴影怎么使用

```
1 text-shadow:水平偏移 垂直偏移 阴影模糊 阴影颜色;  
2 text-shadow: 20px 30px 3px blue;
```

11. 如何自定义字体

```
1 /* 自定义字体 */  
2 @font-face {  
3     /* 定义字体的名称 */  
4     font-family: "bgg";  
5     /* 说明字体在什么地方 */  
6     src: url("fonts/HKSN.ttf");  
7 }  
8 .aaa {  
9     font-family: "bgg";  
10 }
```

12. 单行文本溢出省略

```
1 p{  
2     white-space:nowrap;  
3     overflow:hidden;  
4     text-overflow:ellipsis;  
5 }
```

13. 多行文本溢出省略【3行】

```

1  p{
2      display: -webkit-box;
3      /* 盒子里面的内容垂直排列 */
4      -webkit-box-orient: vertical;
5      /* 在第2行的时候省略掉 */
6      -webkit-line-clamp: 2;
7      /* 溢出的部分隐藏 */
8      overflow: hidden;
9  }

```

14. background-size: 覆盖、包含

- contain 这个属性值是让背景图片完全显示在盒子里面，这样可能会有一个方向空出来
- cover 这个属性值是让背景图片完全覆盖住盒子，这样可能会有一个方向的图片被裁剪掉显示不出来

15. 设置左上角的圆角边框为30px

```

1  border-top-left-radius:30px;

```

16. CSS里面英文进行大小写转换用什么属性和属性值

text-transform 设置英文字母的大小写转换

- uppercase 大写字母
- lowercase 小写字母
- capitalize 单词的首字母

17. 文字大小什么CSS属性，有哪些单位？最小字体多少？默认字体多少？

- px 像素
- pt 字号
- em 一个父级元素【也是自己元素】的字体大小，em的全称是 element 元素
- rem 一个根元素字体的大小，它是一个响应式字体单位，它在 html 标签的 font-size 大小为 标准
- vw/vh 响应式字体单位，全称叫 viewport width 和 viewport height

最小字体是12px,默认字体是16px或12pt

18. 文本首行缩进

```

1  text-indent:2em;          /*首行缩进2个文字*/

```

19. 文字的装饰CSS，它由哪几个属性组成

text-decoration 文字的描述信息，它是一个简写的属性

- text-decoration-line 设置线条的位置
 - underline 下划线
 - overline 上划线
 - line-through 中划线（删除线）
 - none 不要设置任何线条

- `text-decoration-color` 设置的线条的颜色
- `text-decoration-style` 设置线条的类型
 - `solid` 实线
 - `dashed` 虚线
 - `dotted` 点线
 - `double` 双线
 - `wavy` 波浪线

三个属性合起来就变成了上面的 `text-decoration` 一个属性

20. CSS常用的颜色值类型有哪几种?

- 枚举色
- 十六进制色
- RGB三元色
- RGBA透明度颜色

这里要注意, RGBA是可以设置一个颜色的透明度, 同时 `transparent` 是全透明的意思

21. a标签里面target的属性值有哪些?

- `_blank` 新容器打开链接
- `_self` 当前容器打开链接
- `_frameName` 在指定的iframe里面打开
- `_parent` 在父级打开
- `_top` 在顶级容器打开

22. 表格里面, 行合并、列合并

- 行合并 `span`
- 列合并 `colspan`

23. 在CSS外部样式表里面, 有一个文件a.css,怎么导入到网页里面

```
1 <link rel="stylesheet" type="text/css" href="a.css">
2 <style>
3     @import url("a.css");
4 </style>
```

24. 如何将网页设置为UTF-8(中文)

```
1 <meta charset="utf-8">
```

25. 写出HTML里面特殊的文字标记

```
1 &gt;
2 &lt;
3 &deg;
4 &nbsp;
5 &reg;
6 &copy;
```

第二次默写

默写时间：2022年7月13日

1. 如何让后代元素主动的去继承父级元素的样式【属性】

inherit

2. 相对定位，绝对定位的属性值

- 相对定位: relative
- 绝对定位: absolute

3. 字体倾斜的2个单词

```
1 font-style:italic;  
2 font-style:oblique;
```

4. 块级元素让宽度适应内容

```
1 width:fit-content;
```

5. 背景图大小里面的覆盖、包含

- 覆盖cover
- 包含contain

6. 盒子模型由外向内4层分别是什么

margin-border-padding-content

7. 省略号单词

ellipsis

8. 英文单词进行大小写转换的属性，以及3个属性值

text-transform

uppercase

lowercase

capitalize

9. CSS层叠性里面，一个元素最终样式是有哪几部分样式来共同决定的

行内样式>内部样式块>默认样式>继承样式

10. 标准流布局里面，各种元素的水平居中方案

- 块级: margin:auto
- 行内，行内块级：外层块级或行内块这样上面加 text-align:center

11. 浮动脱流以后的元素居中方案

- 父级元素变 inline-block，再到父级设置 text-align:center
- 父级宽度设置 fit-content 然后 margin:auto

12. 定位脱流以后的元素水平居中方案

```

1  div{
2      left:0;
3      right:0;
4      margin:auto
5  }

```

13. 如果恢复浮动以后的元素的父级元素高度

- 在浮动元素的后面添加块级元素clear:both;
- 在浮动元素的父级添加overflow
- 使用伪类::after

14. 什么样的元素类型不遵守标准盒子模型规范，是怎么不遵守的？

- 行内元素不遵守，margin-left/margin-right正常
- margin-top/margin-bottom没有
- padding-left/padding-right是正常
- padding-top/padding-bottom只能撑开自身，撑不开父级

15. 解决margin穿透有什么办法？

- border-top
- BFC:overflow,float,absolute

16. 一个元素宽度丢失了会是什么原因？

脱流，元素类型转换为inline

17. 绝对定位元素以谁为参照物进行定位

以父级的除static的定位为参照物

或父级transform为参照物

18. border-color如果将颜色设置成与内容color保持一致，使用什么属性值？

```

1  border:1px solid currentcolor;

```

主要使用的 currentcolor 这个属性值

19. 一个元素使用定位以后，会多出哪几个CSS属性？

left/right/top/bottom/z-index

20. 请列出权重值为10的CSS选择器

类，伪类，属性

21. 请选出fieldset元素下面的除了第2个h2标签的所有标签

```

1  fieldset *:not(h2:nth-of-type(2))

```

22. 请选出ul下面的li中前3项li

```

1  ul>li:nth-child(-n+3)

```

23. 多行文本溢出省略

```
1 display:-webkit-box;
2 -webkit-box-orient:vertical;
3 -webkit-line-clamp:2;
4 overflow:hidden;
```

24. 滤镜的属性及常用的几个属性值（将页在统一调成灰色）

```
1 filter:blur(2px)
2 filter:alpha(opacity=50);
3 filter:hue-rotate(30deg)
4 filter:grayscale(0.5);
```

25. border-radius四个或八个值分别代表什么？

```
1 border-radius: 50px 0px 0px 0px ; /*简写*/
2 border-radius: 50px 0px 0px 0px / 50px 0px 0px 0px; /*完整写法*/
```

前面的4个值代表横轴半径，后面的3个值代表纵轴半径

```
1 border-radius: 150px 0px 0px 0px / 50px 0px 0px 0px;
```

第三次默写

默写时间：2022年7月20日

1. 列出不能执行过渡的属性【至少写6个】

- float
- display
- visibility
- overflow
- position
- z-index
- font-family
- `text-align/text-align-last
- cursor

2. CSS3对不同浏览器兼容性添加的前缀是什么？

对于低版本的浏览器，我们需要添加特定的前缀

- 以谷歌为核心的浏览器，我们要添加 `-webkit-`
- 以IE为核心的浏览器则要添加 `-ms-`
- 火狐浏览器则是使用 `-moz-`
- 欧朋浏览器使用 `-o-`

3. CSS3里面过渡的4个属性分别是什么？

- `transition-property` 要执行过渡的属性
- `transition-duration` 要执行过渡的时间，它可以以 `s` 秒为单位，也可以以 `ms` 毫秒为单位】
- `transition-timing-function` 过渡的效果，也叫过渡的时间函数【默认值是 `ease` 】
 - `linear` 匀速
 - `ease-in` 先慢后快，`ease` 单词是轻松的意思，`in` 是开始的意思
 - `ease-out` 先快后慢，`out` 代表结束的意思
 - `ease-in-out` 前后都很慢，中间很快
 - 我们还可以在浏览器里面手动的编辑我们需要的效果，如 `cubic-bezier(0.74, -1.06, 0.35, 2.34)`
- `transition-delay` 过渡等待【默认值0s】

4. 变换的属性是什么？

```
1 transform
```

5. 设置变换的起点的属性是什么？

```
1 transform-origin
```

6. 位移，缩放，旋转，变形的属性名分别是什么？

- 位移 `translate()/translateX()/translateY()/translateZ()/translate3d()`
- 缩放 `scaleX()/scaleY()/scaleZ()/scale3d()/scale()`
- 旋转 `rotateX()/rotateY()/rotateZ()/rotate()/rotate3d()`
- 变形 `skewX()/skewY()/skew()`

7. 如何设置元素变换以后背对用户时不可见？

```
1 backface-visibility:hidden;
```

8. 形成3D变换的条件是什么？

- 设置视距 `perspective:距离大小`
- 开启3D空间 `transform-style: preserve-3d`

9. 多个变换结合时语法格式是什么样子的？

```
1 transform:变换1 变换2 ...;
```

10. 变换的注意事项有哪2个点？

- 所有的行内元素都不支持 `transform` 变换的，除非转换成块级或行内块级元素

- 不要将 `background-color` 与 `preserve-3d` 一起使用

11. 线性渐变的语法格式?

```
1 background-image:linear-gradient(to 方向,颜色1 [开始位置] [结束位置],颜色2 [开始位置] [结束位置]...);
```

12. 径向渐变的语法格式?

```
1 background-image:radial-gradient([形状 at 横坐标 纵坐标],颜色1 [开始位置] [结束位置]...);
```

13. 径向渐变的形状, 圆与椭圆?

- 圆 `circle`
- 椭圆 `ellipse`

14. 圆锥渐变的语法格式?

```
1 background:conic-gradient(颜色1 [开始位置] [结束位置],...);
```

15. 定义动画的基本语法格式?

```
1 @keyframes 动画名称 {  
2     from{}  
3     to{}  
4 }
```

16. 在目前所学习的css里面, 有哪些以@开头的命令, 作用分别是什么?

- `@charset utf-8;` 设置字符串
- `@font-face` 自定义字体
- `@import url("a.css")` 导入外部的CSS文件
- `@keyframes` 定义动画
- `@media` 媒体查询

17. 使用动画的8个属性列出来?

- `animation-name` 动画的名称【必填】
- `animation-duration` 动画执行一次的时间【必填】
- `animation-iteration-count` 动画重复的次数【默认值是1】, 如果希望动画一直重复执行, 则可以通过设置 `infinite` 无穷大来实现
- `animation-timing-function` 动画执行的时间函数【默认值是 `ease`】, 如果希望动画匀速执行可以设置 `linear`, 这里的属性值和 `transition-timing-function` 保持一致
- `animation-delay` 动画的等待时间【默认值为0】
- `animation-direction` 动画执行的方向, 【默认值 `normal`】
 - `normal` 正常的

- `reverse` 逆向的
- `alternate` 正向与逆向交替运行
- `alternate-reverse` 逆向与正向交替运行
- `animation-play-state` 动画的播放状态
 - `running` 运行状态【默认值】
 - `paused` 暂停状态
- `animation-fill-mode` 动画在结束以后停留在什么状态
 - `backwards` 回到开始状态
 - `forwards` 停留在结束状态

18. 动画在停止的时候的状态有2个，分别是回到开始状态的和结束状态的？

`animation-fill-mode` 动画在结束以后停留在什么状态

- `backwards` 回到开始状态
- `forwards` 停留在结束状态

19. 动画在执行的时候的方向的属性值4个分别是什么？

- `normal` 正常的
- `reverse` 逆向的
- `alternate` 正向与逆向交替运行
- `alternate-reverse` 逆向与正向交替运行

20. 动画执行时无穷大的值是什么？

```
1 animation-iteration-count:infinite;
```

21. CSS里面如何定义全局变量，如何使用变量？

```
1  /*这里的伪类:root相当于html标签*/
2  :root{
3      --a:100px;    /*定义变量*/
4  }
5  .div1{
6      width:var(--a);    /*使用变量*/
7  }
```

22. CSS里面如何进行四则算术运算，有什么注意事项？

```
1  .div{
2      width: calc(100px + 100px);    /*符号的左右必须要加上空格*/
3  }
```

23. 能够形成BFC的属性和属性值有哪些？

- 浮动元素，`float` 除 `none` 以外的值；
- 绝对定位元素，`position` (`absolute`, `fixed`) ；
- `display` 为以下其中之一值 `inline-block`, `table-cell`, `table-caption`, `flex`弹性盒子；
- `overflow` 除了 `visible` 以外的值 (`hidden`, `auto`, `scroll`)

- 元素加上display:flow-root (没有副作用, 但是兼容性差。display:flow-root属性是css新增的属性, 专门用来触发BFC, 不干别的。)

24. 5种定位分别是什么?

- 相对定位 relative
- 绝对定位 absolute
- 固定定位 fixed
- 静态定位 static
- 粘性定位 sticky

25. 在CSS里面, 滤镜的属性和属性值?

```
1 filter:blur(2px)
2 filter:alpha(opacity=50);
3 filter:hue-rotate(30deg)
4 filter:grayscale(0.5);
```

第四次默写

2022年8月10日

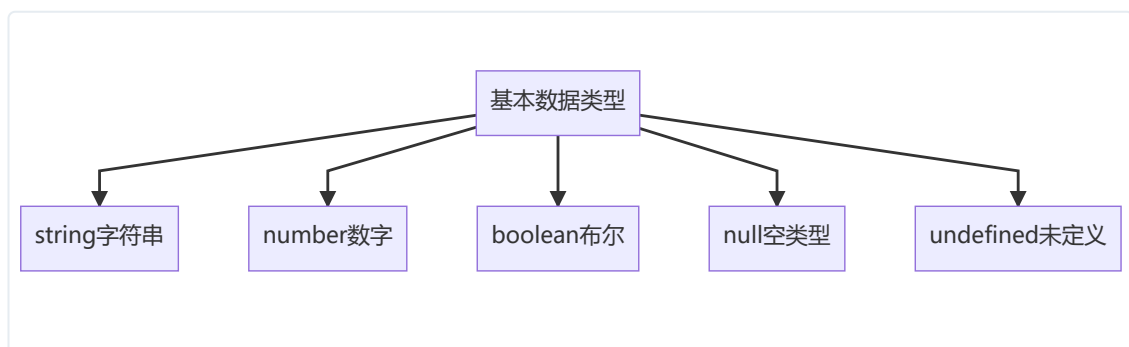
1. 二进制转十进制

```
1 var a = 10110 ;
2 parseInt(a,2);           //parseInt得到的结果一定是10进制
```

2. 下面代码结果是什么?

```
1 Number(null);           //0
2 Number(undefined);      //NaN
```

3. JS基本数据类型有哪些?



4. 下面代码运行结果是什么?

```
1 console.log("0" == false);           //true;
```

5. 下面代码运行结果是什么?

```
1 console.log(2%0);                     //NaN
```

6. 下面代码运行结果是什么?

```
1 var x = ['abcde',123456];  
2 var y = typeof typeof x[1];          //"string"
```

7. 下面代码结果是什么?

```
1 var arr = ["a","b","c","d","e"] ;  
2 arr.slice(2,4);  
3 console.log(arr);                    //[ "a","b","c","d","e"]  
4 //slice()方法提取数组的元素，原数组不会改变
```

8. 相等操作符，关系运算符，布尔运算符，赋值运算符，算术运算符 优先级从高到低排列

- `.() []`
- 一元操作符 `++,--`
- 算术运算符[先乘除，后加减]
- 关系操作符 `>,<,>=,<=,==,!=`
- 逻辑操作符[先非，再与，最后或]
- 三目运算符
- 赋值运算符

9. 字符串转数字有哪些方法

```
1 var a = "123";  
2 parseInt(a);  
3 parseFloat(a);  
4 var b = +a;           //一元操作符转换  
5 Number(a);           //默认的隐式转换
```

10. 使用条件表达式找出最大的变量

```

1  var a =1,b=2,c=3;
2
3  //第一种
4  var max = (a > b ? a : b) > c ? (a > b ? a : b) : c;
5
6  //第二种
7  // var max = a > b ? (成立以后a,c比较) : (不成立b,c比较);
8  var max = a > b ? (a>c?a:c) : (b>c?b:c);

```

11. JS里面有哪些值会被认为是false

```

1  //javascript一定要注意6个明确的false条件
2  //空字符串"",false,0,NaN,null,undefined

```

12. 退出循环中断循环的关键字是什么?

答: 退出循环的关键字是 `break` ;

13. 函数的返回值使用什么关键字?

答: 函数里面通过 `return` 返回

14. 定义数组的方法有哪些?

```

1  var arr = new Array();
2  var arr = new Array(2);           //定义长度
3  var arr = new Array(1,3,"a");    //静态初始化

```

15. arr.length是多少?

```

1  var arr = [2,1,,3,7,];
2  //在定义数组的时候 , 最后的一个逗号会被忽略, 但是中间的逗号是正常的, 所以arr数组的长是5

```

16. 数组的栈方法

- `push()` 入栈
- `pop()` 出栈

17. 数组的队列方法

- `unshift()` 插队
- `shift()` 出队

18. 使用什么方法可以得到一个相同的互不影响的数组

这个题目应该是要使用深拷贝的, 但是如果数组里同存放的是基本数据类型, 则可以通过下面的方法

```
1 var arr = [1,2,4,"a","b"];
2 var arr1 = arr.slice();           //第一种方式
3 var arr2 = arr.concat();          //第二种方式
```

19. 数组里面 `splice` 方法怎么使用，参数各代表什么？

```
1 arr.splice(开始索引,删除元素个数,要放入的新元素...);
```

20. `forEach`与`map`迭代的区别在哪里？

答: `forEach` 与 `map` 都是执行的静态遍历，但是 `map` 可以接收回调函数的返回值，将每次回调函数的返回值变成一个新的数组

21. 如何检测一个变量是否是数组

```
1 var arr = new Array();
2 arr instanceof Array;           //第一种方式
3 Array.isArray(arr);              //第二种方法
```

22. 函数中实参数组叫什么？

答: 函数的实参数组是 `arguments`

23. 写出下面代码的运行结果

```
1 var a=true,b=false,c=null,d=undefined;
2 ""|b&& c;           //false
3 ""&&b&&c;           //""
4 ""&&b|c;             //null
5 !""|b&&c;           //true
6 !d|b&&c;            //true
```

24. 写出下面代码的运行结果

```
1 var arr = new Array("3.1");           //正常定义数组，里面有一个元素"3.1"
2 var arr2 = new Array(-1);             //报错，长度不能为-1
3 var arr3 = new Array(Number("1"));     //正常定义数组，长度为1
```

25. 请写出"A", "a", "0"的ascii码

0011		01/0		0101		0110		0111	
3		4		5		6		7	
十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符
48	0	64	@	80	P	96	`	112	p
49	1	65	A	81	Q	97	a	113	q
50	2	66	B	82	R	98	b	114	r
51	3	67	C	83	S	99	c	115	s
52	4	68	D	84	T	100	d	116	t
53	5	69	E	85	U	101	e	117	u
54	6	70	F	86	V	102	f	118	v
55	7	71	G	87	W	103	g	119	w
56	8	72	H	88	X	104	h	120	x
57	9	73	I	89	Y	105	i	121	y
58	:	74	J	90	Z	106	j	122	z
59	;	75	K	91	[107	k	123	{
60	<	76	L	92	\	108	l	124	
61	=	77	M	93]	109	m	125	}
62	>	78	N	94	^	110	n	126	~
63	?	79	O	95	_	111	o	127	△

第五次默写

1. 写出下面的结果

```

1 console.log([] + []);           //""
2 console.log({} + []);           //"[object Object]"
3 console.log([] == ![]);         //true      注意推断过程

```

2. new关键字干了什么事？

要创建 Person 的新实例，必须使用 new 操作符。以这种方式调用构造函数实际上会经历以下 4 个步骤：

- (1) 创建一个新对象；
- (2) 将构造函数的作用域赋给新对象（因此 this 就指向了这个新对象）；
- (3) 执行构造函数中的代码（为这个新对象添加属性）；
- (4) 返回新对象。

3. 如果获取对象所有的属性名

```

1 Object.getOwnPropertyNames(obj); //获取obj对象的所有属性名

```

4. 数据属性有哪4个特征（特性）

1. 数据属性

数据属性包含一个数据值的位置。在这个位置可以读取和写入值。数据属性有 4 个描述其行为的特性。

- ❑ `[[Configurable]]`: 表示能否通过 `delete` 删除属性从而重新定义属性, 能否修改属性的特性, 或者能否把属性修改为访问器属性。像前面例子中那样直接在对象上定义的属性, 它们的这个特性默认值为 `true`。
- ❑ `[[Enumerable]]`: 表示能否通过 `for-in` 循环返回属性。像前面例子中那样直接在对象上定义的属性, 它们的这个特性默认值为 `true`。
- ❑ `[[Writable]]`: 表示能否修改属性的值。像前面例子中那样直接在对象上定义的属性, 它们的这个特性默认值为 `true`。
- ❑ `[[Value]]`: 包含这个属性的数据值。读取属性值的时候, 从这个位置读; 写入属性值的时候, 把新值保存在这个位置。这个特性的默认值为 `undefined`。

5. 构造函数与普通函数的区别

- 一个函数如果以 `new` 去调用那么它就是构造函数, 如果只是通过 `函数名+()` 这种形式调用它就是普通函数
- 普通函数的返回值是通过 `return` 来完成的, 而构造函数的返回值是**自动返回的**【注意返回对象与返回普通类型的区别, 这一点其实很少遇到, 但还是要注意】
- 构造函数里面的 `this` 指向了当前构造函数所创建的对象, 而普通函数里面的 `this` 指向了浏览器的全局对象 `window`
- 构造函数里面的 `this` 指向了当前构造函数所创建的对象, 而普通函数里面的 `this` 指向了浏览器的全局对象 `window` 【这个知识点在后面的DOM里面会讲到】

6. 下面代码的运行结果?

```
1 function a() {}
2 a instanceof Function ;           //true
3 //这个题目其实充分的说明了, 函数也是对象, 因为 var fun = new Function() 这也可以创建对象,
4 //同时要注意 new出来的就一定是对象
```

7. 定义函数的方法有哪些?

```
1 var a = new Function();           //第一种方式
2 function b() {}                   //第二种方式
3 var c = function () {}           //第三种方式
```

8. 解释一下什么是回调函数?

答: 回调函数就是把函数当成参数传递到另一个函数里面。回调函数也叫回调方法, 它可以解决当函数的返回值不无法回, 或有些逻辑需要分段执行的时候

9. 解释一下什么是深拷贝, 什么是浅拷贝?

答: 深拷贝就是深入到内存的堆里面去拷贝, 这个拷贝出来的2个对象就完全互不影响; 浅拷贝只是拷贝了栈里面的值

```
1 var obj1 = {
2   username: "张珊",
3   sex: "男",
```

```

4     computer: {
5         type: "国产",
6         price: 3999,
7         name: "RedmiBook16"
8     },
9     boyFriends: ["男朋友1号", "男朋友2号", "备胎1号", "2号鱼"]
10 }
11
12 //现在想对上面的对象实现完整的拷贝，怎么办呢？
13
14 //我给你一个对象，你还我一个互不影响 的相同对象
15 function deepCopy(oldObj) {
16     if (typeof oldObj !== "object" || oldObj === null) {
17         //说明不是对象，或是null，直接返回
18         return oldObj;
19     }
20     //第一步：创建对象
21     var newObj = Array.isArray(oldObj) ? [] : {};
22     var arr = Object.getOwnPropertyNames(oldObj);
23     arr.forEach(function (item) {
24         newObj[item] = deepCopy(oldObj[item]);
25     });
26     return newObj;
27 }
28
29 var obj2 = deepCopy(obj1);

```

10. 列出数组所有的迭代方法

ECMAScript 5 为数组定义了 5 个迭代方法。每个方法都接收两个参数：要在每一项上运行的函数和（可选的）运行该函数的作用域对象——影响 `this` 的值。传入这些方法中的函数会接收三个参数：数组项的值、该项在数组中的位置和数组对象本身。根据使用的方法不同，这个函数执行后的返回值可能会也可能不会影响方法的返回值。以下是这 5 个迭代方法的作用。

- ❑ `every()`：对数组中的每一项运行给定函数，如果该函数对每一项都返回 `true`，则返回 `true`。
- ❑ `filter()`：对数组中的每一项运行给定函数，返回该函数会返回 `true` 的项组成的数组。
- ❑ `forEach()`：对数组中的每一项运行给定函数。这个方法没有返回值。
- ❑ `map()`：对数组中的每一项运行给定函数，返回每次函数调用的结果组成的数组。
- ❑ `some()`：对数组中的每一项运行给定函数，如果该函数对任一项返回 `true`，则返回 `true`。

以上方法都不会修改数组中的包含的值。

11. 代码运行的结果是什么？

```

1  var arr = ["a","b","c"];
2  arr.forEach(function(item,index){
3      if(index==1){
4          arr.push("d");
5      }
6      console.log(item);
7  });
8
9  //打印a,b,c    原因:forEach执行的是静态遍历

```

12. //写代码，在第二个"b"元素的后面插入一个"标哥"

```

1  var arr = ["a","b","c","d","e","b","g","b","c"];
2  //代码如下
3  //第一步: 先找出第1个"b"的位置
4  var index = arr.indexOf("b");
5  //第二步: 再找出第2个b的位置
6  var index2 = arr.indexOf("b",index);
7  //第三步: 插入
8  arr.splice(index2+1,"b");

```

13. 在数组里面有哪些方法在执行以后不会影响原数组?

- concat()
- slice()
- 迭代方法 forEach/map/filter/every/some
- 归并方法 reduce/reduceRight
- indexOf/lastIndexOf/toString()/join()

14. 对象调用属性的方式有哪些?

```

1  对象.属性
2  对象["属性"];

```

15. 请说明一下obj对象里面有几个属性，属性值分别是什么?

```

1  var a = "bgg";
2  var b = a;
3  var obj = {
4      [a]: "hello",           //中括号代表的是变量，所以[a]相当于"bgg"
5      [b]: "world"           //[b]也相当于"bgg"
6  }
7  //两个属性个同，所以只有1个属性值，为"world";

```

16. 如果删除对象的某一个属性?

```
1 delete 对象.属性;
```

17. 通过什么方法可以获取对象属性的描述信息

```
1 Object.getOwnPropertyDescriptor(对象);
```

18. 如果判断对象是否具备某一个属性

```
1 /*第一种方法*/  
2 属性 in 对象 //判断自己或原型父级上面有没有这个属性  
3  
4 /*第二种方式*/  
5 对象.hasOwnProperty(属性) //只判断自己是否包含这个属性
```

19. 以下代码输出什么?

```
1 var name = "sex"  
2 var company = {  
3     name: "qdywxs",  
4     age: 3,  
5     sex: "男"  
6 }  
7 console.log(company[name]); //这里的name没有加引号, 是一个变量代表"sex", 结果就是"男"  
8 console.log(company.name); //这里的name是一个普通属性, 结果就是"qdywxs";
```

20. 对象在执行加法与减法的时候分别是怎么操作的?

答: 对象在执行加法, 则字符串优先, 先调用 `toString()`。

对象在执行减法操作的时候, 数字优先, 先调用 `valueOf()`, 调用失败则会再次调用 `toString()`

21. `call`/`apply`有什么区别, 作用是什么?

答: `call/apply` 可以改变函数内部 `this` 的指向, `call` 有呼叫谁来调用方法的意思, `apply` 有申请谁过来调用方法的意思

22. 以下代码结果是什么?

```
1 var arr = [];  
2 arr.push===Array.prototype.push;  
3 //结果为true, 始终要注意一个点, 一个对象的__proto__应该等于它的构造函数的prototype
```

23. 如何判断一个变量是整数？

```
1 //第一种方法
2 Number.isInteger(a);
3 //第二种方法
4 parseInt(a)===a;
5 //第三种方法
6 a%1===0;
```

24. 如何判断一个变量它是NaN？

```
1 Number.isNaN(a);
```

25. 怎么定义一个变量a，才会有下面的结果

```
1 a==1&&a==2&&a==3 //结果为true
2
3 //答案如下：这里要注意，在执行判断操作的时候，要执行数字优先的原则，所以会优先调用
  valueOf的方法
4 //这里我们就应该想到要使用对象了
5 var a = {
6     i:0;
7     valueOf:function(){
8         i++;
9         return i;
10    }
11 }
```