

1. 什么是语义化HTML? 常见的语义化HTML标签有哪些?

- 1 **&1**
- 2 html语义化标签指的是可以通过标签名快速确定标签的内容是什么
- 3 html语义化能够让页面的内容结构更加简单易懂, 便于搜索引擎解析, 便于阅读维护和理解
- 4 **&2** 常见的语义化标签: `<header>` `<footer>` `<nav>` `<aside>` `<section>` `<main>`

2. 行内元素和块级元素的区别?

- 1 块级元素: 独占一行, 默认宽度100%, 可设置宽高
- 2 行内元素: 共享一行, 宽高由内容撑起, 不可设置宽高
- 3
- 4 常见块级元素: `<div>` `<p>` `<h1>`系列标签 ...
- 5 常见行内元素: `` `<a>` `` ...

3. CSS选择器优先级如何计算?

- 1 **&1** !important > id选择器 > 类选择器 > 标签名选择器 > * > 默认 > 继承样式
- 2 权重 1000 100 10 1
- 3
- 4 **&2** 优先级算法三准则:
- 5 1. 内嵌式 > 内联式 > 外链式
- 6 2. 权重相加, 权重大的样式优先
- 7 3. 权重相同的情况下, 采用就近原则, 后加的样式优于前面的样式

4. CSS可以被继承的样式独有哪些?

- 1 文字样式 和 列表样式 可以被继承

5. 清除浮动的方式有哪些?

- 1 **&1** 给父元素添加高度
- 2 缺点: 不灵活
- 3
- 4 **&2** 在浮动的元素后面, 添加一个空的div, 并添加样式clear:both;
- 5 缺点: 代码冗余(存在多余的代码)
- 6
- 7 **&3** 给父元素添加overflow:hidden; (有两个作用: 清除浮动影响, 和溢出隐藏)
- 8 缺点: 超出父元素范围的内容不显示
- 9
- 10 **&4** 使用伪类, 用的比较多
- 11 .clearfix:after{

```
12     content: "";
13     display: block;
14     clear: both;
15 }
```

6. 定位的方式以及区别?

```
1  relative 相对定位:
2      相对于自身位置进行偏移, 不会脱离文档流
3  absolute 绝对定位:
4      相对于最近一级添加了定位的父元素进行定位, 会脱离文档流,
5  fixed 固定定位:
6      相对于窗口进行定位, 会脱离文档流
7
8  使用场景:
9      相对定位与绝对定位搭配使用, 子绝父相
10     固定定位多数用于模态框/浮动菜单
```

7. 盒子模型的结构以及类型和区别?

```
1  盒子结构 = 内容(设置的width/height) + 内边距padding + 边框border + 外边距margin;
2  标准盒子的宽/高 = 内容(设置的width/height) + 内边距padding + 边框border;
3  怪异盒子的宽/高 = 设置的width/height;
4
5  可以通过box-sizing切换盒子模型样式:
6      border-box 怪异盒子
7      content-box 标准盒子
```

8. 如何让一个盒子绝对居中?

```
1  &1. 第一种
2      .父{ position: relative; }
3      .儿{
4          position: absolute;
5          left: 50%;
6          margin-left: -儿宽度/2;
7          top: 50%;
8          margin-top: -儿高度/2
9      }
10
11  &2. 第二种
12      .父{ position: relative; }
13      .儿{
14          position: absolute;
15          left: 50%; top: 50%;
16          transform: translate(-50%, -50%); /*css3平移*/
17      }
18
19  &3. 第三种
```

```
20 .父{ position: relative; }
21 .儿{
22     position: absolute;
23     left: 0; right: 0; top: 0; bottom: 0;
24     margin: auto;
25 }
```

9. 表单的method属性中get请求与post请求的区别?

```
1  &1. get
2      1. 请求参数随着(地址栏)一起发送给后台-可见. url?参数=值&参数=值
3      2. 安全性较低
4      3. 传输数据大小有限制, 一般不超过4kb
5      4. 通常用于查询数据
6
7  &2. post
8      1. 请求参数随着(请求体)一起发送给后台-不可见
9      2. 安全性较高
10     3. 传输数据大小没有限制
11     4. 通常用于操作数据(增加 删除 修改)
```

10. JS的数据都有哪些?

```
1  &1. 基本类型
2      1.Number 2.String 3.Boolean 4.Undefined 5.Null 6.Symbol
3
4  &2. 引用类型
5      Object( 包含 Function Array Date )
```

11. null 和undefined的区别?

```
1  null: 空对象
2  undefined: 未定义(变量未赋值的情况下, 默认值是undefined)
3
4  & 两个都代表空值, 所以null == undefined 返回true, 但是数据类型不同null === undefined
    返回false
```

12. 强制类型转换的方式都有哪些?

```
1  &1.转Number
2      parseInt() 转整数
3      parseFloat() 转浮点数
4      Number() 转数值
5
6  &2.转Boolean
7      Boolean() 只有5个值为假：0 '' NaN null undefined
8
9  &3.转字符串
10     String()
```

13. ==和===的区别?

```
1  a == b: 只判断值是否相等
2  a === b: 不仅判断值，还要判断数据类型是否相等
```

14. 遍历数组和对象的方法?

```
1  &1 遍历数组
2      for(let i = 0;i < 数组.length;i++){ ... }
3      数组.forEach(function(v,i){
4          //v数组的每一项    i数组每一项的下标
5      })
6
7  &2 遍历对象
8      for(const key in 对象){
9          //key为对象的每一个属性名，打印对象对应属性名的值用obj[key]
10     }
```

15. 数组的操作方式都有哪些?

```
1  &1.不影响原数组 - 3种
2      1.arr.concat() : 合并多个数组
3      2.arr.slice() : 截取子数组
4      3.arr.includes() : 判断是否含有指定元素
5
6  &2.影响原数组 - 5种
7      1.arr.pop() : 删除最后一个元素
8      2.arr.shift() : 删除第一个元素
9      3.arr.push() : 数组末尾追加元素
10     4.arr.unshift() : 数组之前追加元素
11     5.arr.splice() : 添加 删除 替换元素
```

16. BOM和DOM分别指的是什么?

```
1  BOM: 浏览器对象模型，提供了操控浏览器的一些方式方法及属性
2  DOM: 文档对象模型，提供了操控文档节点的方式方法及属性
```

17. 定时器的类型以及区别?

```
1 setInterval(function(){ ... },毫秒值)    //周期定时器，间隔多少毫秒多次循环执行
2 setTimeout(function(){ ... },毫秒值)    //超时定时器，间隔多少毫秒之后只执行一次
```

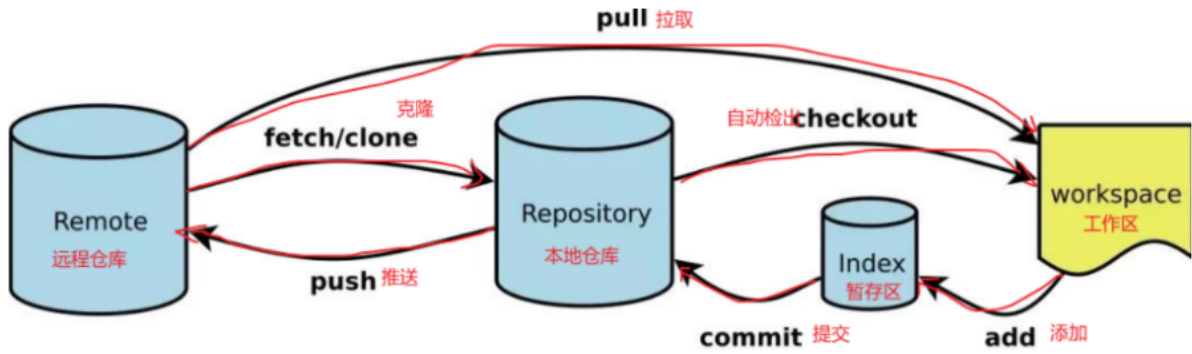
18. 操作元素节点的方式都有哪些?

```
1 &1 元素节点内容(可读可写)
2     元素.innerHTML = 值;    //识别HTML标签
3     元素.textContent = 值;  //纯文字内容
4     表单元素.value = 值;
5
6 &2 元素节点属性(可读可写)
7     元素节点.标准属性 = 值;
8     元素节点.dataset.自定义属性名 = 值;
9
10 &3 元素节点特殊属性
11     元素节点.classList    //类名列表    .add("新类名")    .remove("删除类名")
12     元素节点.style.样式名 = 值
13     元素节点.布尔属性 = 布尔值;    //布尔属性true为启动该属性    false为禁用该属性
14     //常见的布尔值属性: checked disabled readonly selected autoplay controls
```

19. 元素节点添加事件的方式及区别

```
1 &1 标签事件属性
2     <开始标签 on事件名="事件函数(e)">
3
4 &2 绑定事件(0级事件)
5     元素节点.on事件名 = 事件函数(e){ ... }
6
7 &3 监听事件(2级事件)
8     元素节点.addEventListener("事件名",function(e){ ... })
9
10 && 0级事件和2级事件的区别:
11     0级事件:相同的事件只能绑定一次，多次绑定，后面的事件函数会顶替前面的事件函数
12     2级事件:相同的事件可以绑定多次，多次绑定，事件函数会依照顺序依次执行
13 && 阻止标签默认事件:
14     在事件函数的最后return false，仅限于0级事件，因为2级事件允许绑定多个函数，所以无法阻止
```

20. git的工作流程



```
1 #1. 克隆
2   git clone '项目地址'
3 #2. 添加
4   git add .
5 #3. 提交
6   git commit -m "提交信息"
7 #4. 推送
8   git push //提交到当前分支
9   git push origin 分支名 //提交到远程分支上
10 #5. 拉取
11   git pull
12 #6. 查看分支
13   git branch
14 #7. 创建并切换分支
15   git checkout -b 分支名
16 #8. 切换分支
17   git checkout 分支名
18 #9. 合并分支
19   git merge 分支名
```