# CSS

1、Html常见元素：

Head：meta、title、style、link、script、base

Body：div/section/crticle/aside/header/footer

p/

Span/em/strong

Table/thead/tbody/tr/td

Ul/ol/li/dl/dt/dd

A

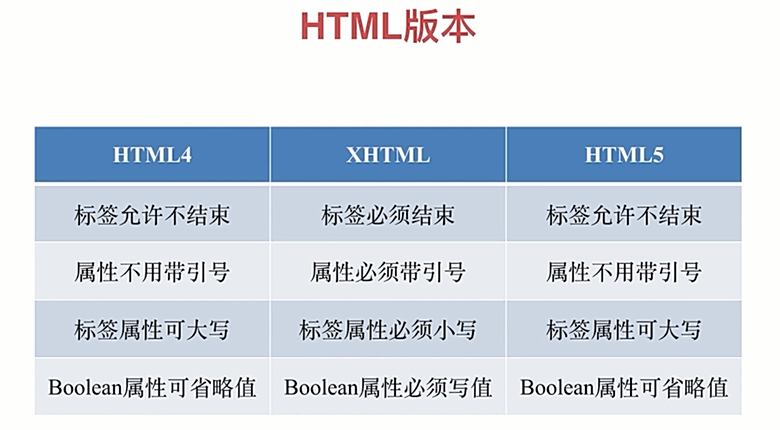
From/input/select/textarea/button

2、<meta charset=”utf-8”> 表示字符集

3、<base href=”/”>表示基础路径

4、<meta name=”viewport” content=”width=device-width,initial-scale=1.0,maximum-scale=1.0, user-scalable=no”>：宽度等于屏幕宽度，适配移动端第一步加上viewport，再调整

1. form里的Input label通过id和for相关联
2. 如何理解Html：html“文档”；描述文档的“结构”；有区块和大纲
3. Html版本

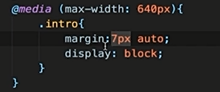


1. html5新增内容：
   1. 新区块标签：section/article/nav/aside
   2. 表单增强：日期、时间、搜索；表单验证；placeholder自动聚焦
2. html5新增语义
   1. Header/footer头尾
   2. Section/article区域
   3. Nav导航
   4. Aside不需要内容
   5. Em/strong强调
   6. I icon
3. Html元素分类
   1. 按默认样式分：块级block、行内inline、inline-block
4. Doctype意义是什么：让浏览器标准模式渲染；让浏览器知道元素的合法性
5. Html、xhtml、html5的关系：html属于sgml；xhtml属于xml，是html进行xml严格化的结果；html5不属于sgml或xml，比xhtml宽松
6. Html5有什么变化：语义化元素；表单增强；新的api（离线、音视频、图形、实时通信、本地存储、设备能力）；分类和嵌套变更
7. Em和i有什么区别：em是语义化标签，表强调；i纯样式，表斜体
8. 语义化的意义是什么：开发者容易理解；机器容易理解结构；有助于seo
9. 哪些元素可以自闭合：
   1. 表单元素input；
   2. 图片img；
   3. br hr；
   4. Meta link
10. html和dom的关系
    1. Html是死的
    2. Dom由html解析而来的，是活的
    3. Js可以维护dom
11. Property和attribute的区别
    1. Attribute是死的
    2. Property是活的
12. Form的作用有哪些
    1. 直接提交表单
    2. 使用submit/reset按钮
    3. 便于浏览器保存表单
    4. 第三方库可以整体提取值
13. 第三方库可以进行表单验证
14. 浏览器的解析CSS方式：从右往左（从性能上考虑，加快css的解析速度）
15. 选择器分类（用于匹配html元素）
    1. 元素选择器a{}
    2. 伪元素选择器 ::before 真实元素，页面可以显示
    3. 类选择器 .link{}
    4. 属性选择器 [type=radio]{}
    5. 伪类选择器 :hover{} 状态
    6. Id选择器 #id{} 唯一
    7. 组合选择器 [type=checkbox]+label{}
    8. 否定选择器 :not(.link){}
    9. 通用选择器 \*{}
16. 选择器权重
    1. Id +100
    2. 类、属性、伪类 +10
    3. 元素、为元素 +1
    4. 其他 +0
    5. ！Important优先级最高
    6. 元素属性 优先级高
    7. 相同权重 后写的生效
17. 非布局样式
    1. 字体、字重、颜色、大小、行高
    2. 背景、边框
    3. 滚动、换行
    4. 粗体、斜体、下划线
18. Css样式的优先级
    1. 计算权重确定
    2. ！Important
    3. 内联样式
    4. 后写的优先级高
19. 雪碧图的作用
    1. 减少http请求数 提高加载性能
    2. 有一些情况下减少图片大小
20. 自定义字体使用场景
    1. 宣传/品牌/banner等固定文案
    2. 字体图标
21. Base64的使用
    1. 用于减少http请求
    2. 适用于小图片
    3. Base64的体积约为原图的4/3
22. 伪元素/伪类区别
    1. 伪类表状态
    2. 伪元素真的有元素
    3. 前者双冒号，后者单冒号
23. 如何美化checkbox
    1. Label[for]和id
    2. 隐藏原生input
    3. ：checked+label
24. 实现两栏（三栏）布局
    1. 表格布局
    2. Float+margin布局
    3. Flexbox布局
    4. Inline-block布局
25. Position：absolute/fixed有什么区别
    1. 前者相对最近的absolute/relative
    2. 后者相对屏幕（viewport）
26. Display：inline-block间隙
    1. 原因字符间距
    2. 解决：消除字符或者消灭间距
27. 如何清除浮动
    1. 原因：浮动不占父元素空间
    2. 让盒子负责自己的布局
    3. Overflow：hidden（auto）
    4. ：：after（clear：both）
28. 如何产生不占空间的边框
    1. Box-shadow
    2. Outline
29. 动画原理
    1. 视觉暂留作用
    2. 画面逐渐变化
    3. 动画的作用；愉悦感；引起注意；反馈；掩饰
30. 响应式布局
    1. 在不同设备正常使用
    2. 一半主要处理屏幕大小问题
    3. 主要方法
       1. 隐藏+折行+自适应空间
       2. Rem/viewport/media query
    4. 案例一 media query（隐藏）



media query（隐藏）

（5）案例二 折行

把inline-block改成block

（6）案例三 自适应

把viewport改成定值

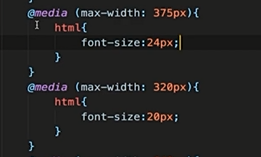


1. 案例四 rem

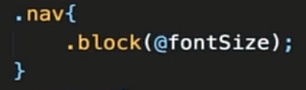
Html{font-size：20px}

自适应改font-size的值，再加几个媒体查询

范围大的放在上面



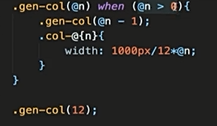
1. 如何实现圆形元素：圆角50%
   1. Transition 过度动画
   2. Keyframes（animation）关键帧动画
2. 过渡动画和关键帧动画区别
   1. 过度动画要有状态变化
   2. 关键帧不需要状态变化
   3. 关键帧动画控制更精细
3. 如何实现逐帧动画
   1. 用关键帧动画定义每一帧
   2. 去掉补间
4. Css动画性能
   1. 性能不坏
   2. 部分优于js
   3. Js可以做到更好
   4. 部分高危
      1. Box-shadow等
5. Less
   1. 嵌套
   2. 变量@fontsize{}
   3. Mixin



* 1. Extend

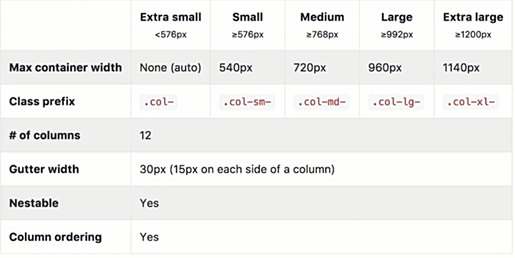


* 1. Loop递归



* 1. Import
     1. 解决文件太细碎加载性能问题
  2. 预处理器框架
     1. Less-Lesshat/est 用nodejs写的
     2. Sass-compass ruby有node版本，c++
     3. 提供现成的mixin
     4. 类似js类库 封装常用功能

1. Bootstrap（tew itter出品）sass写的
   1. Bootstrap 4
      1. 使用flexbox布局
      2. 提供布局和reboot版本
   2. Bootstrap优缺点
      1. 优点：css代码结构合理 现成的样式可以直接用
      2. 缺点：定制较为繁琐 体积大
2. Bootstrap响应式布局



通过mediaquery设置不同分辨率的class

为不同分辨率选择不同的网格class

1. Webpack
   1. Js是整个应用的核心入口
   2. 一切资源均由webpack打包
2. Css-loader 将css变成js

Style-loader 将js样式插入head

ExtractTextPlugin 将css从js提取出来

Css modules 解决css命名冲突问题

1. style-loader