

浏览器

IE Chrome Firefox Safari Opera Yandex

Trident (IE, UC) Gecko (Firefox) Presto (Opera 前内核) Webkit (Safari, Chrome 内核原型) Blink (Chrome, Opera, Yandex)

HTML 结构

node (节点) 页面中的所有内容 (标签, 属性, 文本) (文本包含: 文字, 空格, 换行, 回车) (大段文本的每一个行都是独立节点, 每一个节点都有对应的 **css 属性**)

Element (标签元素) 页面中的所有标签

单标签 ``

双标签 `<p> </p>` (HTML5 中单标签可省略/)

`<!--[if IE]>` `<![endif]-->` `<!--[if IE]>` `<![endif]-->` 支持 IE9 以下版本, 包括 IE9

`<!DOCTYPE html`

`<![ENTITY nbsp " ">]`

`>`

实体

`<`; 小于 `>`; 大于

` `; 空格

`©`; 版权符号

`&laxquo;`; 双向左箭头 `»`; 双向右箭头

`&lasquo;`; `&rasquo;`

` `; 换行符

`​`; 零宽度符号

标签分类

行内元素 inline 一行多个 (无宽高) 有间隙 `<a>` `` `` `<i>` `<s>` `<u>` `` `` `` `<ins>`

1. 容纳 文本 行内 行内块, 不能容纳块级元素 (行内块影响占位, 不会撑开父盒子超过 1em)

2. 行内可设置水平 padding margin

行内块元素 inline-block 一行多个 换行元素有间隙 (去除上下左右间隙: 父级添加 `font-size:0` 或 自己添加 `vertical: baseline;` 但右侧间隙无法去除)

`margin` 会影响一行的行内块 `` `<input />` `<td>` `<textarea>`

块级元素 block 默认为当前页面的宽度 `<h6>` `<div>` `<p>` `<pre>` `` `` `` 【`<p>` 文字块级元素 不能在内存放其它块级元素, 默认 marginpadding `<pre>`】

居中: `position:absolute;` `top = left = right = bottom` (值相等) , `margin: auto;`

`position:absolute;` `top= 50%;` `left = 50%;` `transform: translate(-50%, -50%);`

置换元素 内容会被替换 `` `<input>` `<textarea>` `<select>` `<object>`

不可替换元素 html 大多数元素都是不可替换的, 内容直接显示 `<label>` (属性和值相同时 可以只写这个属性不加值) (父盒子能容忍首个子盒子溢出 其他将被挤开)

排列

低层快速选择: 块级: <h1> <p> <pre> <div> 行内: <i> <s> <u> <ins>

代替 class 便于加属性

测试颜色 添加背景色查看各个盒子排列情况

1. float 横向排列 (浮动排列经常计算宽度, 超出宽度会在下一行, 浮动盒子无法设置百分比宽高)
2. flex 横向排列
3. inline-block 行内块横向排列 (一般用于一行文字排列, 去除右侧间隙用 word-spacing 不稳定, 设置 margin 会整行偏移)
4. block 块级纵向排列 【宽度为 50% 百分比时, 是相对父级的 content 宽度的 (没有 padding 和 margin)】
5. position 完全相对父盒子排列, 可以重叠, 定位排列

父级给定高度时, 子盒子注释标明每个 组合高度

最上层的标签一定要给 class, 只有最底层的标签直接用标签选取 (CSS 要分层次, 不要越级, 否则不便于后期阅读)

盒子内部盒子太多, 不应给定高度, 慢慢被撑开便于维护。

请求

超文本引入 <link href="name.css" rel="stylesheet" type="text/css" /> CSS 文件
name.css 这类请求浏览器会自动加上 index 请求的根路径 <http://www.opru.com:3000/lala/dd/index.html>
→ <http://www.opru.com:3000/lala/dd/name.css>
/name.css 这类请求浏览器直接到根路径 <http://www.opru.com:3000/lala/dd/index.html> →
<http://www.opru.com:3000/name.css>

<link href="favicon.ico" rel="shortcut icon" type="image/x-icon"/> 图标文件 (32*32)
 链接
引用 <script src="name.js" type="text/javascript" > JS 文件
 图片

BFC&IFC

FC 的含义就是 Formatting Context 指页面中的一块渲染区域, 并且有一套渲染规则, 它决定了其子元素将如何定位, 以及和其他元素的关系和相互作用 (CSS2.1 规范中的一个概念)

BFC 和 IFC 都是常见的 FC。分别叫做 Block Formatting Context 和 Inline Formatting Context 【BFCmargin 垂直方向会合并】

HTML 语义化

去掉或者丢失样式的时候能够让页面呈现出清晰的结构

方便其他设备解析 (如屏幕阅读器、盲人阅读器、移动设备) 以意义的方式来渲染网页;

有利于 SEO: 和搜索引擎建立良好沟通, 有助于爬虫抓取更多的有效信息: 爬虫依赖于标签来确定上下文和各个关键字的权重;

便于团队开发和维护, 语义化使得网页更具可读性, 是进一步开发网页的必要步骤, 遵循 W3C 标准的团队都遵循这个标准, 可以减少差异化。

SEO

Search Engine Optimization 搜索引擎优化 主要有站内优化 和 站外优化 (现阶段站内优化)

- <title> 1. 搜索引擎了解网页的入口 一般大小 Google(70kb)35 个中文 Baidu(56kb)28 个中文
2. 最先出现的词语权重越高 主关键词出现 3 次 辅关键词 1 次
3. 一般标题 网站名-介绍

<meta name="description" /> 网页简短介绍 尽量不要超过 120 字

<meta name="keywords" /> 关键字 一般 6-8 个关键字左右

<html>

标签

<!DOCTYPE html> HTML5 标准声明, PUBLIC 网址为规范解析的规则 【标签等号后的内容没有引号时会自动加引号 <div test={}-----> === <div test="{ }-----" >】
<!-- --!> 注释

<html> 根标签（祖先标签 后代标签）
lang="en"英文 **"zh-CN"** 中文 **"zh-tw"**繁体 网页语言声明——对内容没有影响（浏览器会提示是否翻译）

<head>

标签

<title> 网页标题，影响排名

<meta /> **charset="utf-8"** 选择解码字符集

name="keywords" 网站关键字 搜索引擎会检索关键字和描述 但不会影响在搜索引擎中的排名

"description" 被百度检索到时标题下方显示的网站简短介绍

name="viewport" 移动端视口设置

content="height" = 3px 或 device-height, 手机默认高度 **width = 3px 或 device-width,** 手机默认

宽度

initial-scale = 1.0, 初始缩放比例（最大为 10）

minimum-scale = 1.0, 最小缩放比例

maximum-scale = 1.0, 最大缩放比例

user-scalable = no 或 yes, 用户能否缩放（否的画 max 和 min 缩放比例会被忽略）

target-densitydpi = 30dpi | device-dpi | high-dpi | medium-dpi | low-dpi, 像素密度（每英寸所拥

有的像素数量）

content="keywordsContents...." 描述前面关联的信息

http-equiv="refresh" content="5;url=http://www.k.com" 停留 5 秒钟跳转到指定网页（**content="5"** 每 5 秒钟

刷新当前网页）

http-equiv="content-type" content="text/html; charset=UTF-8"

设置请求头

http-equiv="X-UA-Compatible" content="IE7"

IE8 会识别，用 IE7 进行渲染

http-equiv="expires" content="Fri,12Jan200118:18:18GMT"

一旦网页过期，必须到服务器上重新

传输。

<base /> a 链接跳转方式 **target="_blank"** 整体链接用新窗口打开（可被覆盖） **_parent** **_self** **_top**

frame 名称【锚点】

<link /> 引入文档 **rel="next"** 声明与当前文档的联系（去掉 **rel** 不能发送请求）

<style> **div { height: 30px; }** **</style>** CSS 样式

<body>

文字

<blockquote>

文本引用，一般嵌套 p 元素，用于标记被引用的文本，这些引用文本并不是直接位于 blockquote 元素中。

<h1> ~ <h6>

h1 最重要，表示网页中的主要内容，其它重要性依次递减 中间为一级标题内容

h1 重要性仅次于 title 会影响网页在搜索引擎中的排名 只写一个 h1

<p>

标签独占一段 与其他内容间距稍大 【默认有 margin，内部不能有 div】

<pre>

显示原始文本结构（保留空格和换行符） 【默认有 font-family white-space】

换行标签

<hr />

水平线

<a>

`href="http://www.baidu.com"` 地址 `"#"` 空链接 (`href="//www.taobao.com"` 表示忽略 http 协议
`href="mailto:a@c.com"` 发送邮件到 a@c.com)
锚点 `href="dong.html#lalaal"` 跳去制定 id 的盒子位置 (id="lalaal") ,可前加文件名
什么都不做 `"javascript;"` `"javascript:void(0);"` 【`text-decoration` `none` 去掉下划线】
`target="_self"` 跳转到当前窗口 (默认)
 `"_blank"` 新窗口【锚点不能有 target 属性】
 `_parent` `_top` `framename`
<a>为行内元素, 内放 img 时不会被撑开, 显示为一个 img 内部底边上的行内小盒子 。【<a>内部不能再嵌套 a 标签】
<a>有默认 color 字体颜色值, 需要设置 `color: inherit;`
 点击 a 里的内容跳转, 搭配<base /> 指定全局 target 属性

<center>

center 标签中的内容会默认居中显示

<div>

标准块级元素

标准行内元素

字体加粗 strong 有强调的语义 【默认有 font-weight】

<i>

字体倾斜 em 有强 调的语义 【默认有 font-style】

<s>

删除线 【默认有 text-decoration】

<u> <ins>

下划线 【默认有 text-decoration】

<sup> Exponent

无序列表: 嵌套使用 (li 可以单独使用, 但不符合规范)

有序列表 嵌套 (用 12345 排序)

<dl>

自定义列表 嵌套<dt>小标题</dt> <dd>描述</dd>

注意: a. 全为块级元素 dl 默认 marthb dt 无 dd 默认 marl

格式

 引入外部图片 如 jpg,jpeg,png,gif 格式 不能引入 psd img 【@2x.png @3x.png 适应不同的 dpi 设备---设备像素比 dpi iphone6=2 iphone6plus=3】

`src="1.jpg"` 路径 (base64 编码随整个页面加载同时只请求一次, 此格式适用于小图片) 【svg 矢量图片--可以在 iconMoon 制作成字体图标】

`alt="lala":` 图片不能显示时的描述 (搜索引擎可以通过 alt 识别不同的图片, 不写 alt 时搜索引擎不对 img 中的图片进

行收录)

title="标题" 鼠标在图片上显示内容
width="100" 缩小后图片占内存不变
height="300" 一般省略 px
注意: a. 清除老版本浏览器图片蓝边 border: 0;
b. 宽度高度属性只设置一个时, 会等比例放大缩小 (有时出错 CSS 中给宽高 auto 随另一边保持比例变化)

<iframe> 内联盒子 src 属性可以用作插入视频链接, 引入其他网页 (常用)

<table>

width="400" 表格宽度
height="400" 高度
border="1" 默认为 0 像素 大小盒子边框 (小盒子边框最大为 1)
cellspacing="2" 默认为 2 像素 (单元格间距离)
cellpadding="2" 默认为 1 像素 (单元格内距离)
align="center" "left" "right" 设置表格水平对齐方式 对<tr><td>可用
valign="top" 竖直对其方式 对<tr><td>可用

<caption>表格标题</caption>
<thead><tbody>表示头身部分
 <tr> 一行</tr>
 <th>内容加粗居中单元格</th>
 <td>单元格</td>
td th 属性: colspan="2"跨列合并
 rowspan="2"跨行合并 (先上下后 先左后右 多余格删除)
 scope = "col" 列的表头 "row" 行的表头 "colgroup" 列组的表头 "rowgroup" 行组的表头

【语义属性】

TIPS: table 不能放除 tr th td 之外的其他标签

| 表单 | | | | |
|--|-------------|---------------------------------|---------------|---|
| <form> | 表单域 | action="/demo.php" | 提交的目标文件 | 【通过 action= \$_SERVER['PHP_SELF'] 设置为当前路径, 便于维护】 |
| | | method = "get" "post" | | 【get: URL 提交 (不能传文件, 能看到提交信息不安全) post: 请求体提交】 |
| | | enctype = 'multipart/form-data' | 默认 urlencoded | (普通数据 Form-Data 文件 Request Payload) 【上传文件 file 必须使用, 二进制】 |
| | | novalidate | 取消浏览器自动校验 | |
| | | autocomplete="off" | 关闭客户端自动完成功能 | |
| <input /> | type="text" | 文本输入 | "password" | 密码输入 【name 属性为下标, 内容为值】 (没填为空字符串) |
| | "radio" | 单选按钮 | "checkbox" | 复选框 【name 属性为下标, value 为值】 (没选中为空, 多选用 name="arr[]" 索引数组保存) |
| | "button" | 普通按钮 | "reset" | 清除表单内容按钮 |
| | "file" | 选择文件 | 无默认值 | 【name 属性为下标, 文件信息数组为值】 (多选用 name="file[]" 索引数组保存) |
| 兄弟选择美化默认组件 file radio checkbox <label> | | | | |
| <input 美化目标> <i> </i> </label> | | | | |
| 判断是否选中文件 this.files.length | | | | |
| \$(this).prop("files").length | | | | |

"submit" 提交按钮 **"image"** 提交图片按钮 【可用 return false 阻止默认提交跳转】

(默认表单提交为**同步请求**, 客户端会一直等待服务端响应, 并自动将响应体渲染覆盖到当前页面-----返回经过模板渲染的页面-在预先的空盒子里显示错误提示和返回发送的用来判断的用户数据)

(表单注册提交事件发送**异步请求**, 客户端可同时做其他事但服务端对异步请求不能重定向, 并在 ajax 请求里接受响应体----根据返回信息操控 dom 实现各种效果, 比同步请求显示单一信息更灵活)

"hidden" 隐藏-用于储存要提交的数据

| | |
|---------------------------|---|
| value="" | 默认值 |
| name="" | 分组名-同组只能有一个选中 |
| src="" | 图像按钮路径 (配合 type=image 用) |
| checked | 初始选中 (默认初始不选中) |
| maxlength="1" | 允许输入的最多字符数 |
| accept="image/gif" | 限制上传文件类型 // "image/*" 所有类型图片 "audio/*" 所有类型音频 ".lrc,.mpc" 指定多个扩展名 |
| autocomplete="off" | 控制浏览器自动完成功能 【off 禁止浏览器自动填充 new-password Chrome 有效, Firefox 无效】 |

TPIS: 美化上传图片: <label> <input type="file" /> <label> 隐藏上传文件, 对 label 设置样式

<label> 点击任意内部包裹标签 跳转或选中到 内部 input **for="id"** 点击 label 光标到 id 位置

<button> **disabled="disabled"** 禁用

type = "button" "submit"(提交按钮) "reset"(清除表单内容按钮) 【不同浏览器有不同默认值】

<select> 下拉选择列表

<option> 选项 **value="-1"** 默认值 **selected** 默认选中 【name 属性为下标, 默认 value 为值, 值可以为内容】

<textarea> 文本域 中间为默认内容

cols="30" 列数

rows="10" 行数

disabled 禁用

readonly 只读不能修改

name="" 分组名

required 必填项

H5

语义

概念: 语义化 div, IE<=8 不适用

<header> 页眉

<section> 文档中的节

<footer> 页脚

<nav> 导航

<article> 文章

<main> 文档主要内容

<aside> 侧边 【IE9 将<main>解析为行级元素 - 添加 display=block;】 【IE<=8 兼容 H5 -- 引入 html5shiv.min.js】

表单

<input /> **type= "email"** 邮箱 (默认验证 含@ xx.com) **"url"** 网址 (默认验证 含 http://) **"tel"** 手机号码 (移动端打开数字键盘 只能输入数字)

"number" 数字 (只能输入数字小数点) **"search"** 搜索 (右边含有清除全部符号 X, 移动端虚拟键盘 enter 变搜索)

"range" 左右移动滑块

"color" 颜色拾取器

"time" 时间 - 01:10

"date" 日期 - 年/月/日 (含日历选择控件)

"datetime-local" 日期时间 年/月/日 --:-- (含日历选择控件)

"month" 月份 - 2020 年 02 月

"week" 星期 - 2020 年第 3 周

| | |
|-------------------------------------|--|
| placeholder ="" | 提示文本（输入时消失） |
| autofocus | 自动获取光标 |
| multiple | 可多选文件/邮箱（a@a.com, b@b.com）【"file" "email"】 |
| autocomplete ="on" "off" | 自动提示 - 输入框下（必须成功提交过&&含 name 属性） |
| required | 必填项 - 不能为空 |
| pattern ="^\(+86)?1\d{10}\$" | 正则验证【"tel"】 |
| form ="formid" | 域外的元素添加指定 id 的表单域 |
| accesskey ="s" | 规定激活某个元素的快捷键 alt+...的形式 |

accept=".doc,.docx,application/msword,application/vnd.openxmlformats-officedocument.wordprocessingml.document"
接收文件类型

| | |
|------------------------------------|---|
| max ="10" min ="0" | 最大值最小值 【"number" "range"】 |
| aria-label | 阅读器获取焦点后读出内容 role="searchbox" 阅读器识别名称 |
| #afa:-webkit-input-placeholder { } | /* WebKit browsers */ 设置 placeholder 样式 |
| #afa:-moz-placeholder { } | /* Mozilla Firefox 4 to 18 */ |
| #afa:-moz-placeholder { } | /* Mozilla Firefox 19+ */ |
| #afa:-ms-input-placeholder { } | /* Internet Explorer 10+ */ |

去除 number 输入箭头 input::-webkit-outer-spin-button, input::-webkit-inner-spin-button
{ -webkit-appearance: none; }
input[type='number'] { -moz-appearance: textfield; }

<fieldset> 表单边框 **<legend>** 表单标题

<kengen> **name**="security" 表单公钥加密，当提交表单时，私钥存储在本地，公钥发送到服务器。

<output> 表单中执行计算然后在 **<output>** 元素中显示结果（不能修改）

// **<input type="range" id="a" value="50">100 + <input type="number" id="b" value="50">=<output
name="x" for="a b"></output>**

<datalist> 输入框添加 下拉选择列表

<input/> **list**="d" 添加 list 导入列表 id
<datalist id="d"> **<option>** **value**="" 默认值（不通过默认验证不会显示） **label**="" 提示内容（默认值后）【firfox
不支持 value】

音频

<progress> 进度条 **max**="10" **min**="0" 最大值最小值 **value**="" 默认值

<meter> 度量器 **max**="10" **min**="0" 最大值最小值 **low** 低分隔线 **high** 高分隔线 **value**="" 默认值

<audio> 播放音频 支持三种音频格式 Ogg MP3 Wav

IE9 (MP3) Chrome3.0 (Ogg MP3) Firefox3.5 (Ogg Wav) Opera10.5 (Ogg Wav)

width height 宽度高度（只需设置一个）

src="" 引用视频文件

autoplay 自动播放

controls 显示播放控件

loop 无限循环播放（默认播放一次）

<source/> **src**="" **type**="MIME" 从上到下自动选择支持的文件（都不支持显示标签内最后一行文本）

<video> 播放视频 支持三种视频格式 Ogg (F3.5+ O10.5+ C5.0+) MPEG 4 (IE9.0+ C5.0+) WebM (F4.0+ O10.6+ C6.0+)

width height 宽度高度（只需设置一个）

src="" 引用视频文件

autoplay 自动播放

controls 显示播放控件
loop 无限循环播放（默认播放一次）
poster="path" 视频播放前的封面
<source src=" " /> 从上到下选择支持的文件 都不支持最后行可写上版本太低

<embed> 直接插入多媒体文件（调用本机安装的软件播放 - 没有则不能播放）

<canvas> **width height** 宽度高度（不能直接对 canvas 设置样式，可以在外面加父盒子设置样式）

HtmlAndCss / CSS

概念

引入 CSS

行内样式表 <div style="color:pink; font-size:25px;"> 内容</div> 越左侧优先级越高

内部样式表 <head> <style> div { height: 30px; } </style>

外部样式表 <head> <link rel="stylesheet" href="路径" type="text/css" /> (type 一般可省略)

当前 css 文件中引入 @import url("layout.css") [screen]; 或@import "layout.css" (可指定媒体生效)

声明 css 文件在页面的编码 @charset "UTF-8";

ico 文件

根目录下命名为 favicon.ico <link rel="icon" href="favicon.ico" type="image/x-icon"/> (type 一般可省略)

css 变量

body{ --penguin-skin: gray; } body{ background: var(--penguin-skin, black); } 不支持旧的 css 定义使用 black 颜色

:root { --penguin-skin: gray; } 全局生效

选择器

标签 **div** {color:#f0f;} (直接选出标签 效率: div > div.lala) 【5 类

选择器, 5 类连写符, 3 类特殊选择器】

类 **.lala** {color:#f0f;} <div class="lala"></div> (选出使用类的标签)

伪类 **div:link** (未访问) **div:visited** (已访问) **div:hover** (鼠标移到当前位置) **div:active** (鼠标按下当前位置)

连写顺序 link visited hover active (可选子代 div:hover span {}) 【IE6 不支持】

id **#fafa** {color:#f0f;} <div id="fafa"></div> (选出指定 id 的标签)

***** {color:#f0f;} 全部标签 【#name > * name 下的所有标签】

直属后代 **div > p** {color:#f0f;} 所有后代 **.lala div p** {color:#f0f;}

交集选择 **div.lala** {color:#f0f;} **#one.green .pink.green** [选择器][选择器] (不能合并到类里)

并集选择 **div, p, .lala** {color:#f0f;} (逗号优先级最低) **//.search i, s .search** 里的 i 和所有的 s

后面多个兄弟元素 **div ~ .lala** {color:#f0f;} **div ~ .lala** {color:#f0f;}

后面一个兄弟元素 **div + .lala** {color:#f0f;}

属性选择器 (属性值可不带"")

div [class] 带有 class 属性

div [class = demo] 属性 等于 demo (双引号加不加都可以)

div [type ^= demo] 属性 以 demo 开头

div [width \$= demo] 属性 以 demo 结尾

div [src *= demo] 属性 包含 demo

索引选择器 (给子标签添加, empty 和 target 除外)

:nth-child(n) 从上到下 父级第 n 个元素 (**n 2n 2n+1 4n -n+5**)(<=0 时选取无效 **even** 偶数 **odd** 奇数 这里的索引从 0 开始)

:nth-last-child(n) 从下到上 父级第 n 个元素
:first-child 此类型父级第一个元素 (此类型不是第一个找不到标签)
:last-child 此类型父级最后一个元素
:nth-of-type(n) 从上到下 父级第 n 个此类型元素 (限制类型必须为当前目标类型)
:nth-last-of-type(n) 从下到上 父级第 n 个此类型元素 (限制类型必须为当前目标类型)
:not(p) 非 p 标签的元素
:first-of-type 父级第一个此类型元素
:last-of-type 父级最后一个此类型元素
:empty 标签内没有任何内容 - 包括空格
:target 有链接跳到当前标签锚点时, 触发伪类样式 // <div
id="kk"></div> --> #kk:target{ width:100px; }
:global(.ss) 全局规则, 不会被转换成哈希字符串
:root 整个 HTML 文档

伪元素选择器

::before 行内块元素 插入到当前盒子内部最前方 (必须有 content: "\e91b" - 可加字体图标) 【低版本浏览器不识别::before ::after 双冒号 高版本单冒号自动识别为双冒号】
::after 行内块元素 插入到当前盒子内部最后方 (必须有 content: "\e91b" - 可加字体图标) 【IE67 不支持伪类 IE8 支持单冒号】 【用于封装---各个盒子里的公共内容封装成类.arrow】
::first-letter 文本标签中 选出第一个字 (给文字浮动时会实现文本环绕)
::first-line 文本标签中 选出第一行
::placeholder 设置 placeholder 样式
::selection 文本标签中 改变文本被选中的状态

标签特性

层叠性 最后的样式会覆盖之前的 (样式冲突时执行优先级大的或最后添加的)
继承性 四种继承样式 color: text- font- line-
优先级 div!important 无穷大 行内样式 1,0,0,0 ID 0,1,0,0 类伪类 0,0,1,0 标签 0,0,0,1 * 继承 0,0,0,0
【color: #000!important;】

字体

TureType (.ttf) Windows 和 Mac 最常见的字体 一种 RAW 格式
OpenType (.otf) 一种原始的文字格式 其内置在 TureType 的基础上
Web Open Font Format (.woff) Web 字体中最佳格式
Embedded Open Type (.eot) IE 专用字体
SVG (.svg) 基于 SVG 字体渲染的一种格式

导入字体图标

1. 导入 fonts 解压文件夹 (浏览器会自动获取里面适合的格式 .eot 等, 生成字体只适用于生成的那几个字)

```
@font-face {
    font-family: "icomoon"; /*声明字体名称*/
    src: url('fonts/icomoon.emot?7kkyc2'); /*路径必须绝对路径, 不能相对路径*/
    src: url('fonts/icomoon.eot?7kkyc2#iefix') format('embedded-opentype'), /*引入字体文件--约束一段字符代码为什么图案*/
    url('fonts/icomoon.ttf?7kkyc2') format('truetype'),
    url('fonts/icomoon.woff?7kkyc2') format('woff'),
    url('fonts/icomoon.svg?7kkyc2#icomoon') format('svg');
    font-style: normal; /*倾斜字体变正常*/
}
```

2. 字体使用 font-family: "icomoon";

字体图标使用 预定义样式-添加样式 .smile::before{ content: "\e641" } 或 复制字体图标小方盒子 或 复制字体图标实体制作 svg 图片-字体图标导入 1. icomoon 中 Import Icons 选中文件 2. 选中下载
追加图标 1. 打开要添加的 svg 文件 2. Import Icons 添加主文件中的 json 文件 3. 下载 (引入新图标时要重新覆盖 fonts 文件)

阿里图标使用 导入 font 文件夹 css 文件中声明字体类  【导入新文件夹到 node, 需要重启 vscode 才会生效】

浏览器前缀

-webkit-	(Chrome Safari Android Browser)
-moz-	(Firefox)
-o-	(Opera)
-ms-	(IE)
-khtml-	(Konqueror)

样式 @keyframes xxx {} transition-xxx: animation-xxx: border-radius: box-shadow: backface-visibility: column: flex: perspective:

js 方法 dom.mozRequestFullScreen();

js 成员 document.mozFullScreenElement

位置样式

盒子

width: calc(100% + 1px);

box-sizing 改变盒子模型的组成方式 **border-box** 长宽为 content+padding+border (IE 盒子盒模型, padding 不改变整体宽度) 【子盒子 width: 100%时, 为当前盒子 width - padding】

content-box 长宽为 content (标准盒模型, 去除 body 底部空白 overflow-y: hidden;) 【子盒子 width: 100%时, 为当前盒子 width-----当前盒子有 padding 时会溢出】

padding padding: 10px; 上下左右内边距, 会覆盖 padding-top (可对行内设置水平方向)
padding: 0 10px; 上下 0 左右 18
padding 10px 20px 30px; 上 10 左右 20 下 30
padding 10px 20px 30px 40px; 上 10 右 20 下 30 左 40

padding-top padding-bottom padding-left padding-right

margin margin: 10px; 上下左右外边距 (可对行内设置水平方向, 父子间无内边距和边框时垂直外边距会合并) 【盒子伪装成间距, 封装成类 .divider 】

margin: 0 10px; 上下 0 左右 18px margin: 0 auto; 水平居中 (auto 只对块级水平方向有效, 浮动定位无效)

margin 10px 20px 30px; 上 10 左右 20 下 30
margin 10px 20px 30px 40px; 上 10 右 20 下 30 左 40

margin-top margin-bottom margin-left margin-right margin-top xx%由于某种原因取决于页面的宽度, 而不是高度

文字

line-height 1. 从文本基线上下延伸的距离 (配合内部文本使用, 对浮动盒子不起效) 【25px 1.5 (font-size 的 1.5 倍) 】
2. 能撑开父盒子

vertical-align (文本 行内 行内块) 基线对准的位置 baseline (默认基线对齐) bottom (底线对齐) middle (中线对齐) top (顶线对齐) 【基线以最高的盒子为准, 其他一行的盒子会被吊起来】

1. 用对齐线代表行内块/行内/文字 (默认与离顶线最远的对齐线对齐, top 直接与顶线对齐)

2. 行内块 `baseline` `bottom` 均为占位下边缘【`baseline` 有一个默认的小行高】

可以添加一个 100%高的伪元素或直接在内部添加 100%的盒子实现居中

`overflow-wrap` `normal` `break-word` 文字换行

`column-count`: 列个数 3 (为块级元素添加, 内部所有元素都统一显示--通常用来显示文本)

`column-width`: 列宽度 50px (默认宽度较大时优先)

`column-gap`: 列间隙宽度 50px

`column-rule`: 列边框样式 dashed 3px red

`column-span`: 内部子元素应横跨多少列 1 指定跨 1 列 all 跨所有列 (内部元素添加样式)

位置

`display: flex`; 父级内伸缩布局 - 所有子元素 float 【子元素总宽度超出父元素时, 子元素平均收缩-不换行】【需要横向嵌套盒子同比例缩放时, 横向应使用百分比, 不用 flex】【子元素 block 会默认 flex:0 消失】

`justify-content`: 父级内主轴上的排列顺序 `flex-start` 子元素靠左 `flex-end` 子元素靠右 `center` 子元素在中间

`space-around` 子元素两边产生适应间距 `space-between` 左右两端对齐, 中间产生适应间距

`align-items` 父级内侧轴上的排列顺序 `stretch` 默认拉伸填充整个侧轴 (定义高度不填充) `flex-start` 子元素靠上

`flex-end` 子元素靠下 `center` 子元素在中间 `baseline` 根据文本基线对齐

【设置了 align-items 后, 内部子元素再给 margin-bottom 会让盒子上升】

`flex-flow` 父级内主轴和换行方式 `row` `wrap` (flex-direction flex-wrap)

`flex-direction` 父级内主轴方向 `row` 默认水平方向 `row-reverse` 水平翻转方向 `column` 垂直方向 `column-reverse` 垂直翻转方向

`flex-wrap` 父级内换行方式 `nowrap` 不换行-平均收缩 `wrap` 换行 `wrap-reverse` 上下翻转排列

`align-self` 当前元素侧轴排列位置 `stretch` 默认拉伸填充整个侧轴-定义高度不填充 `flex-start` 子元素靠上 `flex-end` 子元素靠下 `center` 子元素在中间 `baseline` 根据文本基线对齐

`flex` 当前元素分配剩余空间 默认 0 1 auto (flex-grow [flex-shrink] [flex-basis]) 【子盒子未占满父级时, 按照 flex 比例分配剩余空间】

`flex-grow` 定义放大比例--分配剩余空间 默认 0

`flex-shrink` 定义收缩比例--显示不下时, 分配超出空间 默认 1

`flex-basis` 主轴方向上的初始长度 默认 auto 20px

`order: 1`; 多个 flex 子盒子配列顺序, 从 1 开始排列

`min-width` 盒子伸缩最小宽度 (移动端全用百分比) 400px

`max-width` 盒子伸缩最大宽度 (移动端全用百分比) 300px

`float` `left` `right` `none` 不占位置, 给需要排列的子盒子添加 (默认转换为行内块, 影响文字占位-文字环绕, 能撑开浮动和定位盒子, 左靠齐 margin 负值会压住边缘, 文字会撑开不用给子盒子宽度)

浮动超出宽度必换行 (可以给一个很宽的父盒子再给上层父级 overflowhidden) 【display:flex 不会换行】

脱离文档流==浮起来不占位置 (relative 不会脱离文档流)

清除浮动 (子盒子高度超出父级盒子时, 会以父盒子高度为准 父盒子无高度时, 会被子盒子撑开)

1. 父盒子添加 (clearfix 会让浮动子盒子溢出并且下方会出现文字环绕) 【需要子盒子溢出时用 clearfix, 不需要子盒子溢出用 overflow】

```
.clearfix:after { content: ""; display: block; visibility: hidden; height: 0;
```

```
line-height: 0; clear: both; }
```

```
.clearfix { *zoom: 1; }
```

* ie7 及以下的版本执行 其它浏览器不执行

2. 父盒子添加

```
.clearfix:before, .clearfix:after {content: ""; display: table;}
```

```
.clearfix:after { clear: both;}
```

```
.clearfix { *zoom: 1;}
```

3. 父级最后的子元素添加样式 clear: both (清除左右浮动) left (清除左浮动) right (清除右浮动)

4. 父级添加样式 overflow: hidden; (清除子盒子溢出 并且有清除浮动的撑开效果)

position: static 默认静态定位 (用来取消定位, 定位在浮动之上) 【同时给 float 和 relative 时, relative 浮起来时不占位置, 但会被后方的 float 压住】

relative 相对定位-原位置 移动, 且原位置的 padding, margin 有效 (占有原位置, 只是在原来的空壳盒子上浮起来了)

absolute 绝对定位-最近的 相对定位 relative 父级 移动, 子元素要想显示在底部, 必须设置 -1 (不占位置, 左右冲突时不会执行 right: bottom:)

fixed 固定定位-以当前屏幕基准 (不占位置 - 只有 float absolute fixed 自动行内块转换)

absolute 和 fixed 宽高 100% 不会生效 添加外部 position: absolute, padding: xxx width: 100% 限制盒子 再在内部添加 width: 100% (这个宽度由外部决定)

left: top: right: bottom: (要初始化默认值 left: 0; top: 0) 盒子定位到中心 top: 50%; margin-top: 高一半 或 transform: translate(50%)

参照父元素中心偏移 left: 50% top: 50% transform: translate(0, -50%)

z-index: 999 叠放次序 (层级默认为 0 无单位) 只有定位的盒子才有 z-index (默认同级后来标签居上)

外观样式

盒子

display: none 隐藏 inline 转行内元素 block 转块级元素-显示 inline-block 转行内块元素 【position 和 float 没有产生盒子, display: none 无效】

table 父元素添加, 显示为块级表格, 前后带有换行符 inline-table 此元素会作为内联表格来显示 (类似 <table>), 表格前后没有换行符。

table-cell 此元素会作为一个表格单元格显示 (类似 <td> 和 <th>) table-row-group 此元素会作为一个或多个行的分组来显示 (类似 <tbody>)。

table-header-group 此元素会作为一个或多个行的分组来显示 (类似 <thead>) table-footer-group 此元素会作为一个或多个行的分组来显示 (类似 <tfoot>)。

table-row 此元素会作为一个表格行显示 (类似 <tr>)。 table-column-group 此元素会作为一个或多个列的分组来显示 (类似 <colgroup>)。

table-column 此元素会作为一个单元格列显示 (类似 <col>) table-caption 此元素会作为一个表格标题显示 (类似 <caption>)

appearance: 元素外观 normal 正常 icon 小图片 button 按钮 none 消除默认外观 【-webkit- -moz-】

touch-action: pan-y; 移动端在指定盒子内, 用户能够怎样滑动

background: color image repeat attachment position/size; (可多个, 随意缺值) 【position 和 size 不能同时都为数字值, 错误写法: 0 0/100% 100%】

background-color #f0f (默认完全透明 transparent)

background-image url(images/g.gif); url("images/g.gif"); linear-gradient(to right/45deg, pink, blue) 渐变会遮挡父级背景

background-repeat repeat 默认横纵平铺 no-repeat 不平铺 repeat-x 横向平铺 repeat-y 纵向平铺 round 缩放适应大小再平铺

background-attachment scroll 默认滚动背景-随父盒子一起移动 fixed 固定背景-父盒子动, 背景不动

background-position 1. 50px -12px; 或 50% 0% 水平偏移 垂直偏移 (省略偏移默认 center 居中)

2. x(left right center) y(top bottom center)

background: linear-gradient(35deg, red 0%, green 50%, blue 100%); **repeating-linear-gradient(35deg, red 0%, green 50%, blue 100%);** 目标位置, 起始结束颜色 (to right top 右上角)

background: radial-gradient([ellipse/circle] [closest-side/farthest-side/closest-corner/farthest-corner] [at 50px 50px] red, blue) 形状, 最远近边/角, 位置, 起始结束颜色

background: repeating-linear-gradient() 重复线性渐变

background: repeating-radial-gradient() 重复放射渐变

background-size: 20px 30px; 背景宽高 (省略默认 auto)

40% 40%; 相对父级大小

contain 宽度占满容器

cover 高度占满容器 (可配合设置水平居中)

background-origin: border-box/padding-box/content-box; 背景图片坐标原点位置 - 从 边框/padding/内容 开始填充 (默认 padding-box)

background-clip: border-box/padding-box/content-box; 背景图片裁切 - 只显示 边框/padding/内容 里面的内容 (默认 padding-box)

border: 1px solid #ccc; none 去除边框 (border: 0 none;) 【用:after 伪装成边框再封装成类 .bdr::after】

border-top: border-bottom border-left border-right

border-width: 1px

border-style: none 无边框 **solid** 实线边框 **dashed** 虚线 **dotted** 点线 【三角形: border: 100px; border-style: solid dashed dashed dashed; border-color: blue transparent transparent transparent;】

border-color:

border-radius: 2px 或 50% 边框的边角圆的半径 (表单框有投影可以 border-style 设置为 none 再重新设置边框)

border-radius: 1px 1px 1px 1px; 左上 右上 右下 左下

border-radius: 10px 0; 左上右下 右上左下

border-radius: 1px 1px 1px 1px/1px 1px 1px 1px; (水平方向和垂直方向) 左上 右上 右下 左下

border-bottom-right-radius: 2px; 指定特定边的 radius

border-collapse: separate 表格边框分离 **collapse** 表格边框合并 **inherit** 从父级继承 border-collapse 的值 (需要给 td 加上此样式)

border-spacing: 10px 50px 表格相邻单元格的边框间的 水平和垂直距离

border-image: source slice/width/outset repeat // url() 27/27px/0px round

border-image-source: url("") 边框图片 (原有边框占有位置但是变为完全透明)

border-image-slice: 27 设置四个方向上的裁切距离 **fill** 做内容的内部填充;

border-image-width: 20px 边框图片的宽度 (边框图片本质时背景, 不影响内容的位置)

border-image-repeat: stretch 默认拉伸 (slice 区域会被保护) **repeat** 横纵平铺 **round** 缩放适应大小再平铺

border-image-outset: 0px 扩展边框

object-fit: cover; **fill** 默认, 拉伸填满 **contain** 缩放 **cover** 保持比例填满

list-style: none 取消小点

visibility: inherit (默认继承父 visibility) **hidden** (占位置) **visible** 【适合对 position 和 float 的显示】

cursor: **default** (尖尖) **pointer** (小手) **grab** (抓手) **alias** (弯箭头) **cell** (十字) **progress** (转圈) **text** (文本)
move (十字) **wait** (漏斗)

opacity: 0.8; 盒子及内容全部透明 (除非定位)

outline: **none** 取消轮廓线 【表单输入默认有蓝边框可去掉】

resize: **none** 无法重新设置大小 (防止 textarea 拖拽)

box-shadow **2px** 水平偏移 **2px** 垂直偏移 **4px** 模糊距离 **[-4px]** 模糊延伸距离 **#000** 颜色 (影子不占位置) **inset**
变为内阴影 // 京东 0 -12px 10px rgba(0, 0, 0, .2);
box-shadow: 2px 2px 4px #000, 2px 2px 2px #000 多重阴影
box-shadow: 0 0 0 1000px white inset; 去掉输入框记忆数据显示的蓝色背景

文字

user-select: **none** 文本不能被选择 **text** 可以选择文本 **all** 可选择整体文本 **element** 可以选择文本, 但选择范围受元素边界的约束

font: **style** **weight** **size/line-height** **family** (size family 不能省略)

font-style: **normal** **italic** 斜体

font-weight: **normal** - 400 正常 **bold** - 700 加粗 **bolder** 更粗 **lighter** 更细

font-size: **12px** **12em** 基于父级字体大小 **12rem** 相对-基于 html 标签字体大小-默认大小

font-family: -apple-system, BlinkMacSystemFont, 'Helvetica Neue', Helvetica, Segoe UI, Arial, Roboto, 'PingFang SC', miui, 'Hiragino Sans GB', 'Microsoft Yahei',

Serif, 宋体 sans-serif; 无衬线字体, 一定显示

优先从前到后选择字体 没找到用系统默认 (中文或含特殊符号加引号) // font-family: "Times New

Roman", "microsoft yahei"!important; ==> Times New Roman

宋体 simsun \5B8B\4F53 微软雅黑 microsoft yahei \5FAE\8F6F\96C5\9ED1

font-variant: **normal** **small-caps** 显示小型大写字母的字体

font-feature-settings: "smcp"; 字体功能设置

color: 文本颜色 1. red blue pink
2. #ff0000 (red red green green blue blue) 或 #f00 (red green blue)
3. **rgb**(255,0,0) **rgb**(100%,0%,0%) **rgba**(0, 0, 0, .3); 透明度 0-1 .3 可省略 0
4. **hsl**(360, 100%, 0%) **hsla**(360, 100%, 0%, .3) 色相 (0-red 120-green 240-blue)

饱和度 亮度

overflow: **visible** 默认可见 **hidden** 溢出隐藏不合并 **scroll** 显示滚动条 **auto** 超出显示滚动条 (切框-加一个小父盒子隐藏溢出, 不让其他浮动影响内容, 定位盒子会被隐藏) // 父盒子的 overflow: hidden 会导致 子盒子的 overflow-y:auto 失效, 必须 overflow-y 对应 overflow-y

overflow-y: **hidden** 隐藏 y 轴滚动条 **scroll** 显示 y 轴滚动条 【通过外加盒子 overflow:hidden; 隐藏溢出的滚动条, 上层 overflow 会影响下层】

overflow-x: **hidden** 隐藏 x 轴滚动条 **scroll** 显示 x 轴滚动条

::-webkit-scrollbar 滚动条整体部分, 可以设置宽度啥的

::-webkit-scrollbar-button 滚动条两端的按钮

::-webkit-scrollbar-track 外层轨道

::-webkit-scrollbar-track-piece 内层轨道, 滚动条中间部分(除去)

::-webkit-scrollbar-thumb 滚动条里面可以拖动的那个

::-webkit-scrollbar-corner 边角

::-webkit-resizer 定义右下角拖动块的样式

// 滚动条样式

```
&::-webkit-scrollbar {  
    width: 5px;  
    background-color: #ff0;  
}
```

/*定义滚动条轨道

内阴影+圆角*/

```
&::-webkit-scrollbar-track {  
    -webkit-box-shadow: inset 0 0 6px rgba(0, 0, 0, 0.3);  
    border-radius: 10px;  
    background-color: #f5f5f5;  
}
```

/*定义滑块

内阴影+圆角*/

```
&::-webkit-scrollbar-thumb {  
    border-radius: 10px;  
    -webkit-box-shadow: inset 0 0 6px rgba(0, 0, 0, 0.3);  
    background-color: #555;  
}
```

white-space: 文本显示方式 **normal** 默认超出换行 **nowrap** 强制一行显示文本除非换行 **pre** 空白会被浏览器保留。其行为方式类似 HTML 中的 `<pre>` 标签。

letter-space: 字体间距

word-break: **normal** 使用浏览器默认的换行规则 **break-all** 允许在单词内换行 **keep-all** 只能在半角空格或连字符处换行。

word-spacing: **-1.5px** 内部行内块之间的间距，无法完全去除，不精确

word-wrap: break-word;

line-clamp: 2 控制行数（需要配合 `-webkit-box-orient: vertical;` `display: -webkit-box;`）【`-webkit-`】

display: **-webkit-box** 将对象作为弹性伸缩盒子模型显示【`display` 加 `box-orient` 加 `line-clamp` 加 `text-overflow` 实现两行超出省略号】

-webkit-box-orient: **vertical** 子元素的排列方式。

-webkit-line-clamp: **2** 用于显示的行数

text-overflow: 文字溢出 **clip** 默认不显示省略号 **ellipsis** 溢出文字显示省略号（必须和 `overflow: hidden` 生效，搭配 `white-space`）

text-shadow **2px** 水平 **2px** 垂直 **4px** 羽化距离 **#000** 颜色（影子不占位置，逗号添加多个）

text-align 父盒子内 文本/行内/行内块水平对齐方式 **center** **right** **left** **justify** （默认两端对齐）

text-indent 首行缩进 **2em;** 两个字的距离-可负值

text-decoration **none** **underline**（下划线） **overline**（上划线） **line-through**（删除线）

text-transform **none** 默认 **capitalize** 每个单词以大写字母开头 **uppercase** 仅有大写字母 **lowercase** 仅有小写字母 **inherit** 规定应该从父元素继承 `text-transform` 属性的值。

动画

transform: **translate(-20px, 50%)** 参照盒子左上角 xy 坐标移动 - 百分比为参照当前盒子本身移动 (y 省略-y 方向不移动, 空格分隔添加其他变换)

scale(0.8, 3) 以盒子中心点 xy 坐标缩放 (y 省略-xy 方向相同比例缩放)

rotate(90deg) 以盒子中心点顺时针旋转 (负值为逆时针旋转, 旋转会将坐标旋转-混乱)

skew(20deg, 0deg) 平行四边形水平垂直倾斜 (垂直方向省略为 0)

translate3d(2px, 2px, 2px) 以 xyz 三维移动 (z 轴和平面垂直无法显示 上升移动) //

translateX(x); translateY(y); translateZ(z)

scale3d(0.8, 3, 2) 以 xyz 三维缩放 (z 轴和平面垂直无法显示 沉下移动)

//**scaleX(x); scaleY(y);**

rotate3d(1, 1, 0, 20deg) 以 xyz 三个向量和的方向旋转 20deg 度=此方向旋转便于观察立体变换

//**rotateX(180deg); rotateY(180deg); rotateZ(180deg);**

none 默认无变换-可用于清除效果, 为 hover 清除动画时, 必须同时为当前盒子也清除动画。transition: none !important;

transform-origin: center bottom; 下方中心为 旋转中心点

20px 30px

transform-style: preserve-3d 子元素保留在 z 轴变换的效果 **flat** 默认还原在 z 轴移动的效果 (父级添加保留并旋转 子级在旋转后的 xyz 轴中变换)

backface-visibility: hidden; 定义当元素不面向屏幕时是否可见 //双图旋转 backface-visibility: hidden;

transition: transform 将要变化的样式 **2s** 耗时 **ease** 速度曲线 **2s** 延迟 (逗号分隔设置多个, 过渡执行完毕会恢复到原状态, 只能两个状态间的变换) -交集选择器动态添加 class // transition: all 0 ease 0;

transition-property 将要变化的含准确值的 CSS 样式 (all 过渡所有变化的样式)

transition-duration 花费时间 (默认 0s 特殊 500ms)

transition-timing-function 速度曲线 **ease** 默认慢-快-慢 **linear** 匀速 **ease-in** 加速 **ease-out** 减速 **steps(4)** 匀速分步进行 ease-in-out 慢 - 慢 **cubic-bezier(0.25, 0.25, 0.75, 0.75);** 自定义函数

transition-delay 延迟时间 (默认 0, x 秒过后在移动)

perspective: 20px 视角从 xy 轴中心面后退 20px 观察

perspective-origin: 20px 30px 摄像机从向右移动 20px 向上移动 30px

@keyframes xxx { 0% {} 50% {} 100% {} } 多状态变化 百分比表示状态在总耗时的位置 (不会影响之前的属性, 动画执行完毕恢复原状态)

{ from {} 50% {} to {} } 可以用 from to 代替 0% 100%≈

animation: xxx 将要执行的动画 **2s** 耗时 **ease** 速度曲线 **2s** 延迟时间 **1** 播放次数 **alternate** 播放顺序 (可以多个状态间的变换)

animation-name 动画名称 (不可省略)

animation-duration 花费时间 (默认 0s) (不可省略) 0% - 100%

animation-timing-function 速度曲线 **ease** 默认快变慢 **linear** 匀速 **ease-in** 低速开始 ease-in-out 低速开始和结束

ease-out 低速结束 **steps(4)** 匀速分步进行

cubic-bezier(0.25, 0.25, 0.75, 0.75); 贝塞尔函数 cubic-bezier (x1,y1,x2,y2) x1,y1

初始时间和状态 x2y2 结束时间和状态 (0<x1x2<1)

animation-delay 延迟时间 (默认 0s)

animation-iteration-count: 播放次数 **1** 默认一次 **infinite** 无限播放

animation-direction 播放顺序 **normal** 默认每次从开始到结束 **alternate** 去回两次变化

animation-play-state 播放状态 **paused** 暂停 **running** 播放

animation-fill-mode

开始结束状态 backwards 默认立刻进入初始状态, 不保留结束状态

forwards 不会立刻进入初始状态, 保留结束状态

both 立刻进入初始状态, 保留结束状态

媒体

@media screen and (min-width: 320px) and (max-width: 768px) { .container { } } //

响应式布局, 根据屏幕宽度

@media(-webkit-min-device-pixel-ratio:1.5), (min-device-pixel-ratio:1.5) .border-1px &::after

-webkit-transform:scaleY(0.7) transform:scaleY(0.7)

@media(-webkit-min-device-pixel-ratio:2), (min-device-pixel-ratio:2) .border-1px &::after

-webkit-transform:scaleY(0.5) transform:scaleY(0.5)

min-height 在移动端代替 height (height 在移动端经常显示不出来)

max-height 限制最大高度防止 固定高度时, 屏幕过小时的换行

max-width 限制宽度能防止 百分比宽度超出范围

HtmlAndCss / LESS

基础

@import "../banner"; @import url(../banner);

导入其他 less 文件

@color: red; @size: 14px; @style: ffff;

.ca { background: @color; font-size: @size; } .@{style}aa { } 变量

.cb { .ca(); .cc(); width: ~'calc(100%-20)'; } 引入其他类, 引入函数函数内容 (less 不能识别的

语法, 加 ~ 不编译)

.ca { .son { } > div { &:hover { &::before { }

嵌套 (&为父级, 直接使用选择器会在前方加上

父级)

运算符

body {

margin: (14px/2);

top: 50px + 100px;

right: \$var * 10%;

}

函数

.func(@val){

函数不会编译成 css, 不能和选择器名相同, 同类一样直接替换 (可重载调用)

height:@val;

}

.func (@val: 20px) when(@val<=@len){ 满足 when 条件执行函数形成循环 (:20px 为默认值)

height:@val; .func(@val+1)

}

内置

lighten(red, 20%) 变亮

darken(red, 20%) 变暗

length(@List) 取数组长度

extract(@List, @index) 获取数组值

HtmlAndCss / SASS

基础

@import "../banner"; @import url(../banner);

导入其他 sass 文件

```

$color: red;                定义变量
$size: 14px;
$style: ffff !default;
.use {                       使用变量
    background: $color;
    font-size: $size;
}
.#{ $style }aa { }          模板变量
.use {
    @extend .ca;             引入其他类,
    @include func(20px);     引入函数内容, 调用自定义函数
}
.use {
    .son{ } > div{ } &:hover{ } &::before{ }
}

```

嵌套 (&为父级, 直接使用选择器会在前

方加上父级)

```

[layout] {                  @extend [layout];
    box-sizing: border-box;
    display: -webkit-flex;
    display: -moz-flex;
    display: -ms-flexbox;
    display: flex;
}
[layout=column] {          @extend [layout=column];
    -ms-flex-direction: column;
    flex-direction: column;
    -webkit-flex-direction: column;
}
@font-face {
    font-family: 'numfont'; /*声明字体名称*/
    src: url('~@/assets/font/DINLight.ott');
    font-style: normal; /*倾斜字体变正常*/
}

```

语句

```

p {
    @if 1 + 1 == 2 { border: 1px solid; }
    @else { background-color: #fff; }
    @while $i > 0 {
        .item-#{ $i } { width: 2em * $i; }
        $i: $i - 2;
    }
}

```

运算符

```

body {
    margin: (14px/2);
    top: 50px + 100px;
}

```

```
right: $var * 10%;  
}
```

函数

@mixin func(\$x, \$y, \$blur, \$c :10px){ 函数定义以@mixin 开头，后跟自定义名称（可重载调用，设置默认值）
 box-shadow: \$x, \$y, \$blur, \$

```
}
```

@function funcA(\$n) { 自定义函数
 @return \$n * 2;

```
}
```

内置

```
lighten(#cc3, 10%)    // #d6d65c  
darken(#cc3, 10%)    // #a3a329  
grayscale(#cc3)       // #808080  
complement(#cc3)     // #33c
```