

# 前端常用英语汇总

此部分是对小满整理总结的进行二次整理

## 前端常用英语汇总

HTML (只统计常用标签)

Css (只统计常用的)

js (统计常用 API)

- Node系列
- parentNode接口
- ChildNode接口
- document快捷方式
- 节点集合属性
- 文档静态信息属性
- 文档状态属性
- 其他属性
- 元素特性相关属性
- 元素状态的相关属性
- 尺寸属性
- 节点相关属性
- 属性方法
- 查找方法
- 事件方法
- 其他

## HTML (只统计常用标签)

< !DOCTYPE >	声明必须位于 HTML5 文档中的第一行，也就是位于 < html > 标签之前。该标签告知浏览器文档所使用的 HTML 规范
< html >	表示一个 HTML 文档的根（顶级元素），所以它也被称为根元素。所有其他元素必须是此元素的后代。
< head >	HTML head 元素 规定文档相关的配置信息（元数据），包括文档的标题，引用的文档样式和脚本等
< meta >	元素可提供有关页面的元信息（meta-information），比如针对搜索引擎和更新频度的描述和关键词
< body >	表示文档的内容。 属性提供了可以轻松访问文档的 body 元素的脚本
< title >	定义文档的标题，显示在浏览器的标题栏或标签页上。它只应该包含文本，若是包含有标签，则它包含的任何标签都将被忽略
< style >	定义 HTML 文档的样式信息
< link >	大多数时候都用来连接样式

< !DOCTYPE >	声明必须位于 HTML5 文档中的第一行，也就是位于 < html > 标签之前。该标签告知浏览器文档所使用的 HTML 规范
< header >	用于展示介绍性内容，通常包含一组介绍性的或是辅助导航的实用元素。它可能包含一些标题元素，但也可能包含其他元素，比如 Logo、搜索框、作者名称，等等
< footer >	表示最近一个章节内容或者根节点元素的页脚。
< article >	元素表示文档、页面、应用或网站中的独立结构，其意在成为可独立分配的或可复用的结构，如在发布中，它可能是论坛帖子、杂志或新闻文章、博客、用户提交的评论、交互式组件，或者其他独立的内容项目
< section >	表示 HTML 文档中一个通用独立章节，它没有更具体的语义元素来表示。一般来说会包含一个标题。
< p >	表示文本的一个段落。该元素通常表现为一整块与相邻文本分离的文本，或以垂直的空白隔离或以首行缩进（块级元素）
< div >	是一个通用型的流内容容器，在不使用CSS其对内容或布局没有任何影响（块级元素）
< span >	元素是短语内容的通用行内容容器，并没有任何特殊语义。（内联元素）
< img >	元素将一份图像嵌入文档
< aside >	元素表示一个和其余页面内容几乎无关的部分，被认为是独立于该内容的一部分并且可以被单独的拆分出来而不会使整体受影响
< audio >	元素用于在文档中嵌入音频内容
< video >	元素用于在文档中嵌入视频内容
< source >	标签为媒介元素（比如 < video > 和 < audio >）定义媒介资源
< canvas >	元素可被用来通过 JavaScript（CanvasAPI 或 WebGL）绘制图形及图形动画
< nav >	表示页面的一部分，其目的是在当前文档或其他文档中提供导航链接。导航部分的常见示例是菜单，目录和索引
< ul >	表示一个内可含多个元素的无序列表或项目符号列表
< ol >	表示有序列表，通常渲染为一个带编号的列表。
< li >	用于表示列表里的条目。
< script >	用于嵌入或引用可执行脚本。
< h1 - h6 >	代表网页中的标题数字越小越大
< a >	标签定义超链接，用于从一个页面链接到另一个页面。
< b >	标签定义粗体的文本。
< br / >	标签插入简单的换行符。
< hr >	标签水平线

< !DOCTYPE >	声明必须位于 HTML5 文档中的第一行，也就是位于 < html > 标签之前。该标签告知浏览器文档所使用的 HTML 规范
< iframe >	嵌套另一个网页
< form >	用于创建供用户输入的 HTML 表单
< input >	表单输入元素
< textarea >	文本域
< button >	定义按钮
< table >	定义表格
< tbody >	定义一段表格主体
< thead >	定义表格的表头主体
< tfoot >	定义表格的页脚
< tr >	标签定义表格中的行
< th >	标签定义 HTML 表格中的表头单元格
< td >	标签定义 HTML 表格中的标准单元格。

## Css (只统计常用的)

color	定义颜色
opacity	检索或设置对象的不透明度
font-size	定义字体大小
font-style	定义字体风格例如斜体
font-weight	定义字体粗细
font-family	嵌入字体
text-indent	首行缩进
text-align	水平对其方式
text-overflow	文本溢出
text-shadow	文本阴影
white-space	处理换行和空格
line-height	行高
background	定义背景
background-color	指定背景颜色

color	定义颜色
background-image	背景图片
background-repeat	背景平铺排列
background-position	背景定位
background-attachment	背景是否固定
border	定义边框
border-color	边框颜色
border-radius	边框圆角
border-style	边框样式
border-image	边框图片
box-shadow	阴影
padding	内边距
margin	外边距
width	宽度
height	高度
max-width min-width	最大宽度 最小宽度
max-height min-height	最大高度 最小高度
display flex	弹性盒
flex-direction	调整主轴XY
flex-wrap	换行
justify-content	水平方位调整
align-items	垂直方位调整
flex-grow	调整比例
flex-shrink	flex 元素的收缩规则
flex-basis	指定 flex 元素在主轴方向上的初始大小
flex	三个属性合体flex-grow flex-shrink flex-basis
position	固定定位 相对定位 绝对定位 粘性定位
left top bottom right	四个方位
z-index	定位层级
transform	旋转，缩放，倾斜或平移给定元素

color	定义颜色
transition	动画过度
animation	动画
linear-gradient()	颜色渐变
filter	背景模糊 变色等
calc()	计算函数
overflow	内容溢出操作
float	浮动
clear	清除浮动
长度	px rem em vh vw cm mm pt pc ch ex

## js (统计常用 API)

var let const	声明变量
if else	条件语句
switch case	条件语句
alert confirm prompt	弹框
function	函数
for	循环
while	循环
break	终止循环
continue	跳过循环
事件省略，在Vue3笔记(小满版本)中有详细整理	
try catch	异常捕获
throw	抛异常
String Number Boolean Null Object Array Symbol BigInt undefined	基本类型
Promise then catch reject resolve all any race allSettled	DOM
window location screen history Navigator	BOM

## Node系列

Node.nodeName	返回节点名称
Node.nodeType	返回节点类型的常数值
Node.nodeValue	返回Text或Comment节点的文本值
Node.textContent	返回当前节点和它的所有后代节点的文本内容，可读写
Node.baseURI	返回当前网页的绝对路径
Node.ownerDocument	返回当前节点所在的顶层文档对象，即document
Node.nextSibling	返回紧跟在当前节点后面的第一个兄弟节点
Node.previousSibling	返回当前节点前面的、距离最近的一个兄弟节点
Node.parentNode	返回当前节点的父节点
Node.parentElement	返回当前节点的父Element节点
Node.childNodes	返回当前节点的所有子节点
Node.firstChild	返回当前节点的第一个子节点
Node.lastChild	返回当前节点的最后一个子节点
Node.isConnectoted	返回一个布尔值，表示当前节点是否在文档之中
Node.appendChild(node)	向节点添加最后一个子节点
Node.hasChildNodes()	返回布尔值，表示当前节点是否有子节点
Node.cloneNode(true);	默认为false(克隆节点), true(克隆节点及其属性，以及后代)
Node.insertBefore(newNode,oldNode)	在指定子节点之前插入新的子节点
Node.removeChild(node)	删除节点，在要删除节点的父节点上操作
Node.replaceChild(newChild,oldChild)	替换节点
Node.contains(node)	返回一个布尔值，表示参数节点是否为当前节点的后代节点。
Node.compareDocumentPosition(node)	返回一个7个比特位的二进制值，表示参数节点和当前节点的关系
Node.isEqualNode(node)	返回布尔值，用于检查两个节点是否相等。所谓相等的节点，指的是两个节点的类型相同、属性相同、子节点相同。
Node.isSameNode(node)	返回一个布尔值，表示两个节点是否为同一个节点

Node.nodeName	返回节点名称
Node.normalize()	用于清理当前节点内部的所有Text节点。它会去除空的文本节点，并且将毗邻的文本节点合并成一个。
Node.getRootNode()	返回当前节点所在文档的根节点document，与ownerDocument属性的作用相同。
Node.length	返回 NodeList 实例包含的节点数量
Node.forEach(fn, this)	用于遍历 NodeList 的所有成员
Node.item(index)	接受一个整数值作为参数，表示成员的位置，返回该位置上的成员
Node.keys()	返回键名的遍历器
Node.values()	返回键值的遍历器
Node.entries()	返回的遍历器同时包含键名和键值的信息

## parentNode接口

Node.children	返回指定节点的所有Element子节点
Node.firstElementChild	返回当前节点的第一个Element子节点
Node.lastElementChild	返回当前节点的最后一个Element子节点
Node.childElementCount	返回当前节点所有Element子节点的数目
Node.append()	为当前节点追加一个或多个子节点，位置是最后一个元素子节点的后面。
Node.prepend()	为当前节点追加一个或多个子节点，位置是第一个元素子节点的前面。

## ChildNode接口

Node.remove()	用于从父节点移除当前节点
Node.before()	用于在当前节点的前面，插入一个或多个同级节点。两者拥有相同的父节点。
Node.after()	用于在当前节点的后面，插入一个或多个同级节点，两者拥有相同的父节点
Node.replaceWith()	使用参数节点，替换当前节点

## document快捷方式

document.defaultView	返回document对象所属的window对象
document.doctype	返回文档类型节点
document.documentElement	返回当前文档的根节点
document.body	返回当前文档的< body >节点
document.head	返回当前文档的< head >节点
document.activeElement	返回当前文档中获得焦点的那个元素
document.fullscreenElement	返回当前以全屏状态展示的 DOM 元素

## 节点集合属性

document.links	返回当前文档的所有a元素
document.forms	返回页面中所有表单元素
document.images	返回页面中所有图片元素
document.embeds	返回网页中所有嵌入对象
document.scripts	返回当前文档的所有脚本
document.styleSheets	返回当前网页的所有样式表

## 文档静态信息属性

document.documentURI	表示当前文档的网址
document.URL	返回当前文档的网址
document.domain	返回当前文档的域名
document.lastModified	返回当前文档最后修改的时间
document.location	返回location对象，提供当前文档的URL信息
document.referrer	返回当前文档的访问来源
document.title	返回当前文档的标题
document.compatMode	返回浏览器处理文档的模式
document.characterSet	返回渲染当前文档的字符集，比如UTF-8、ISO-8859-1



## 文档状态属性

<b>document.readyState</b>	<b>返回当前文档的状态</b>
document.hidden	返回一个布尔值，表示当前页面是否可见
document.visibilityState	返回文档的可见状态

## 其他属性

<b>document.cookie</b>	<b>用来操作Cookie</b>
document.designMode	控制当前文档是否可编辑，可读写
document.currentScript	返回当前脚本所在的那个 DOM 节点，即< script >元素的 DOM 节点
document.implementation	返回一个DOMImplementation对象

## 元素特性相关属性

<b>Element.id</b>	<b>返回指定元素的id属性，可读写</b>
Element.tagName	返回指定元素的大写标签名
Element.dir	用于读写当前元素的文字方向
Element.accessKey	用于读写分配给当前元素的快捷键
Element.draggble	返回一个布尔值，表示当前元素是否可拖动
Element.lang	返回当前元素的语言设置。该属性可读写
Elemnt.tabIndex	返回一个整数，表示当前元素在 Tab 键遍历时的顺序。该属性可读写。
Element.title	用来读写当前元素的 HTML 属性title
Element.attributes	返回当前元素节点的所有属性节点
Element.className	返回当前元素的class属性，可读写
Element.classList	返回当前元素节点的所有class集合
Element.innerHTML	返回该元素包含的HTML代码，可读写
Element.outerHTML	返回指定元素节点的所有HTML代码，包括它自身和包含的的所有子元素，可读写
Element.dataset	返回元素节点中所有的data-*属性

## 元素状态的相关属性

Element.hidden	返回一个布尔值，表示当前元素的hidden属性，用来控制当前元素是否可见。该属性可读写
Element.contentEditable	返回一个字符串，表示是否设置了contenteditable属性
Element.isContentEditable	返回一布尔值，表示是否设置了contenteditable属性，只读

## 尺寸属性

Element.clientHeight	返回元素节点可见部分的高度
Element.clientWidth	返回元素节点可见部分的宽度
Element.clientLeft	返回元素节点左边框的宽度
Element.clientTop	返回元素节点顶部边框的宽度
Element.scrollHeight	返回元素节点的总高度
Element.scrollWidth	返回元素节点的总宽度
Element.scrollLeft	返回元素节点的水平滚动条向右滚动的像素数值,通过设置这个属性可以改变元素的滚动位置
Element.scrollTop	返回元素节点的垂直滚动向下滚动的像素数值
Element.offsetParent	返回最靠近当前元素的、并且 CSS 的position属性不等于static的上层元素
Element.offsetHeight	返回元素的垂直高度(包含border,padding)
Element.offsetWidth	返回元素的水平宽度(包含border,padding)
Element.offsetLeft	返回当前元素左上角相对于Element.offsetParent节点的垂直偏移
Element.offsetTop	返回水平位移
Element.style	返回元素节点的行内样式

## 节点相关属性

Element.children	包括当前元素节点的所有子元素
Element.childElementCount	返回当前元素节点包含的子HTML元素节点的个数
Element.firstChild	返回当前节点的第一个Element子节点
Element.lastElementChild	返回当前节点的最后一个Element子节点
Element.nextElementSibling	返回当前元素节点的下一个兄弟HTML元素节点
Element.previousElementSibling	返回当前元素节点的前一个兄弟HTML节点

## 属性方法

<b>Element.getAttribute()</b>	<b>读取指定属性</b>
Element.getAttributeNames()	返回当前元素的所有属性名
Element.setAttribute()	设置指定属性
Element.hasAttribute()	返回一个布尔值，表示当前元素节点是否有指定的属性
Element.hasAttributes()	当前元素是否有属性
Element.removeAttribute()	移除指定属性

## 查找方法

<b>Element.querySelector()</b>	<b>接受 CSS 选择器作为参数，返回父元素的第一个匹配的子元素</b>
Element.querySelectorAll()	接受 CSS 选择器作为参数，返回一个NodeList实例，包含所有匹配的子元素
Element.getElementsByTagName()	返回一个HTMLCollection实例，成员是当前节点的所有匹配指定标签名的子元素节点
Element.getElementsByClassName()	返回一个HTMLCollection实例，成员是当前元素节点的所有具有指定 class 的子元素节点
Element.closest()	接受一个 CSS 选择器作为参数，返回匹配该选择器的、最接近当前节点的一个祖先节点（包括当前节点本身）

## 事件方法

<b>Element.addEventListener()</b>	<b>添加事件的回调函数</b>
Element.removeEventListener()	移除事件监听函数
Element.dispatchEvent()	触发事件

## 其他

<b>Element.matches()</b>	<b>返回一个布尔值，表示当前元素是否匹配给定的 CSS 选择器</b>
Element.scrollIntoView()	滚动当前元素，进入浏览器的可见区域
Element.getBoundingClientRect()	返回一个对象，包含 top,left,right,bottom,width,height
Element.getClientRects()	返回当前元素在页面上形参的所有矩形
Element.insertAdjacentHTML(where, htmlString);	解析HTML字符串，然后将生成的节点插入DOM树的指定位置

Element.matches()	返回一个布尔值，表示当前元素是否匹配给定的 CSS 选择器
Element.insertAdjacentHTML('beforeBegin', htmlString)	在该元素前插入
Element.insertAdjacentHTML('afterBegin', htmlString)	在该元素第一个子元素前插入
Element.insertAdjacentHTML('beforeEnd', htmlString)	在该元素最后一个子元素后面插入
Element.insertAdjacentHTML('afterEnd', htmlString)	在该元素后插入
Element.remove()	用于将当前元素节点从DOM中移除
Element.focus()	用于将当前页面的焦点，转移到指定元素上
Element.blur()	用于将焦点从当前元素移除