# 字符串常用的属性和方法

## 属性

#### length:返回字符串的长度

## 方法

#### chatAt():返回在指定位置的字符

#### charCodeAt():返回在指定位置字符的unicode编码(ASCII编码)

#### concat():连接字符串

#### indexOf():从字符串的开头向后搜索字符串

#### lastIndexOf():从字符串的末尾向前搜索字符串

#### match():找到一个或多个正则表达式的匹配

#### replace():替换与正则表达式匹配的字串

#### search():检索与正则表达式相匹配的值

#### slice():提取字符串的片段，并在新的字符串中返回被提取的部分

#### split():把字符串分割成字符串数组

#### substr():从起始索引号提取字符串中指定数目的字符

#### substring():截取字符串中两个指定的索引号之间的字符

#### toLowerCase():将字符串转换为小写

#### toUpperCase():将字符串转换为大写

#### toString():返回字符串本身

#### valueOf():返回某个对象的原始值

#### trim():删除前置及后缀的所有空格

#### includes():判断字符串是否包含目标字符串

# 数组常用的属性和方法

| **方法名** | **对应版本** | **功能** | **原数组是否改变** |
| --- | --- | --- | --- |
| concat() | ES5- | 合并数组，并返回合并之后的数据 | n |
| join() | ES5- | 使用分隔符，将数组转为字符串并返回 | n |
| pop() | ES5- | 删除最后一位，并返回删除的数据 | y |
| shift() | ES5- | 删除第一位，并返回删除的数据 | y |
| unshift() | ES5- | 在第一位新增一或多个数据，返回长度 | y |
| push() | ES5- | 在最后一位新增一或多个数据，返回长度 | y |
| reverse() | ES5- | 反转数组，返回结果 | y |
| slice() | ES5- | 截取指定位置的数组，并返回 | n |
| sort() | ES5- | 排序（字符规则），返回结果 | y |
| splice() | ES5- | 删除指定位置，并替换，返回删除的数据 | y |
| toString() | ES5- | 直接转为字符串，并返回 | n |
| valueOf() | ES5- | 返回数组对象的原始值 | n |
| indexOf() | ES5 | 查询并返回数据的索引 | n |
| lastIndexOf() | ES5 | 反向查询并返回数据的索引 | n |
| forEach() | ES5 | 参数为回调函数，会遍历数组所有的项，回调函数接受三个参数，分别为value，index，self；forEach没有返回值 | n |
| map() | ES5 | 同forEach，同时回调函数返回数据，组成新数组由map返回 | n |
| filter() | ES5 | 同forEach，同时回调函数返回布尔值，为true的数据组成新数组由filter返回 | n |
| every() | ES5 | 同forEach，同时回调函数返回布尔值，全部为true，由every返回true | n |
| some() | ES5 | 同forEach，同时回调函数返回布尔值，只要由一个为true，由some返回true | n |
| reduce() | ES5 | 归并，同forEach，迭代数组的所有项，并构建一个最终值，由reduce返回 | n |
| reduceRight() | ES5 | 反向归并，同forEach，迭代数组的所有项，并构建一个最终值，由reduceRight返回 | n |

# 变量的类型判断

var v1 = [1, 2, 3, 4, 5];

    // typeof可以进行类型判断

    console.log(typeof v1);

    if (typeof v1 == "string") {

        console.log("是字符串");

    }

    // instanceof 判断某个对象是否属于某个属性

    if (v1 instanceof Array) {

        console.log("是数组");

    }

## **Math对象的属性和方法**

## Math 对象属性

|  |  |
| --- | --- |
| **属性** | **描述** |
| E | 返回算术常量 e，即自然对数的底数（约等于2.718）。 |
| ln2 | 返回 2 的自然对数（约等于0.693）。 |
| ln10 | 返回 10 的自然对数（约等于2.302）。 |
| log2e | 返回以 2 为底的 e 的对数（约等于 1.414）。 |
| log10e | 返回以 10 为底的 e 的对数（约等于0.434）。 |
| PI | 返回圆周率（约等于3.14159）。 |
| [sqrt1](https://www.w3school.com.cn/jsref/jsref_sqrt1_2.asp)\_2 | 返回返回 2 的平方根的倒数（约等于 0.707）。 |
| sqrt2 | 返回 2 的平方根（约等于 1.414）。 |

## Math 对象方法

|  |  |
| --- | --- |
| **方法** | **描述** |
| [abs(x)](https://www.w3school.com.cn/jsref/jsref_abs.asp) | 返回数的绝对值。 |
| [acos(x)](https://www.w3school.com.cn/jsref/jsref_acos.asp) | 返回数的反余弦值。 |
| [asin(x)](https://www.w3school.com.cn/jsref/jsref_asin.asp) | 返回数的反正弦值。 |
| [atan(x)](https://www.w3school.com.cn/jsref/jsref_atan.asp) | 以介于 -PI/2 与 PI/2 弧度之间的数值来返回 x 的反正切值。 |
| [atan2(y,x)](https://www.w3school.com.cn/jsref/jsref_atan2.asp) | 返回从 x 轴到点 (x,y) 的角度（介于 -PI/2 与 PI/2 弧度之间）。 |
| [ceil(x)](https://www.w3school.com.cn/jsref/jsref_ceil.asp) | 对数进行上舍入。 |
| [cos(x)](https://www.w3school.com.cn/jsref/jsref_cos.asp) | 返回数的余弦。 |
| [exp(x)](https://www.w3school.com.cn/jsref/jsref_exp.asp) | 返回 e 的指数。 |
| [floor(x)](https://www.w3school.com.cn/jsref/jsref_floor.asp) | 对数进行下舍入。 |
| [log(x)](https://www.w3school.com.cn/jsref/jsref_log.asp) | 返回数的自然对数（底为e）。 |
| [max(x,y)](https://www.w3school.com.cn/jsref/jsref_max.asp) | 返回 x 和 y 中的最高值。 |
| [min(x,y)](https://www.w3school.com.cn/jsref/jsref_min.asp) | 返回 x 和 y 中的最低值。 |
| [pow(x,y)](https://www.w3school.com.cn/jsref/jsref_pow.asp) | 返回 x 的 y 次幂。 |
| [random()](https://www.w3school.com.cn/jsref/jsref_random.asp) | 返回 0 ~ 1 之间的随机数。 |
| [round(x)](https://www.w3school.com.cn/jsref/jsref_round.asp) | 把数四舍五入为最接近的整数。 |
| [sin(x)](https://www.w3school.com.cn/jsref/jsref_sin.asp) | 返回数的正弦。 |
| [sqrt(x)](https://www.w3school.com.cn/jsref/jsref_sqrt.asp) | 返回数的平方根。 |
| [tan(x)](https://www.w3school.com.cn/jsref/jsref_tan.asp) | 返回角的正切。 |
| [toSource()](https://www.w3school.com.cn/jsref/jsref_tosource_math.asp) | 返回该对象的源代码。 |
| [valueOf()](https://www.w3school.com.cn/jsref/jsref_valueof_math.asp) | 返回 Math 对象的原始值。 |

## **[JS获取浏览器窗口大小 获取屏幕，浏览器，网页高度宽度](https://www.cnblogs.com/wujindong/p/5592908.html)**

网页可见区域宽：document.body.clientWidth

网页可见区域高：document.body.clientHeight

网页可见区域宽：document.body.offsetWidth (包括边线的宽)

网页可见区域高：document.body.offsetHeight (包括边线的宽)

网页正文全文宽：document.body.scrollWidth

网页正文全文高：document.body.scrollHeight

网页被卷去的高：document.body.scrollTop

网页被卷去的左：document.body.scrollLeft

网页正文部分上：window.screenTop

网页正文部分左：window.screenLeft

屏幕分辨率的高：window.screen.height

屏幕分辨率的宽：window.screen.width

屏幕可用工作区高度：window.screen.availHeight

屏幕可用工作区宽度：window.screen.availWidth

HTML精确定位:scrollLeft,scrollWidth,clientWidth,offsetWidth

scrollHeight: 获取对象的滚动高度。

scrollLeft:设置或获取位于对象左边界和窗口中目前可见内容的最左端之间的距离

scrollTop:设置或获取位于对象最顶端和窗口中可见内容的最顶端之间的距离

scrollWidth:获取对象的滚动宽度

offsetHeight:获取对象相对于版面或由父坐标 offsetParent 属性指定的父坐标的高度

offsetLeft:获取对象相对于版面或由 offsetParent 属性指定的父坐标的计算左侧位置

offsetTop:获取对象相对于版面或由 offsetTop 属性指定的父坐标的计算顶端位置

event.clientX 相对文档的水平座标

event.clientY 相对文档的垂直座标

event.offsetX 相对容器的水平坐标

event.offsetY 相对容器的垂直坐标

document.documentElement.scrollTop 垂直方向滚动的值

event.clientX+document.documentElement.scrollTop 相对文档的水平座标+垂直方向滚动的量

IE，FireFox 差异如下：

IE6.0、FF1.06+：

clientWidth = width + padding

clientHeight = height + padding

offsetWidth = width + padding + border

offsetHeight = height + padding + border

IE5.0/5.5：

clientWidth = width - border

clientHeight = height - border

offsetWidth = width

offsetHeight = height

(需要提一下：CSS中的margin属性，与clientWidth、offsetWidth、clientHeight、offsetHeight均无关)

网页可见区域宽： document.body.clientWidth

网页可见区域高： document.body.clientHeight

网页可见区域宽： document.body.offsetWidth (包括边线的宽)

网页可见区域高： document.body.offsetHeight (包括边线的高)

网页正文全文宽： document.body.scrollWidth

网页正文全文高： document.body.scrollHeight

网页被卷去的高： document.body.scrollTop

网页被卷去的左： document.body.scrollLeft

网页正文部分上： window.screenTop

网页正文部分左： window.screenLeft

屏幕分辨率的高： window.screen.height

屏幕分辨率的宽： window.screen.width

屏幕可用工作区高度： window.screen.availHeight

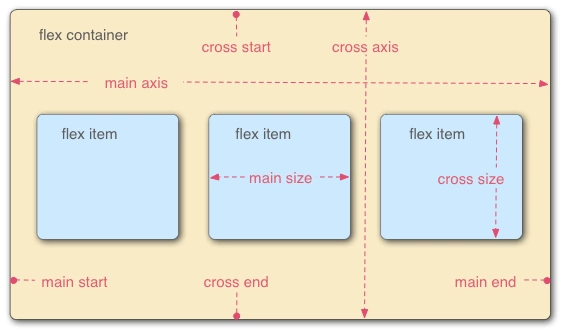
屏幕可用工作区宽度： window.screen.availWidth

**弹性布局**

|  |  |
| --- | --- |
| [flex-direction](https://www.w3cschool.cn/cssref/css3-pr-flex-direction.html" \t "https://www.w3cschool.cn/css3/_blank) | 指定了弹性容器中子元素的排列方式 |
| [justify-content](https://www.w3cschool.cn/cssref/css3-pr-justify-content.html" \t "https://www.w3cschool.cn/css3/_blank) | 设置弹性盒子元素在主轴（横轴）方向上的对齐方式。 |
| [align-items](https://www.w3cschool.cn/cssref/css3-pr-align-items.html" \t "https://www.w3cschool.cn/css3/_blank) | 设置弹性盒子元素在侧轴（纵轴）方向上的对齐方式。 |
| [flex-wrap](https://www.w3cschool.cn/cssref/css3-pr-flex-wrap.html" \t "https://www.w3cschool.cn/css3/_blank) | 设置弹性盒子的子元素超出父容器时是否换行。 |
| [align-content](https://www.w3cschool.cn/cssref/css3-pr-align-content.html" \t "https://www.w3cschool.cn/css3/_blank) | 修改 flex-wrap 属性的行为，类似align-items, 但不是设置子元素对齐，而是设置行对齐 |
| [flex-flow](https://www.w3cschool.cn/cssref/css3-pr-flex-flow.html" \t "https://www.w3cschool.cn/css3/_blank) | flex-direction 和 flex-wrap 的简写 |
| [order](https://www.w3cschool.cn/cssref/css3-pr-order.html" \t "https://www.w3cschool.cn/css3/_blank) | 设置弹性盒子的子元素排列顺序。 |
| [align-self](https://www.w3cschool.cn/cssref/css3-pr-align-self.html" \t "https://www.w3cschool.cn/css3/_blank) | 在弹性子元素上使用。覆盖容器的 align-items 属性。 |
| [flex](https://www.w3cschool.cn/cssref/css3-pr-flex.html" \t "https://www.w3cschool.cn/css3/_blank) | 设置弹性盒子的子元素如何分配空间。 |

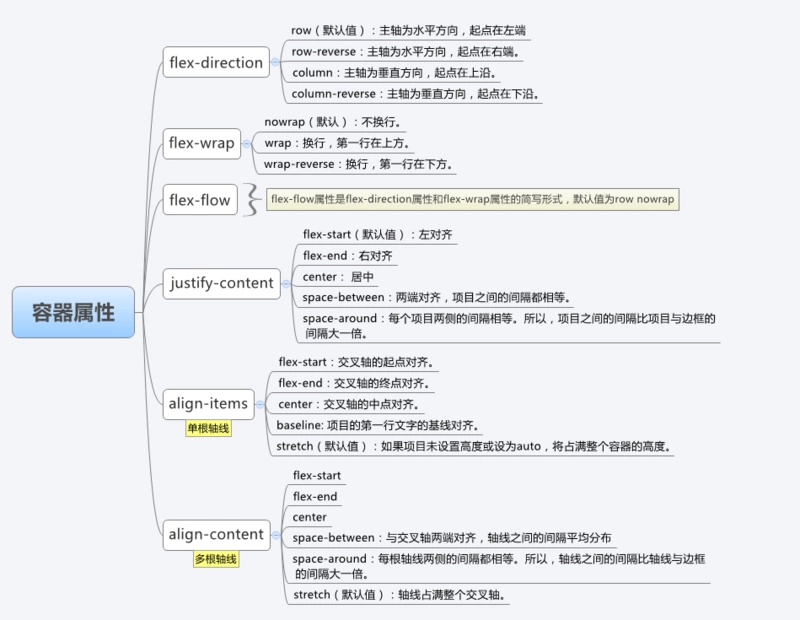
## 一、基本概念

采用 Flex 布局的元素，称为 Flex 容器（flex container），简称"容器"。它的所有子元素自动成为容器成员，称为 Flex 项目（flex item），简称"项目"。



在 Flexbox 模型中，有三个核心概念：  
– flex 项（注：也称 flex 子元素），需要布局的元素  
– flex 容器，其包含 flex 项  
– 排列方向（direction），这决定了 flex 项的布局方向

## 二、容器属性



### 2.1  flex-direction:

* row（默认值）：主轴为水平方向，起点在左端。
* row-reverse：主轴为水平方向，起点在右端。
* column：主轴为垂直方向，起点在上沿。
* column-reverse：主轴为垂直方向，起点在下沿。

### 2.2   flex-wrap:

* nowrap（默认）：不换行。
* wrap：换行，第一行在上方。
* wrap-reverse：换行，第一行在下方。

### 2.3  justify-content:

* flex-start（默认值）：左对齐
* flex-end：右对齐
* center： 居中
* space-between：两端对齐，项目之间的间隔都相等。
* space-around：每个项目两侧的间隔相等。所以，项目之间的间隔比项目与边框的间隔大一倍。

### 2.4  align-items:

* flex-start：交叉轴的起点对齐。
* flex-end：交叉轴的终点对齐。
* center：交叉轴的中点对齐。
* baseline: 项目的第一行文字的基线对齐。
* stretch（默认值）：如果项目未设置高度或设为auto，将占满整个容器的高度。

## 2.5  align-content:

定义了多根轴线的对齐方式，如果项目只有一根轴线，那么该属性将不起作用

* flex-start：与交叉轴的起点对齐。
* flex-end：与交叉轴的终点对齐。
* center：与交叉轴的中点对齐。
* space-between：与交叉轴两端对齐，轴线之间的间隔平均分布。
* space-around：每根轴线两侧的间隔都相等。所以，轴线之间的间隔比轴线与边框的间隔大一倍。
* stretch（默认值）：轴线占满整个交叉轴。

### 结合justify-content和align-items，看看在flex-direction两个不同属性值的作用下，轴心有什么不同：

## 三、项目属性



# **Grid网格布局**

http://www.ruanyifeng.com/blog/2019/03/grid-layout-tutorial.html

# **代码片段**

"Print to console":{

        "prefix":"reac",

        "body": [

            "import React,{Component} from \"react\";",

            "import \"./$1.css\";",

            "",

            "class $1 extends Component {",

            "\tconstructor(props){",

            "\t\tsuper(props);",

            "\t}",

            "\trender(){",

            "\t\treturn (",

            "\t\t\t<div>",

            "\t\t\t\t",

            "\t\t\t</div>",

            "\t\t);",

            "\t}",

            "}",

            "",

            "export default $1;"

        ],

        "description": "React 组件类 代码片段"

    }