动画

2019年12月10日

9:06

@keyframes         规定动画.

animation         所有动画属性的简写属性，除了 animation-play-state 属性。

animation-name         规定 @keyframes 动画的名称。

animation-duration         规定动画完成一个周期所花费的秒或毫秒 。默认是 0。

animation-timing-function 规定动画的速度曲线。默认是 "ease"。

animation-delay         规定动画何时开始。默认是 0。

animation-iteration-count         规定动画被播放的次数。默认是 1。

animation-direction         规定动画是否在下一周期逆向地播放。默认是 "normal"。alternate/reverse/alternate-reverse,分别表示交替/反向/反向交替

animation-play-state         规定动画是否正在运行或暂停。默认是 "running"。

animation-fill-mode         规定对象动画时间之外的状态。

**设置了所有的参数**

|  |  |
| --- | --- |
|  | div  {  animation-name: myfirst;  animation-duration: 5s;  animation-timing-function: linear;  animation-delay: 2s;  animation-iteration-count: infinite;  animation-direction: alternate;  animation-play-state: running;  } |

例子:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | div  {  width:100px;  height:100px;  background:red;  animation:myfirst 5s;  -moz-animation:myfirst 5s; /\* Firefox \*/  -webkit-animation:myfirst 5s; /\* Safari and Chrome \*/  -o-animation:myfirst 5s; /\* Opera \*/  }    @keyframes myfirst  {  0% {background:red;}  25% {background:yellow;}  50% {background:blue;}  100% {background:green;}  }    @-moz-keyframes myfirst /\* Firefox \*/  {  0% {background:red;}  25% {background:yellow;}  50% {background:blue;}  100% {background:green;}  }    @-webkit-keyframes myfirst /\* Safari and Chrome \*/  {  0% {background:red;}  25% {background:yellow;}  50% {background:blue;}  100% {background:green;}  }    @-o-keyframes myfirst /\* Opera \*/  {  0% {background:red;}  25% {background:yellow;}  50% {background:blue;}  100% {background:green;}  } |

**实现卡片滚入效果 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | .transition{          animation: scroll-in 3s alternate;          }  @keyframes scroll-in {  0% {transform: translate(0,1000px) rotate(0deg)}  100% {transform: translate(0) rotate(360deg)}  } |

**跑马灯:**

**JS插入style实现:**

const list = 'document.getElementById("marquee")';

const marquee = document.getElementById("marquee");

const clientWidth = marquee.clientWidth;

marquee.innerHTML = list;

const length = list.length\*2;

const keyframe = '@keyframes marquee {' +

'0% {left:-' + length\*4 +'px}'+

'70% {left:' + clientWidth +'px}' +

'100% {left:' + clientWidth +'px}';

const style = document.createElement('style');

style.innerHTML = keyframe;

marquee.appendChild(style);

marquee.style.animation = list.length/5 + 's marquee linear infinite';

**JS实现:**

const content = document.getElementById('marquee');

content.innerText = 'marquee.style.animation = marquee.style.animation = limarquee.style.animation = limarquee.style.animation = lilist.lenth/5 + \'s marquee linear infinite\'th/5 + \'s marquee linear infinite\'gth/5 + \'s marquee linear infinite\';';

window.onload = function () {

const boxWidth = content.clientWidth;

let textWidth = content.innerText.length \* window.getComputedStyle(content).fontSize.split('px')[0];

console.log(content.innerText);

console.log(textWidth);

console.log(boxWidth);

content.innerHTML += content.innerHTML;

// 更新

if (boxWidth < textWidth) {

// 开始滚动 触发事件

toScrollLeft();

}

};

//实现跑马灯效果

function toScrollLeft(content,totalWidth) {

$(content).html($(content).innerHTML+$(content).innerHTML);

setInterval(function () {

content.scrollLeft++;

// 如果文字长度大于滚动条距离，则递归拖动

if (totalWidth <= content.scrollLeft) {

content.scrollLeft-=totalWidth;

}

},18);

}

**实现菜单动画效果:**

**HTML:**

<ul id="menu1" style="display: block;margin: auto;height: 50px;width: fit-content">

<li class="menu">

<ul>

<li class="title">title</li>

<li class="list">list</li>

<li class="list">list</li>

<li class="list">list</li>

</ul>

</li>

<li class="menu">

<ul>

<li class="title">title</li>

<li class="list">list</li>

<li class="list">list</li>

<li class="list">list</li>

</ul>

</li>

<li class="menu">

<ul>

<li class="title">title</li>

<li class="list">list</li>

<li class="list">list</li>

<li class="list">list</li>

</ul>

</li>

</ul>

<ul id="menu2" style="display: block;margin: 20px auto;height:50px;width: fit-content">

<li class="menu">

<ul>

<li class="title">title</li>

<li class="list">list</li>

<li class="list">list</li>

<li class="list">list</li>

</ul>

</li>

<li class="menu">

<ul>

<li class="title">title</li>

<li class="list">list</li>

<li class="list">list</li>

<li class="list">list</li>

</ul>

</li>

<li class="menu">

<ul>

<li class="title">title</li>

<li class="list">list</li>

<li class="list">list</li>

<li class="list">list</li>

</ul>

</li>

</ul>

**CSS:**

@keyframes move-in-1 {

0% {

transform: translate(100%);

opacity: 0;

}

100% {

transform: translate(0);

opacity: 1;

}

}

@keyframes move-in-2 {

0% {

transform: translate(-100%);

opacity: 0;

}

100% {

transform: translate(0);

opacity: 1;

}

}

ul {

padding: 0;

}

li {

position: relative;

height: 30px;

width: 120px;

line-height: 30px;

color: #fff;

text-align: center;

list-style: none;

background-color: #1E9FFF;

cursor: pointer;

}

li:before {

content: "";

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 2px;

background-color: #fff;

}

.menu {

background-color: #fff;

position: relative;

float: left;

margin: 5px;

}

.menu:hover :nth-child(1n).list {

display: block;

animation: move-in-1 .3s;

}

.menu:hover :nth-child(2n).list {

display: block;

animation: move-in-2 .3s;

}

.list {

position: relative;

display: none;

z-index: 10;

}

/\*第二个menu\*/

@keyframes fade-in {

0% {

opacity: 0;

}

50%{

opacity: 0.7;

}

100% {

opacity: 1;

}

}

#menu2 .menu:hover .list{

display: block;

animation: fade-in .3s ease-in-out;

}