

[笔记][LIKE-H5][21-动画模块]

前端

[笔记][LIKE-H5][21-动画模块]

- 175. 动画模块(掌握)
- 176. 动画模块-其它属性上(理解)
- 177. 动画模块-其它属性下(理解)
- 178. 动画模块-连写(掌握)
- 179. 动画模块-云层动画(理解)
- 180. 动画模块-无限滚动上(掌握)
- 181. 动画模块-无限滚动下(掌握)

175. 动画模块(掌握)

过渡和动画之间的异同

- 不同点：
 - 过渡必须人为触发才会执行动画
 - 动画不需要人为的触发就能执行动画
- 相同点：
 - 过渡和动画都是用来给元素添加动画的
 - 过渡和动画都是系统新增的一些属性
 - 过渡和动画都需要满足三要素才会有效果

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>动画模块</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }
    div{
      width: 100px;
      height: 100px;
      background-color: red;
      /* 1. 告诉系统需要执行哪个动画 */
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```

```

        animation-name: suibianxie;
        /* 3. 告诉系统动画持续的时长 */
        animation-duration: 3s;
    }
    /* 2. 告诉系统我们需要创建一个名称叫做suibianxie的动画 */
    @keyframes suibianxie{
        from{
            margin-left: 0;
        }
        to{
            margin-left: 500px;
        }
    }
</style>
</head>
<body>
<div></div>
</body>
</html>

```

176. 动画模块-其它属性上(理解)

CSS3 动画属性 (Animation)

属性	描述	CSS
<u>@keyframes</u>	规定动画。	3
<u>animation</u>	所有动画属性的简写属性，除了 animation-play-state 属性。	3
<u>animation-name</u>	规定 @keyframes 动画的名称。	3
<u>animation-duration</u>	规定动画完成一个周期所花费的秒或毫秒。	3
<u>animation-timing-function</u>	规定动画的速度曲线。	3
<u>animation-delay</u>	规定动画何时开始。	3
<u>animation-iteration-count</u>	规定动画被播放的次数。	3
<u>animation-direction</u>	规定动画是否在下一周期逆向地播放。	3
<u>animation-play-state</u>	规定动画是否正在运行或暂停。	3
<u>animation-fill-mode</u>	规定对象动画时间之外的状态。	3

`animation-iteration-count` 可以取 `infinite`，无穷次。

`animation-direction: alternate;`

往返动画，假如是走三次，过去，回来，过去，这样就算三次了。

默认是 `normal`

`animation-play-state: running;` 默认取值为 `running`，代表动画正在执行。

还有一个取值 `paused`。

177. 动画模块-其它属性下(理解)

定义动画的第二种方式，使用百分号来定义：

```
@keyframes myRotate {
  0%{
    transform: rotate(10deg);
  }
  50%{
    transform: rotate(50deg);
  }
  100%{
    transform: rotate(70deg);
  }
}
```

动画有状态的：

- 等待状态
- 执行状态
- 结束状态

animation-fill-mode

语法

```
animation-fill-mode : none | forwards | backwards | both;
```

值	描述
none	不改变默认行为。
forwards	当动画完成后，保持最后一个属性值（在最后一个关键帧中定义）。
backwards	在 animation-delay 所指定的一段时间内，在动画显示之前，应用开始属性值（在第一个关键帧中定义）。
both	向前和向后填充模式都被应用。

作用：指定动画等待状态和结束状态的样式。

none：不做任何改变

forwards：让元素结束状态保持动画最后一帧的样式

backwards：让元素等待状态的时候显示状态第一帧的样式

both：结合了 **forwards** 和 **backwards**，让元素等待状态显示元素第一帧的样式，让元素结束状态保持动画最后一帧的样式

企业开发中用的不多，了解即可。

178. 动画模块-连写(掌握)

语法

```
animation: name duration timing-function delay iteration-count direction;
```

值	描述
<u>animation-name</u>	规定需要绑定到选择器的 keyframe 名称。。
<u>animation-duration</u>	规定完成动画所花费的时间，以秒或毫秒计。
<u>animation-timing-function</u>	规定动画的速度曲线。
<u>animation-delay</u>	规定在动画开始之前的延迟。
<u>animation-iteration-count</u>	规定动画应该播放的次数。
<u>animation-direction</u>	规定是否应该轮流反向播放动画。

简写形式

```
animation: name duration;
```

animation-fill-mode 不在这里面

179. 动画模块-云层动画(理解)



背景图片和 `img` 的区别：

- 背景图片不占位置，`img` 占位置
- `img` 权重较高，会被搜索引擎收录

我最后给 `ul` 加了一个 `overflow: hidden;` , 这样可以剪切超出屏幕的部分, 不会有下方的横向滚动条

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>动画模块之云层效果</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }
    ul{
      height: 400px;
      background-color: skyblue;
      margin-top: 100px;
      animation: change 5s linear 0s infinite alternate;
      position: relative;
      overflow: hidden;
    }
    @keyframes change {
      from{
        background-color: skyblue;
      }
      to{
        background-color: black;
      }
    }
    ul li{
      width: 400%;
      height: 100%;
      position: absolute;
      left: 0;
      top: 0;
      list-style: none;
    }
    ul li:nth-child(1){
      background-image: url("img/cloud_one.png");
      animation: one 30s linear 0s infinite alternate;
    }
    ul li:nth-child(2){
      background-image: url("img/cloud_two.png");
      animation: two 30s linear 0s infinite alternate;
    }
    ul li:nth-child(3){
      background-image: url("img/cloud_three.png");
      animation: three 30s linear 0s infinite alternate;
    }
    @keyframes one {
```

```

        from{
            margin-left: 0;
        }
        to{
            /* 移动一个自己的宽度 */
            margin-left: -100%;
        }
    }
    @keyframes two {
        from{
            margin-left: 0;
        }
        to{
            /* 移动一个自己的宽度 */
            margin-left: -200%;
        }
    }
    @keyframes three {
        from{
            margin-left: 0;
        }
        to{
            /* 移动一个自己的宽度 */
            margin-left: -300%;
        }
    }
}
</style>
</head>
<body>
<ul>
    <li></li>
    <li></li>
    <li></li>
</ul>
</body>
</html>

```

180. 动画模块-无限滚动上(掌握)



关键是在后面又多拼接了开头的两张图片，实现了无限滚动。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>动画之无限滚动</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }
    div{
      width: 600px;
      height: 188px;
      border: 1px solid #000;
      margin: 100px auto;
      overflow: hidden;
    }

    ul{
      width: 2000px;
      height: 188px;
      animation: move 10s linear 0s infinite normal;
    }
    ul li{
      float: left;
      list-style: none;
      width: 300px;
      height: 188px;
      box-sizing: border-box;
    }

    @keyframes move {
      from{
        margin-left: 0;
      }
      to{
        margin-left: -1200px;
      }
    }
  </style>
</head>
<body>
  <div>
    <ul>
      <li><img alt="iOS Double Platform banner" data-bbox="116 38 382 196"></li>
      <li><img alt="Android Discipline banner" data-bbox="382 38 730 196"></li>
      <li><img alt="C/C++ banner" data-bbox="730 38 812 196"></li>
      <li><img alt="iOS Double Platform banner" data-bbox="116 38 382 196"></li>
      <li><img alt="Android Discipline banner" data-bbox="382 38 730 196"></li>
      <li><img alt="C/C++ banner" data-bbox="730 38 812 196"></li>
    </ul>
  </div>
</body>
</html>
```

```

</style>
</head>
<body>
  <div>
    <ul>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
    </ul>
  </div>
</body>
</html>

```

181. 动画模块-无限滚动下(掌握)

完善细节：一旦鼠标悬停，停止动画，并给没有悬停的两张图片添加灰色蒙版。

如果元素是半透明的，就能看到它父元素的颜色。

我还添加了蒙版的过渡效果。

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>动画之无限滚动2</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }
    div{
      width: 600px;
      height: 188px;
      border: 1px solid #000;
      margin: 100px auto;
      overflow: hidden;
    }
    ul{
      width: 2000px;
      height: 188px;
      animation: move 10s linear 0s infinite normal;

```



```

        background-color: black;
    }
    ul li{
        float: left;
        list-style: none;
        width: 300px;
        height: 188px;
        box-sizing: border-box;
        transition-property: opacity;
        transition-duration: 0.4s;
    }
    ul:hover{
        animation-play-state: paused;
    }
    ul:hover li{
        opacity: 0.4;
    }
    ul li:hover{
        opacity: 1;
    }
    @keyframes move {
        from{
            margin-left: 0;
        }
        to{
            margin-left: -1200px;
        }
    }
</style>
</head>
<body>
<div>
    <ul>
        <li></li>
        <li></li>
        <li></li>
        <li></li>
        <li></li>
        <li></li>
    </ul>
</div>
</body>
</html>

```