

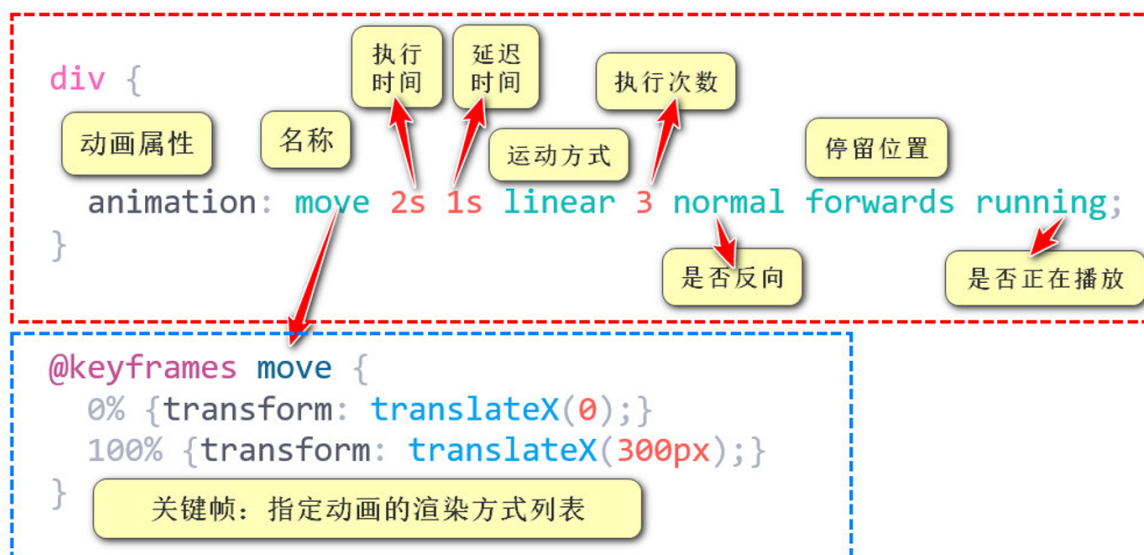
day16

动画

1.animation【9:03】

`animation` 动画，可以使一个元素从一个css样式配置转换到另一个css样式配置。

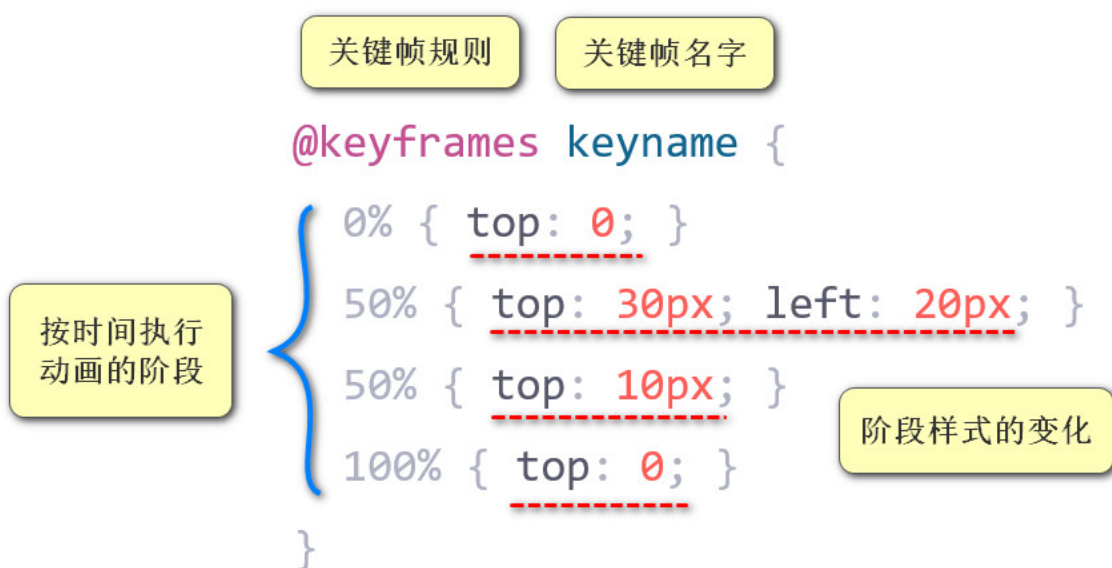
动画包含两组内容：动画属性（包括多种属性）和关键帧



2.关键帧

关键帧 `@keyframes` 这个规则通过在动画序列中定义关键帧来控制css动画序列的步骤。创建关键帧需要一个名字，这个名字最好见名知意，不要以数字开头。

关键帧的语法规则：



练习

```
/* 关键帧 */
@keyframes keyname{
  0%{transform: translateX(0);}
  50%{transform: translateX(300px);}
  100%{transform: translateX(0);}
}
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation: 1s keyname;
}
```

3.动画属性拆解【9:45】

属性名	属性和默认值	描述	常用值
动画名称	animation-name: keyname;	执行关键帧的名字（必写）	自定义
动画执行时间	animation-duration: 0s;	完整执行动画的时间（必写）	时间s
动画延迟时间	animation-delay: 0s;	延迟执行关键帧动画的时间	时间s
动画运动方式	animation-timing-function: ease;	动画运动方式可以写贝塞尔曲线	linear匀速
动画执行次数	animation-iteration-count: 1;	执行动画次数，无单位，数字	infinite无限次
是否反向播放	animation-direction: normal;	按照关键帧的执行顺序或者反向	reverse反向
动画执行后的状态	animation-fill-mode: none;	一组动画执行后元素的位置或状态	forwards停在结束位置
动画播放状态	animation-play-state: running;	默认动画是播放状态，可以通过伪类改变状态	paused暂停

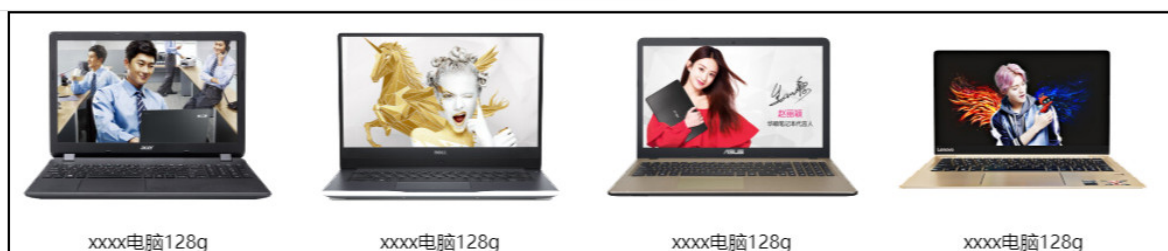
简写

只有执行时间和延迟时间有顺序，执行时间必须写延迟之前。

```
animation:关键帧 动画执行时间 延迟时间 运动方式 次数 反向 停留 播放;
最简写法:
animation:关键帧 动画执行时间;
```

练习

1000px 宽度



四张图填满1000px，父级是1000的2倍，8张图一张占12.5%

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style type="text/css">
      @keyframes move{
        0%{}
        100%{transform: translateX(-1000px);}
      }
      * {
        margin: 0;
        padding: 0;
      }
      .wrap {
        /* 固定宽度 居中 溢出隐藏*/
        width: 1000px;
        border: 2px solid black;
        margin: 0 auto;
        /* 溢出隐藏先不写 */
        overflow: hidden;
      }
      .wrap ul {
        width: 200%;
        background-color: hotpink;
        list-style: none;
        display: flex;
        /* 动画 */
        animation: 12s move infinite linear;
      }
      .wrap ul li {
        /* 临时宽度 */
        width: 12.5%;
        text-align: center;
      }
      .wrap ul li img {
        width: 70%;
      }
      /* 鼠标移入改变某元素执行动画中的某一个属性 */
      .wrap ul:hover {
        animation-play-state: paused;
      }
    </style>
  </head>
  <body>
    <div class="wrap">
      <ul>
        <li><img alt="Meeting" data-bbox="144 124 299 199" data-kind="parent" data-rs="2" /></li>
        <li><img alt="Unicorn" data-bbox="331 124 486 199" data-kind="parent" data-rs="2" /></li>
        <li><img alt="Red Dress" data-bbox="511 124 666 199" data-kind="parent" data-rs="2" /></li>
        <li><img alt="Wings" data-bbox="694 124 849 199" data-kind="parent" data-rs="2" /></li>
      </ul>
    </div>
  </body>
</html>
```

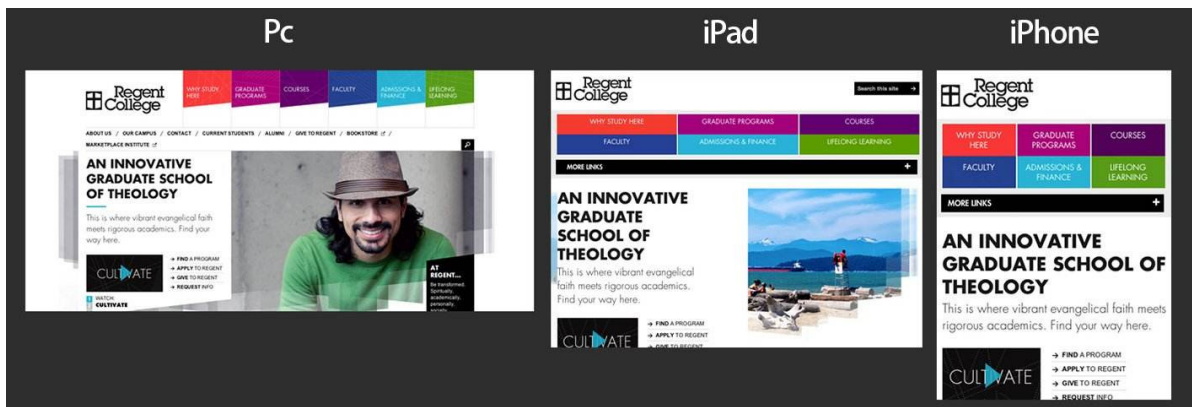
```

</style>
<title></title>
</head>
<body>
  <div class="wrap">
    <ul>
      <li>
        
        <p>联想x3000笔记本</p>
      </li>
      <li>
        
        <p>联想x3000笔记本</p>
      </li>
      <li>
        
        <p>联想x3000笔记本</p>
      </li>
      <li>
        
        <p>联想x3000笔记本</p>
      </li>
      <li>
        
        <p>联想x3000笔记本</p>
      </li>
      <li>
        
        <p>联想x3000笔记本</p>
      </li>
      <li>
        
        <p>联想x3000笔记本</p>
      </li>
      <li>
        
        <p>联想x3000笔记本</p>
      </li>
    </ul>
  </div>
</body>
</html>

```

响应式

1.响应式的概念【14:00】



随着移动终端的兴起，多种规格终端页面需要展示。响应式就是可以让多种终端同时看到相近相同的内容。响应式也叫“自适应”页面。可以根据浏览器设备的不同，而改变不同的布局，文本，图片效果。

特点是使用同一套html结构，根据设备终端尺寸的不同变化，同时使用 @media 媒体查询规则，将不同宽度的css加入到不同的媒体查询中，从而随着页面宽度的变化，改变布局。

2.设置设备宽度【14:13】

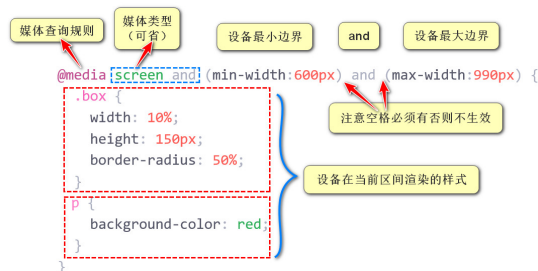
如果没有以下的meta标签，写响应式和移动端是不可以的，一定要加上这段标签到 head 标签里。

如果没有快捷键 meta:vp

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- viewport 视口
- width=device-width 宽度=设备宽度
- initial-scale=1.0 初始化视口不缩放

3.媒体查询【14:35】



```
/* 通用区域 */
* {
  margin: 0;
  padding: 0;
}

/* 100*100 蓝色圆形 */
@media (min-width:600px) and (max-width:900px) {
  body {background-color: red;}
  div {
    width: 100px;
    height: 100px;
    background-color: blue;
    border-radius: 50%;
  }
}

/* 200*200 红色方形 */
```

```

/* 900px以上都是蓝色背景 */
@media (min-width:900px) {
  body{background-color: blue;}
  div {
    width: 200px;
    height: 200px;
    background-color: red;
  }
}
/* 元素消失 */
/* 600px以下都是黄色背景 */
@media (max-width:600px) {
  body {background-color: yellow;}
}

```

4.设计响应式断点

```

/* 统一原则，以最小宽度为断点边界从小往大写 */
/* 手机竖屏（有点特殊先留着） */
@media (max-width:480px) {
  /* ... */
}
/* 手机横置，平板竖置 */
@media (min-width:480px) {
  /* ... */
}
/* 平板和小屏电脑 */
@media (min-width:768px) {
  /* ... */
}
/* 默认屏幕大小 */
@media (min-width:980px) {
  /* ... */
}
/* 大屏显示器 */
@media (min-width:1200px) {
  /* ... */
}

```

5.相对于视口的尺寸【15:53】

`viewport width (vw)`视口的宽度, `viewport height (vh)`视口的高度。

整个视口尺寸宽度=100vw , 1vw可见视口的宽度的1%

整个视口尺寸高度=100vh , 1vh可见视口的高度的1%

```

.box1 {
  width: 50%;
  height: 200px;
  background-color: red;
}
.box2 {
  width: 50vw;
}

```

```
        height: 200px;
        background-color: blue;
    }
    .box3 {
        /* 不是说vh单位只能写给高度 */
        width: 50vh;
        height: 200px;
        background-color: green;
    }
```

练习【16:37】