**纯CSS制作一个小球上台阶的动画**

思路：

1.创建class为loader的父容器，包含5个span，代表5个台阶

2.用flex布局把loader容器水平垂直居中显示

3.显示出5个span用flex布局在水平分割，从下部开始显示

4.用变量把5个span显示出台阶，由低到高的状态

5.使用animation动画效果为台阶增加转换排序方向的效果

6.用伪元素制作两个小球（2个小球交替运动，像1个小球往复运动）

7.增加小球的垂直方向运动的动画效果（向上，指上楼）

8.增加小球在垂直方向运动同时向两侧运动，形成上台阶效果

9.为上台阶改变弹跳上台阶，模拟人上楼

知识点：

1.水平垂直居中

flex布局：父元素添加

display：flex；

align-items：center；

justify-content：center；

2.主轴两端对齐，之间间隔相等

flex布局：父元素添加

display：flex；

justify-content：space-between；

3.交叉轴的终点对齐

flex布局：父元素添加

display：flex；

align-items：flex-end；

4.动画

animation：名称 执行时间 执行次数；

5.显示隐藏

visibility：hidden； --> 隐藏

visibility：visible； --> 显示

步骤：

1.定义 dom，容器中包含 5 个元素，代表 5 个台阶

<div class="loader">

<span></span>

<span></span>

<span></span>

<span></span>

<span></span>

</div>

2.居中显示

body {

margin: 0;

height: 100vh;

display: flex;

align-items: center;

justify-content: center;

background-color: black;

}

3.定义容器尺寸

.loader {

width: 7em;

height: 5em;

font-size: 40px;

}

4.画出 5 个台阶

.loader {

display: flex;

justify-content: space-between;

align-items: flex-end;

}

.loader span {

width: 1em;

height: 1em;

background-color: white;

}

5.用变量让 5 个台阶从低到高排序

.loader span {

height: calc(var(--n) \* 1em);

}

.loader span:nth-child(1) {

--n: 1;

}

.loader span:nth-child(2) {

--n: 2;

}

.loader span:nth-child(3) {

--n: 3;

}

.loader span:nth-child(4) {

--n: 4;

}

.loader span:nth-child(5) {

--n: 5;

}

6.为台阶增加转换排序方向的动画效果

.loader span {

animation: sort 5s infinite;

}

@keyframes sort {

0%, 40%, 100% {

height: calc(var(--n) \* 1em);

}

50%, 90% {

height: calc(5em - (var(--n) - 1) \* 1em);

}

}

7.用伪元素画出 2 个小球

.loader::before,

.loader::after {

content: '';

position: absolute;

width: 1em;

height: 1em;

background-color: white;

border-radius: 50%;

bottom: 1em;

}

.loader::before {

left: 0;

}

.loader::after {

left: 6em;

}

8.增加让小球向上运动的动画效果

.loader::before,

.loader::after {

animation: climbing 5s infinite;

visibility: hidden;

}

.loader::after {

animation-delay: 2.5s;

}

@keyframes climbing {

0% {

bottom: 1em;

visibility: visible;

}

10% {

bottom: 2em;

}

20% {

bottom: 3em;

}

30% {

bottom: 4em;

}

40% {

bottom: 5em;

}

50% {

bottom: 1em;

}

50%, 100% {

visibility: hidden;

}

}

9.在向上运动的同时向两侧运动，形成上台阶的动画效果

.loader::before {

--direction: 1;

}

.loader::after {

--direction: -1;

}

@keyframes climbing {

0% {

bottom: 1em;

left: calc(3em - 2 \* 1.5em \* var(--direction));

visibility: visible;

}

10% {

bottom: 2em;

left: calc(3em - 1 \* 1.5em \* var(--direction));

}

20% {

bottom: 3em;

left: calc(3em - 0 \* 1.5em \* var(--direction));

}

30% {

bottom: 4em;

left: calc(3em + 1 \* 1.5em \* var(--direction));

}

40% {

bottom: 5em;

left: calc(3em + 2 \* 1.5em \* var(--direction));

}

50% {

bottom: 1em;

left: calc(3em + 2 \* 1.5em \* var(--direction));

}

50%, 100% {

visibility: hidden;

}

}

10.为上台阶的动作增加拟人效果

@keyframes climbing {

0% {

bottom: 1em;

left: calc(3em - 2 \* 1.5em \* var(--direction));

visibility: visible;

}

7% {

bottom: calc(2em + 0.3em);

}

10% {

bottom: 2em;

left: calc(3em - 1 \* 1.5em \* var(--direction));

}

17% {

bottom: calc(3em + 0.3em);

}

20% {

bottom: 3em;

left: calc(3em - 0 \* 1.5em \* var(--direction));

}

27% {

bottom: calc(4em + 0.3em);

}

30% {

bottom: 4em;

left: calc(3em + 1 \* 1.5em \* var(--direction));

}

37% {

bottom: calc(5em + 0.3em);

}

40% {

bottom: 5em;

left: calc(3em + 2 \* 1.5em \* var(--direction));

}

50% {

bottom: 1em;

left: calc(3em + 2 \* 1.5em \* var(--direction));

}

50%, 100% {

visibility: hidden;

}

}