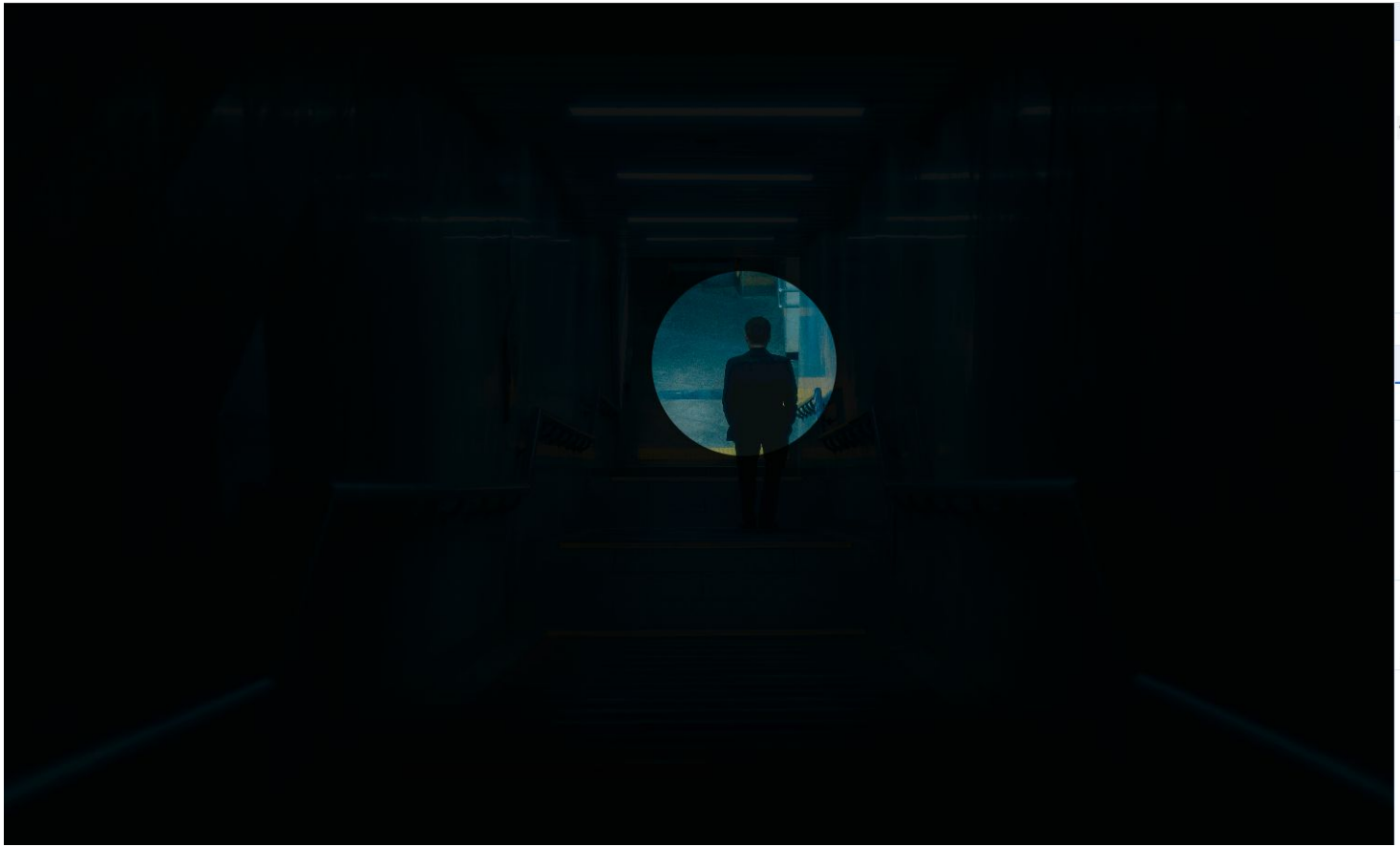
인터랙션스튜디오:
웹인터랙션프로그래밍

웹인터랙션프로그래밍

odododog12@naver.com
010 9215 5512

lms 11주차 파일에서
img 다운 받기

호버 인터랙션


수업준비 - Google Drive
class11 - Google Slides
한 남자가 어두운 복도를 걷고
hover
Google Chrome 고객센터
localhost:63342/24_2학기_인터랙션/html/ckass11/5.html?_ijt=g9rq0u0iu4a2hcrf98rfav2vcc&_ij_reload=RELOAD_ON_SAVE
요소
<!DOCTYPE html>
<html lang="en">
<head></head>
<body cz-shortcut-listen="true"> == $0

<div class="highlight" id="highlight"
style="left: 802px; top: 390px;">
<script></script>
<script></script>
</body>
</html>
html
body
스타일
계산됨
레이아웃
필터
:hov .cls
element.style {
}
body {
5.html?_ijt…D_ON_SAVE:9
margin: 0;
height: 100vh;
box-sizing: border-box;
}
body {
사용자 에이전트 스타일시트
display: block;
margin: 8px;
}
margin
border
padding
1506×911

```html
<div class="content"></div>
<div id="highlight"></div>
```

```js
const highlight = document.getElementById('highlight');
const width = window.innerWidth
const height = window.innerHeight

document.addEventListener('mousemove', (e) => {
    highlight.style.left = `${e.pageX}px`;
    highlight.style.top = `${e.pageY}px`;
});
```

```css
body {
    margin: 0;
    height: 100vh;
    box-sizing: border-box;
    overflow: hidden;
}
.content {
    position: absolute;
    top:0;
    left:0;
    width:100%;
    height:100vh;
    background:url('img/bg.jpg') no-repeat;
    background-size:cover;
    filter: brightness(0.08);
}
#highlight {
    position: absolute;
    width: 200px;
    height: 200px;
    border-radius: 50%;
    backdrop-filter: brightness(8);
    pointer-events: none;
    transform: translate(-50%, -50%);
}
```

투명처리

```
<div class="background"></div>
<div class="container">
   <svg width="100%" height="100%" viewBox="0 0 50 50">
      <defs>
         <mask id="mask">
            <rect width="65%" height="65%" fill="white" />
            <polygon points="15.54 0 19.21 11.29 31.08 11.29 21.47 18.27 25.14 29.55 15.54 22.58 5.93 29.55 9.6 18.27 0 11.29
11.87 11.29 15.54 0" fill="black"/>
         </mask>
      </defs>
      <rect width="100%" height="100%" fill="white" mask="url(#mask)" />
   </svg>
</div>
```

흰색은 보이고 검은색을
뚫림

```css
* {
    margin: 0;
    padding: 0;
}
body {
    overflow: hidden;
}
.background {
    position: absolute;
    width: 100%;
    height: 100%;
    background: url('img/bg.jpg') no-repeat;
    background-size:cover;
}
.container {
    position: absolute;
    width: 100%;
    height: 100%;
}
```

화면전환

```
<div id="wrap">
    <ul>
        <li>1</li>
        <li>2</li>
        <li>3</li>
    </ul>
    <div class="screen"></div>
</div>
```

```css
* {
    margin:0;
    padding:0;
}
#wrap {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    width:100%;
    height:100vh;
    background: red;
}
ul {
    list-style: none;
    display: flex;
    font-size:40px;
}
li {
    padding:40px;
}
```

```css
.screen {
    width:300px;
    height:300px;
    background:#000;
}
.screen.num1 {
    background:pink;
}
.screen.num2 {
    background:blue;
}
.screen.num3 {
    background:greenyellow;
}

@keyframes large {
    30% {
        width:300px;
        height:300px;
    }
    100% {
        width:100%;
        height:100vh;
    }
}
```

```
const lists = document.querySelectorAll('li')
const screen = document.querySelector('.screen')

lists.forEach((list, index) => {

   list.addEventListener('click', () => {

       screen.className = 'screen'
       screen.classList.add(`num${index + 1}`)
       screen.style.animation = 'large 1s forwards'
   })
})
```

canvas

기본 세팅

```
<canvas id="canvas" width="400" height="400"></canvas>
```

```
#canvas {
    border:1px solid #000;
}
```

```
var canvas = document.getElementById("canvas");
var ctx = canvas.getContext("2d");
```

네모 그리기 - 면

```
ctx.fillStyle = 'red'
ctx.fillRect(10, 50, 50, 50);
```

```
var rect = {
   x : 10,
   y : 50,
   width: 50,
   height : 50,
   draw() {
        ctx.fillStyle = 'red'
        ctx.fillRect(this.x, this.y, this.width, this.height);
   }
}

rect.draw();
```

네모 그리기 - 선

```
ctx.strokeStyle = ‘blue’
ctx.strokeRect(80, 50, 50, 50);
```

```
var rect2 = {
   x : 80,
   y : 50,
   width: 50,
   height : 50,
   draw() {
        ctx.strokeStyle  = 'blue'
        ctx.strokeRect(this.x, this.y, this.width, this.height);
   }
}

rect2.draw()
```

네모 그리기 - 지우기

```
ctx.clearRect(40, 50, 50, 50);
```

```
var rect3 = {
   x : 40,
   y : 50,
   width: 50,
   height : 50,
   draw() {
        ctx.clearRect(this.x, this.y, this.width, this.height);
   }
}

rect3.draw()
```

이미지 넣기

```
var img = new Image()
img.src = 'img/dino.png'
```

```
var rect = {
   x : 10,
   y : 50,
   width: 50,
   height : 50,
   draw() {
       ctx.fillStyle = 'red'
       ctx.fillRect(this.x, this.y, this.width, this.height);
       ctx.drawImage(img, 10, 50, 50, 50)
   }
}

img.addEventListener('load', () => {
   rect.draw();
})
```

애니메이션 - 오른쪽으로 이동하기

```
var canvas = document.getElementById("canvas");
var ctx = canvas.getContext("2d");

canvas.width = window.innerWidth - 50
canvas.height = window.innerHeight - 50
```

```
var rect = {
    x : 10,
    y : 50,
    width: 50,
    height : 50,
    draw() {
        ctx.fillStyle = 'red'
        ctx.fillRect(this.x, this.y, this.width, this.height);
    }
}

function move() {
    requestAnimationFrame(move)
    ctx.clearRect(0, 0, canvas.width, canvas.height)
    rect.x++
    rect.draw()

}
move()
```

애니메이션 - 벽에 충돌하면 멈추기

```
function move() {

    if(rect.x >= canvas.width - rect.width) {
        cancelAnimationFrame(move)
    }
    else {
        requestAnimationFrame(move)
        ctx.clearRect(0, 0, canvas.width, canvas.height)
        rect.x++
        rect.draw()
    }
}
move()
```

애니메이션 - 벽에 충돌하면 돌아오기

```
var direction = 1

function move() {

    requestAnimationFrame(move)
    ctx.clearRect(0, 0, canvas.width, canvas.height)

    if(rect.x >= canvas.width - rect.width || rect.x <= 0) {
        direction *= -1
    }

    rect.x += 5 * direction
    rect.draw()

}
move()
```

애니메이션 - x, y 로 충돌하기

```
function move() {

    requestAnimationFrame(move)
    ctx.clearRect(0, 0, canvas.width, canvas.height)

    if(rect.x >= canvas.width - rect.width || rect.x <= 0 || rect.y >= canvas.height - rect.height || rect.y <= 0) {
        direction *= -1
    }

    rect.x += 5 * direction
    rect.y += 5 * direction
    rect.draw()

}
move()
```

애니메이션 - 랜덤으로 충돌하기

```
var rect = {
    x : 10,
    y : 50,
    width: 50,
    height : 50,
    dx : Math.random() * 10 - 5,
    dy : Math.random() * 10 - 5,
    draw() {
        ctx.fillStyle = 'red'
        ctx.fillRect(this.x, this.y, this.width, this.height);
    }
}
```

애니메이션 - 랜덤으로 충돌하기

```
function move() {
    requestAnimationFrame(move)
    ctx.clearRect(0, 0, canvas.width, canvas.height)

    if(rect.x >= canvas.width - rect.width || rect.x <= 0){
        rect.dx *= -1;
    }
    if(rect.y >= canvas.height - rect.height || rect.y <= 0){
        rect.dy *= -1;
    }

    rect.x += rect.dx
    rect.y += rect.dy

    rect.draw()

}
move()
```

본인 작업 START