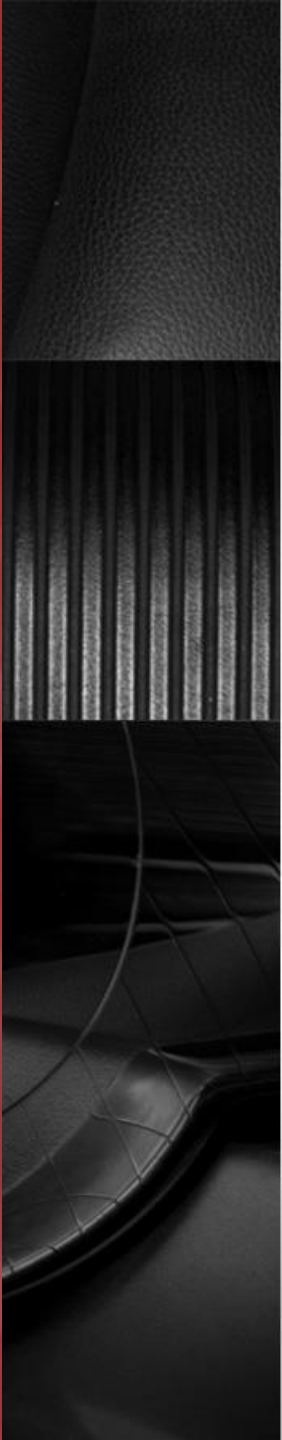


# 스프라이트 이미지

캐릭터 움직임



# 학습 목표

## 1. 캐릭터 이동



## 2. 배경 이동



## 3. 배경 속 캐릭터 이동





# 1. 최초로 실행되는 init 함수 호출 방법

- 방법 1

- `<body onload="init()">`

- 방법 2

- `document.addEventListener('DOMContentLoaded', init);`

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>전체이미지출력 0ld</title>
5   <meta charset="utf-8">
6   <script src="00_전체이미지출력0ld.js"></script>
7   <style type="text/css">
8     canvas{
9       border: 3px solid blue
10    }
11  </style>
12 </head>
13 <body onload="init()">
14   <div id="gameArea">
15     <canvas id="myCanvas" width="1500" height="700">
16       이 브라우저는 HTML5 canvas 태그를 제공하지 않습니다.
17     </canvas>
18   </div>
19 </body>
20 </html>
```

최초 실행되는  
함수 호출 방법 1

```
1 "use strict";
2 var vcanvas, ctx, img;
3
4 function draw() {
5   ctx.drawImage(img, 0, 0, img.width, img.height, 0, 0, img.width, img.height);
6 }
7
8 function init() {
9   vcanvas = document.getElementById("myCanvas");
10  ctx = vcanvas.getContext("2d");
11
12  img = new Image();
13  img.src = "walking1.png";
14
15  window.setTimeout(draw, 500);
16 }
17
```

```

1  <!DOCTYPE html>
2  ▼ <html>
3  ▼ <head>
4      <title>전체이미지출력</title>
5      <meta charset="utf-8">
6      <script src="01_전체이미지출력.js"></script>
7      <style type="text/css">
8  ▼      canvas{
9          border: 3px solid blue
10         }
11     </style>
12 </head>
13 ▼ <body>
14 ▼     <div id="gameArea">
15 ▼         <canvas id="myCanvas" width="1500" height="700">
16             이 브라우저는 HTML5 canvas 태그를 제공하지 않습니다.
17         </canvas>
18     </div>
19 </body>
20 </html>

```

최초 실행되는  
함수 호출 방법 2

```

1  "use strict";
2  var vcanvas, ctx, img;
3
4  ▼ function draw() {
5      ctx.drawImage(img, 0, 0, img.width, img.height, 0, 0, img.width, img.height);
6  }
7
8  ▼ function init() {
9      vcanvas = document.getElementById("myCanvas");
10     ctx = vcanvas.getContext("2d");
11
12     img = new Image();
13     img.src = "walking1.png";
14
15     window.setTimeout(draw, 500);
16 }
17
18 document.addEventListener('DOMContentLoaded', init);
19

```

## 2. 스프라이트 이미지 출력



- `img = new Image();`
- `img.src = "walking1.png";`
- `window.setTimeout(draw, 500);`



- `ctx.drawImage(img, 0, 0, img.width, img.height, 0, 0, img.width, img.height);`

## 2. 스프라이트 이미지 출력 Cont...

- `drawImage(img, sx, sy, swidth, sheight, x, y, width, height);`
  1. `img` : 이미지 객체
  2. `sx` : 스프라이트 이미지 안에서 가져올 이미지 x 좌표
  3. `sy` : 스프라이트 이미지 안에서 가져올 이미지 y 좌표
  4. `swidth` : 가져올 이미지의 가로 길이
  5. `sheight` : 가져올 이미지의 세로 길이
  6. `x` : 가져온 이미지를 출력할 캔버스 상의 x 좌표
  7. `y` : 가져온 이미지를 출력할 캔버스 상의 y 좌표
  8. `width` : 출력할 이미지의 캔버스 상의 가로 길이
  9. `height` : 출력할 이미지의 캔버스 상의 세로 길이

```
1  "use strict";
2  var vcanvas, ctx, img;
3
4  ▼ function draw() {
5      ctx.drawImage(img, 0, 0, img.width, img.height, 0, 0, img.width, img.height);
6  }
7
8  ▼ function init() {
9      vcanvas = document.getElementById("myCanvas");
10     ctx = vcanvas.getContext("2d");
11
12     img = new Image();
13     img.src = "walking1.png";
14
15     window.setTimeout(draw, 500);
16 }
17
18 document.addEventListener('DOMContentLoaded', init);
19
```





### 3. 단순 캐릭터 출력

1. 스프라이트 이미지 내에서의 위치 : 출력할 이미지  $sx, sy$  좌표 확인  
:  $sx, sy$
2. 스프라이트 이미지 내에서의 가로세로 크기 : 추출할 이미지의 가로세로 길이  
:  $swidth, sheight$
3. 캔버스 내에서의 위치 : 가져온 이미지를 캔버스 정 중앙에 출력  
:  $x, y$
4. 캔버스 내에서의 크기 : 어느 정도 크기로 출력할 것인지 확인  
:  $width, height$

- walking1.png 의 전체 이미지의

- width = 1142

- height = 635

- 캐릭터 개수 : 12 x 4

- 캐릭터 width = 95.6

- 캐릭터 height = 158.75



[가로 6번째 세로 3번째 출력]

sx : ?

sy : ?



## 4. 캐릭터 정면 이동

```
"use strict";  
var vcanvas, ctx, img, i = 0;  
var charaWidth = 95.6, charaHeight = 158.75;  
var sx = 0;
```

```
function update() {  
  if (i > 11) {  
    i = 0;  
  } else {  
    sx = i * charaWidth;  
    i += 1;  
  }  
}
```

// 스프라이트 이미지 한 줄의 12번째 캐릭터가 지나면  
// 다시 해당 줄의 처음 캐릭터로 이동  
// 아직 마지막 캐릭터를 지나지 않았다면  
// 출력할 캐릭터의 스프라이트 이미지 내에서 x 좌표 확보  
// 다음 출력을 위해 캐릭터 순번 i를 1 증가

```
function gameLoop() {  
  ctx.clearRect(0, 0, vcanvas.width, vcanvas.height);  
  update();  
  ctx.drawImage(img, sx, 0, charaWidth, charaHeight, 700, 300, charaWidth, charaHeight);  
}
```

```
function init() {  
}
```

## 4.캐릭터 좌측 이동

```
"use strict";  
var vcanvas, ctx, img, i = 0;  
var charaWidth = 95.6, charaHeight = 158.75;  
var sx = 0, sy = charaHeight;
```

```
function update() {  
    if (i > 11) {  
        i = 0;  
    } else {  
        sx = i * charaWidth;  
        i += 1;  
    }  
}
```

**// 스프라이트 이미지 한 줄의 12번째 캐릭터가 지나면  
// 다시 해당 줄의 처음 캐릭터로 이동  
// 아직 마지막 캐릭터를 지나지 않았다면  
// 출력할 캐릭터의 스프라이트 이미지 내에서 x 좌표 확보  
// 다음 출력을 위해 캐릭터 순번 i를 1 증가**

```
function gameLoop() {  
    ctx.clearRect(0, 0, vcanvas.width, vcanvas.height);  
    update();  
    ctx.drawImage(img, sx, sy, charaWidth, charaHeight, 700, 300, charaWidth, charaHeight);  
}
```

```
function init() {  
}
```



캐릭터 정면 이동 완성 : 정면이동.html & 정면이동.js

캐릭터 좌측 이동 완성 : 좌측이동.html & 좌측이동.js

캐릭터 우측 이동 완성 : 우측이동.html & 우측이동.js

캐릭터 후방 이동 완성 : 후방이동.html & 후방이동.js

## 인클래스 실습



키를 이용한 캐릭터 4방 이동

4방이동.html & 4방이동.js

# 인클래스 과제

