BS Study S67

Implementing DrawingML-based Shape Editor with ES6

S67 WARNING

본 스터디는 매우 폭력적, 선정적 개드립을 함유하고 있으며, 만 99세 미만의 어른이들에게 매우 유해합니다.

본 스터디에서 등장하는 ES6 코드는 고도로 연출된 판타지이므로, 회사에서 따라하실 때는 반드시 파트너의 동의를 구하십시오.

연사도 ES6를 잘 모르므로, 스터디 도중 곤란한 질문을 삼가해 주십시오. 페북 그룹이나 슬랙을 이용하시면 '언젠가는' 찾아서 답변해 줍니다.


Previously on S67#4...

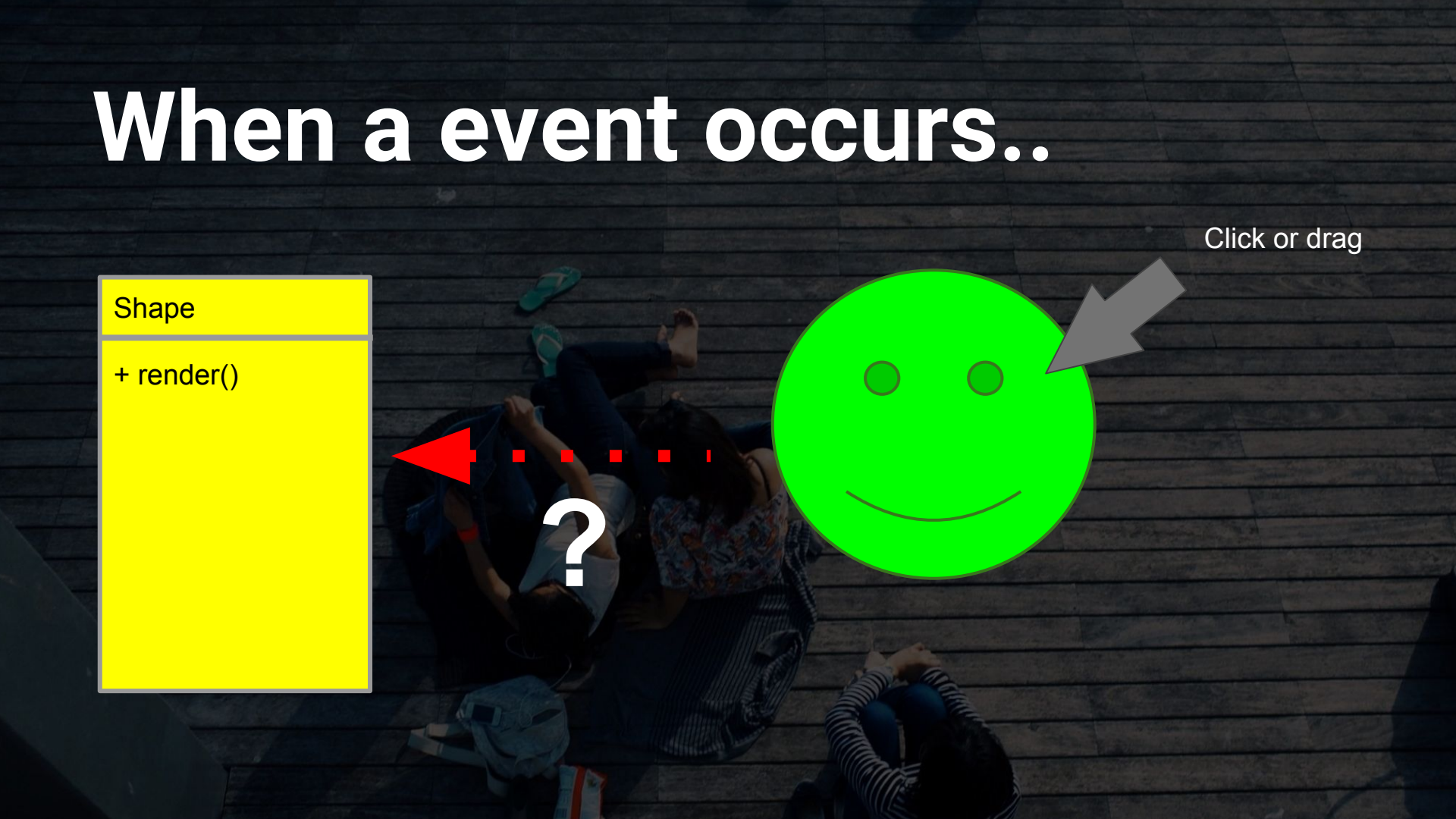
Previously on S67...

- Shape Drawing Panel -> Epic Fail (orz)
- Promise gives you a control over asynchronous callbacks

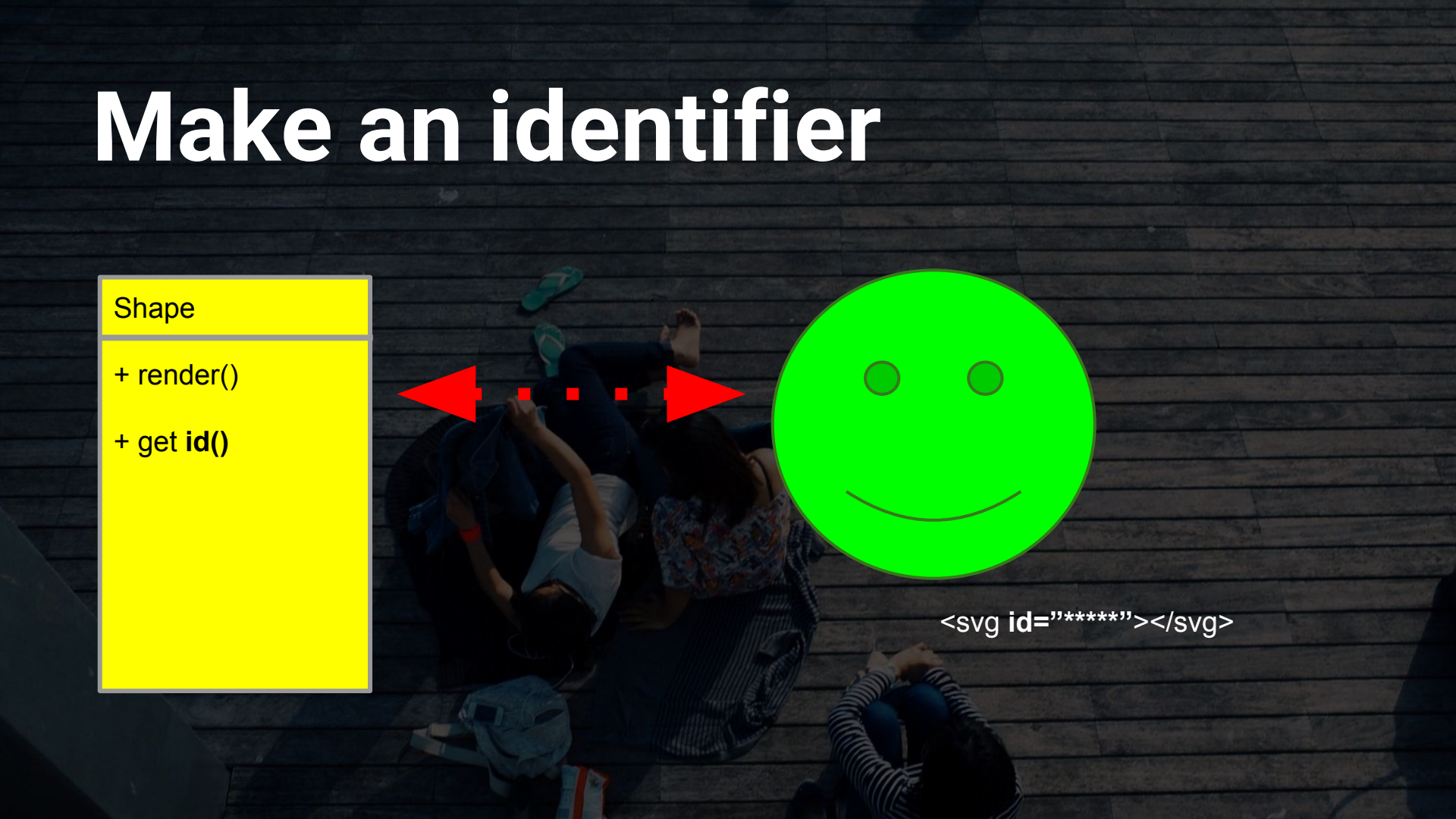

View-Model binding

Model > View

When a event occurs..

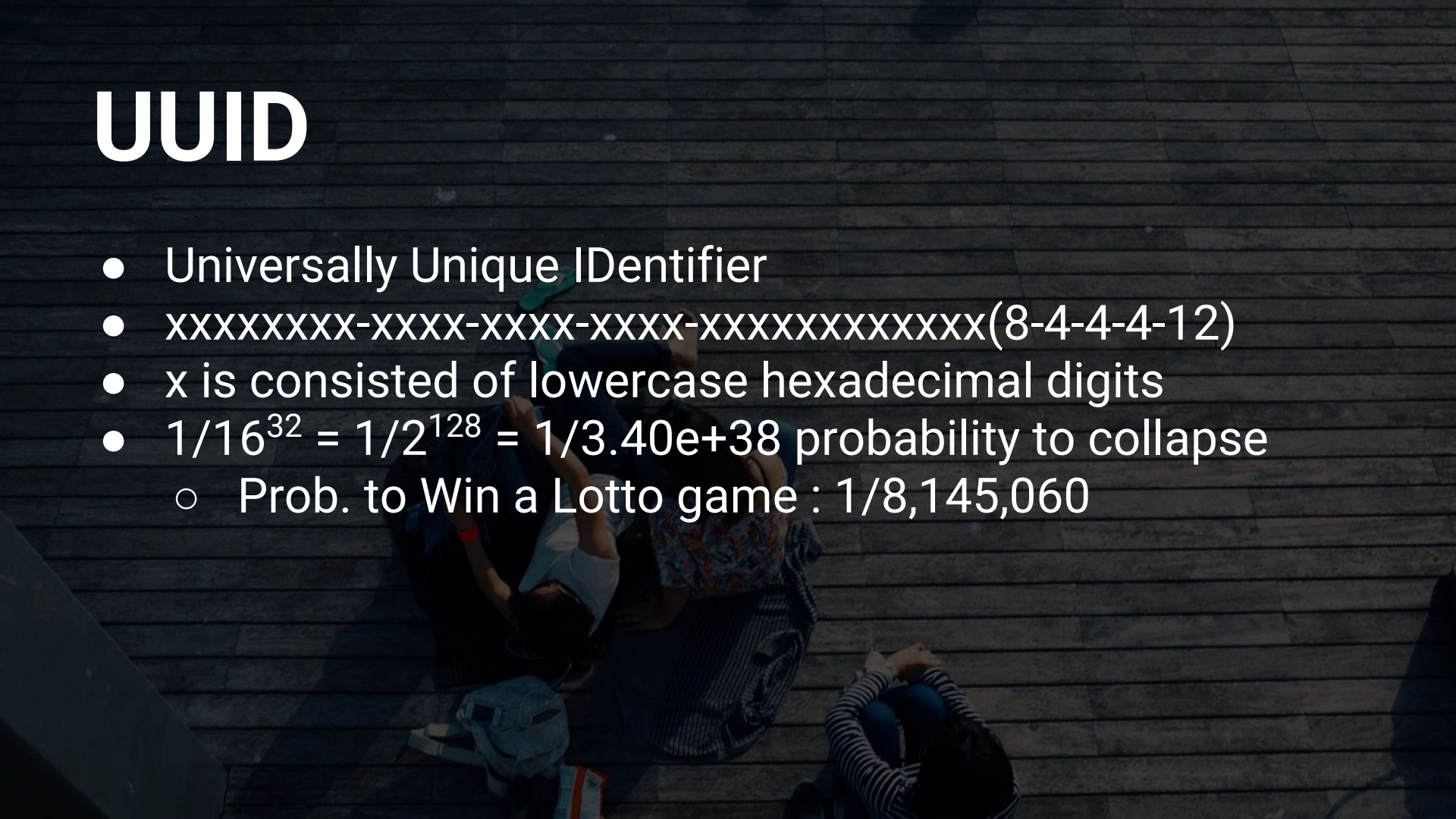


Make an identifier



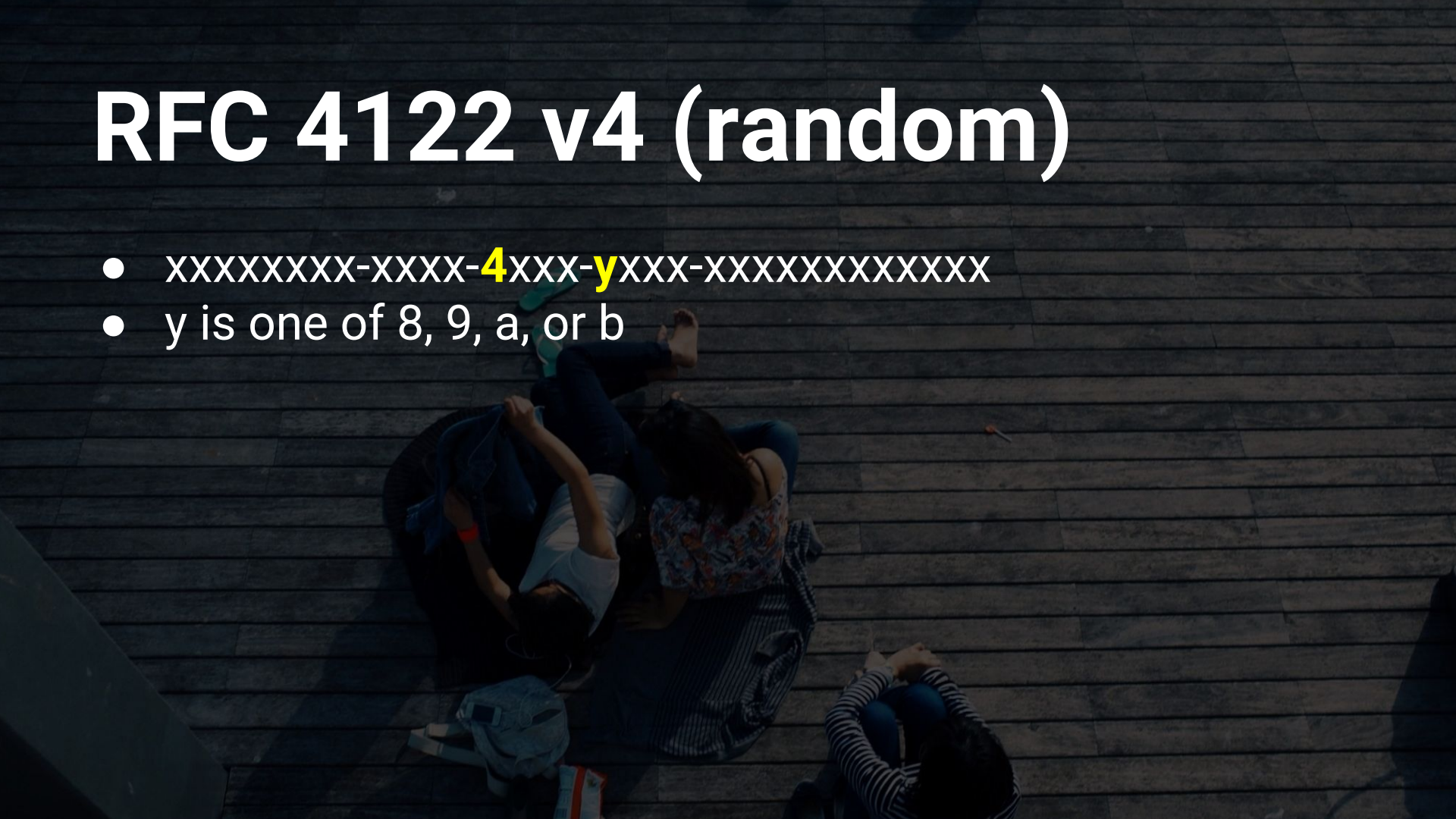
UUID

- Universally Unique IDentifier
- xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx(8-4-4-4-12)
- x is consisted of lowercase hexadecimal digits
- $1/16^{32} = 1/2^{128} = 1/3.40e+38$ probability to collapse
  - Prob. to Win a Lotto game : 1/8,145,060



RFC 4122 v4 (random)

- xxxxxxxx-xxxx-**4**xxx-**y**xxx-xxxxxxxxxxxx
- y is one of 8, 9, a, or b



RFC 4122 v4 (random)

```
function generateUUID() {
    let d = new Date().getTime();
    const uuid = 'xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'.replace(/[xy]/g, c => {
        const r = (d + Math.random()*16)%16 | 0;
        d = Math.floor(d/16);
        return (c=='x'? r : r&0x3|0x8).toString(16);
    });
    return uuid;
};
```

Hex	Bin
8	0100
9	0101
a	0110
b	0111

Simplify, huh?

```
function generateUUID() {
    return 'xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'.replace(/[xy]/g, c => {
        const r = Math.random()*16%16 | 0;
        return (c=='x'? r : r&0x3|0x8).toString(16);
    });
};
```


Shape handling

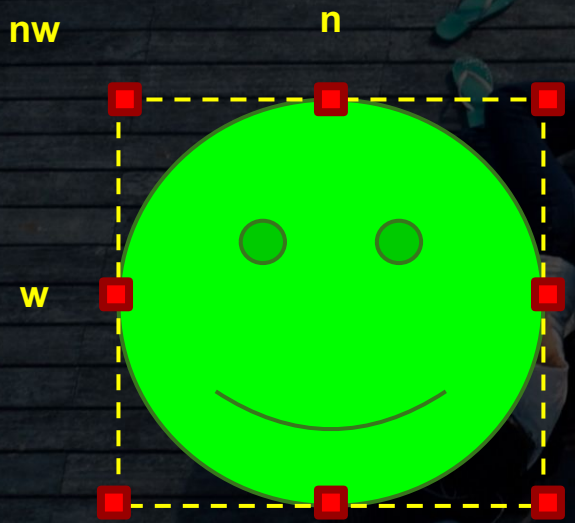
Drag&drop event handler

```
let mousedown = e => {
    let startX = e.pageX, startY = e.pageY,
        mousemove = e => {
            let dX = e.pageX - offsetX, dY = e.pageY - offsetY;
        }, mouseup = e => {
            window.removeEventListener('mousemove', mousemove);
            window.removeEventListener('mouseup', mouseup);
        }
    window.addEventListener('mousemove', mousemove);
    window.addEventListener('mouseup', mouseup);
}
```

Moving Shape


```
shape.l += dX;
shape.t += dY;
```

Resizing Shape

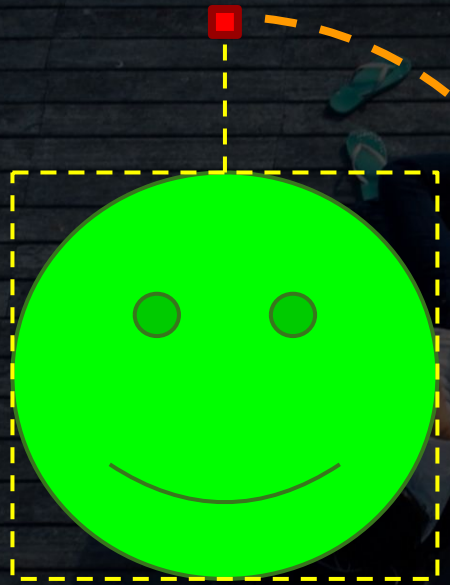


Sometimes, moving shape is required.

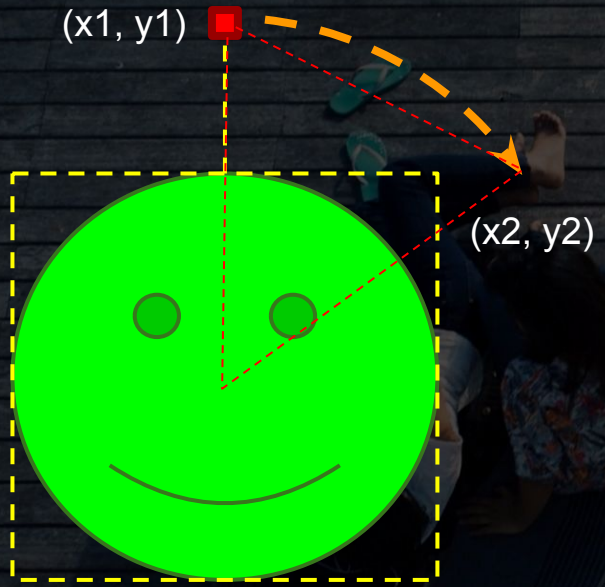
Resizing Shape


```
let direction, dX, dY;

// blahblah

let revX = direction.indexOf('w')>=0,
    revY = direction.indexOf('n')>=0;

shape.w += revX? -dX : dX;
shape.h += revY? -dY : dY;
shape.l += rexX? dX : 0;
shape.t += revY? dY : 0;
```

Rotating Shape



Angle Detection




```
let x1, y1, x2, y2,

cx = shape.l + shape.w/2,
cy = shape.t + shape.t/2,

stAng = Math.atan2(x1-cx, y1-cy),
endAng = Math.atan2(x2-cx, y2-cy);

shape.rotation += stEnd - endAng;
```