keydown and keyup events

event.key and event.keyCode properties

set CSS

random hex color

keyboard events

The pressing of a key is a **keydown** event The release of a key is a **keyup** event

Like any event, a keyboard event can call a function. The event target (thing that fires the event) is the document, so the syntax is: **document.addEventListener('keydown', function)** An if-statement can check the **event.keyCode** to know which key was pressed.

The key **event** has **key** and **keyCode** properties:

The q **event.key** is **q**. The q **event.keyCode** is **81**.

The left arrow **event.key** is **ArrowLeft**. The left arrow **event.keyCode** is **37**.

In this activity, we will set up several keys to do different things, most of which involve setting CSS:

random body background color on 'c' keydown

When the **c** or **Enter** key is pressed, our function will set the body background to a random color.

toggle dark mode on 'd' key press

When the **d** key is pressed, our function will toggle the container between dark mode and light mode.

toggle font on 'f' key press

When the **f** key is pressed, our function will toggle the font between *serif* and *sans-serif*.

generate random 4-digit PIN on 'p' keydown

When the **p** or **n** key is pressed, our function will generate and output a random 4-digit PIN number.

move space ship to the left and right

When the left or right arrow is pressed, our function will move the space ship 10 pixels in that direction. The numeric codes for the left and right arrows are **37** and **39**, respectively.

There are four DOM elements we need:

- **container** div that holds instructions and some output divs:
- **rand-pin-box** div for outputting the **key** and **keyCode** of the pressed key
- **key-box** div for outputting hold a random number
- an image of a space ship

1. Get the DOM elments:

```
const container = document.querySelector('.container');
const keyBox = document.getElementById('key-box');
const spaceShip = document.getElementById('space-ship');
const randPinBox = document.getElementById('rand-pin-box');
```

2. Set the left position of the space ship to be approximately in the middle of the screen. Also set a var for the speed of the ship:

```
let leftPos = window.innerWidth / 2 - 150;
spaceShip.style.left = leftPos + 'px';
let shipSpeed = 20;
```

3. Set booleans to keep track of font (serif or not) and dark mode:

```
let serif = true;
let dark = false;
```

4. Have the document listen for **keyup** and call an inline anonymous function which outputs the **key** and its **keyCode**:

```
document.addEventListener('keyup', function(event) {
    keyBox.textContent = `Key: ${event.key}
    Code: ${event.keyCode}`;
});
```

5. Have document listen for **keydown** event and call the **respondToKeyPress**. The function checks if the key is **n**, **p**, **d**, **l**, **c**, **Enter** or the left or right arrow. If one of these keys was pressed, the specified action takes place.

```
document.addEventListener('keydown', respondToKeyPress);
```

6. Define the function **respondToKeyPress**, which runs on **keydown** (press). All functions take **event** object as its default argument, but pass in **event** explicitly, so that we can access its properties:

```
function respondToKeyPress(event) {
}
```

7. Inside the function, set up a scaffoling of a big if-else if block to check for each key: c, d, f, p, Enter (13), Left (37) and Right (39) arrows:

```
function respondToKeyPress(event) {

    if(event.key === 'c' || event.key === 'Enter') {

    } else if(event.key === 'd' ) {

    } else if(event.key === 'f') {

    } else if(event.key === 'p' || event.key === 'n') {

    } else if(event.keyCode === 37) { // ArrowLeft

    } else if(event.keyCode === 39) { // ArrowRight

    }

}
```

Next, we will use conditinal logic to handle what to do (if anything) when a key is pressed:

8. If the **c** key is pressed, call the **makeRandHexColor()** function that generates and returns a random hexadecimal color. Save the color to a variable, and use that to set the background color:

```
function respondToKeyPress(event) {

    if(event.key === 'c'  || event.key === 'Enter') {
        let randHex = makeRandHexColor();
        document.body.style.backgroundColor = randHex;
    }

} // end function
```

9. If the **d** key is pressed, toggle between dark and light mode by adding-removing their styles. Toggle the boolean each time, so that the behavior also toggles.

```
    } else if (event.key === 'd') {
        if(!dark) {
            container.classList.remove('light-mode');
            container.classList.add('dark-mode');
            dark = true;
```

```
            } else {
                container.classList.remove('dark-mode');
                container.classList.add('light-mode');
                dark = false;
            }
        }

    } // end function
```

10. If the **f** key is pressed, toggle between serif and sans-serif font. Toggle the boolean each time, so that the behavior also toggles.

```
        } else if (event.key === 'f') {
            if(serif) {
                document.body.style.fontFamily = "sans-serif";
                serif = false;
            } else {
                document.body.style.fontFamily = "serif";
                serif = true;
            }
        }

    } // end function
```

11. If the **p** or **n** key is pressed, generate a random 4-digit number from 0-9999. If the number is less than 4 digits, add leading zero(es) by means of if-statements with ranges; utput the random pin:

```
        } else if (event.key === 'p' || event.key === 'n') {

            let r = Math.floor(Math.random() * 10000);

            // add leading zero(es), as needed
            if(r < 1000 && r > 99) r = '0' + r; // 3 digits
            if(r < 100 && r > 9) r = '00' + r; // 2 digits
            if(r < 10) r = '000' + r; // 1 digit
            if(r === 0) r = '0000';

            randPinBox.textContent = 'PIN:\n' + r;
        }

    } // end function
```

12. If the **ArrowLeft (37)** key is pressed, and if the space ship is not all the way to the left, move the ship **shipSpeed** pixels to the left:

```
        } else if (event.keyCode === 37) {
            if(leftPos > 0) {
```

```
                leftPos -= shipSpeed;
                spaceShip.style.left = leftPos + 'px';
            }
        }

    } // end function
```

13. If the **ArrowRight (39)** key is pressed, and if the space ship is not all the way to the right, move the ship **shipSpeed** pixels to the right:

```
        } else if (event.keyCode === 39) {
            if(leftPos < window.innerWidth - 300) {
                leftPos += shipSpeed;
                spaceShip.style.left = leftPos + 'px';
            }
        }

    } // end function
```

14. If the spaceship doesn't move, try switching from event.keyCode to event.key:

```
        } else if (event.key === 'ArrowLeft') {
            if(leftPos > 0) {
                leftPos -= shipSpeed;
                spaceShip.style.left = leftPos + 'px';
        } else if (event.key === 'ArrowRight') {
            if(leftPos < window.innerWidth - 300) {
                leftPos += shipSpeed;
                spaceShip.style.left = leftPos + 'px';
            }
        }

    } // end function
```

Random Hex Color

It is possible to generate a random hexadecimal color. Here's the concept and procedure:

- A base-10 number consists of the digits 0-9
- A base-16 string uses the characters 0-9, A-F
- A hexadecimal color is a 6-character, base-16 string
- The **toString()** method converts a number to a string
- **toString(16)** converts a base-10 number to a base-16 string
- There exist 16,777216 (256 ** 3) colors in the RGB spectrum
- Calling **toString(16)** on an integer in the RGB range from 0-16777215 returns a hexadecimal color value
- Putting '#' before the hex value comletes the hex color

This function makes and returns a random hex color. It is called from within the **respondToKeyPress** function when the **c** key is pressed.

```
    function makeRandHexColor() {

        // A. Generate a random 16-digit float from 0-1
        let rando = Math.random();

        // B. Multiply the random float by 16,777,216 (RGB range)
        rando = rando * 16777216;

        // C. round down the random number:
        rando = Math.floor(rando);

        // D. convert the random number to a base-16 hexadecimal string
        rando = rando.toString(16);

        // E. add the # to complete the hexadecimal color
        rando = '#' + hex;

        // F. return the hexadecimal color value
        return hex;

    }
```

END: Lesson 03.08

NEXT: Lesson 03.09