기초 웹

22 Winter CNU 기초 스터디

21 남정연
21 박준서

Javascript Functions

```
function draw() {
  ctx.clearRect(0,0,WIDTH,HEIGHT);
  for (let i = 0; i < 100; i++) {
    ctx.beginPath();
    ctx.fillStyle = 'rgba(255,0,0,0.5)';
    ctx.arc(random(WIDTH), random(HEIGHT), random(50), 0, 2 * Math.PI);
    ctx.fill();
  }
}
```

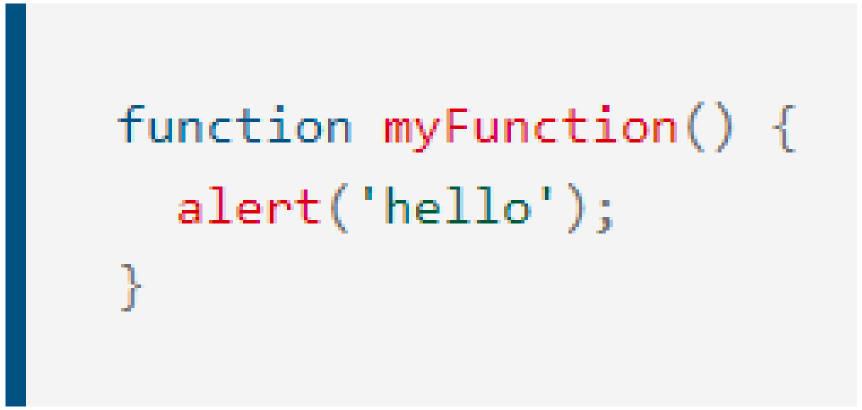
```
draw();
```

Function Parameters

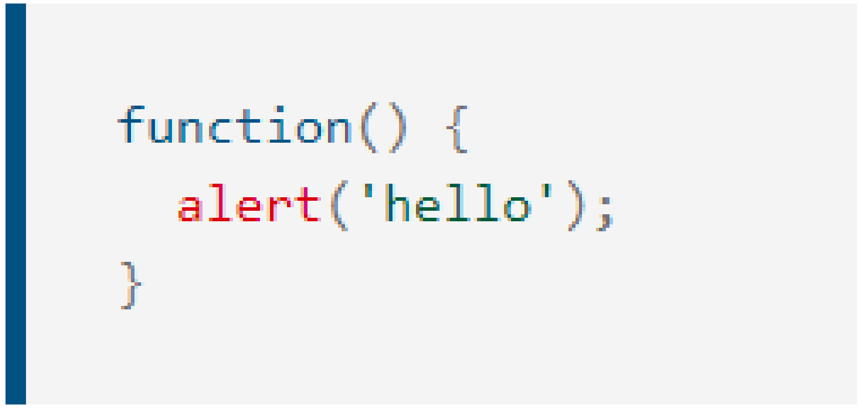
```javascript
const myArray = ['I', 'love', 'chocolate', 'frogs'];
const madeAString = myArray.join(' ');
console.log(madeAString);
// returns 'I love chocolate frogs'

const madeAnotherString = myArray.join();
console.log(madeAnotherString);
// returns 'I,love,chocolate,frogs'
```

```javascript
function hello(name='Chris') {
  console.log(`Hello ${name}!`);
}

hello('Ari'); // Hello Ari!
hello();      // Hello Chris!
```

Anonymous Function



```
function myFunction() {
  alert('hello');
}
```



```
function() {
  alert('hello');
}
```

Anonymous Function

```
function logKey(event) {
  console.log(`You pressed "${event.key}".`);
}

textBox.addEventListener('keydown', logKey);
```

```
textBox.addEventListener('keydown', function(event) {
  console.log(`You pressed "${event.key}".`);
});
```

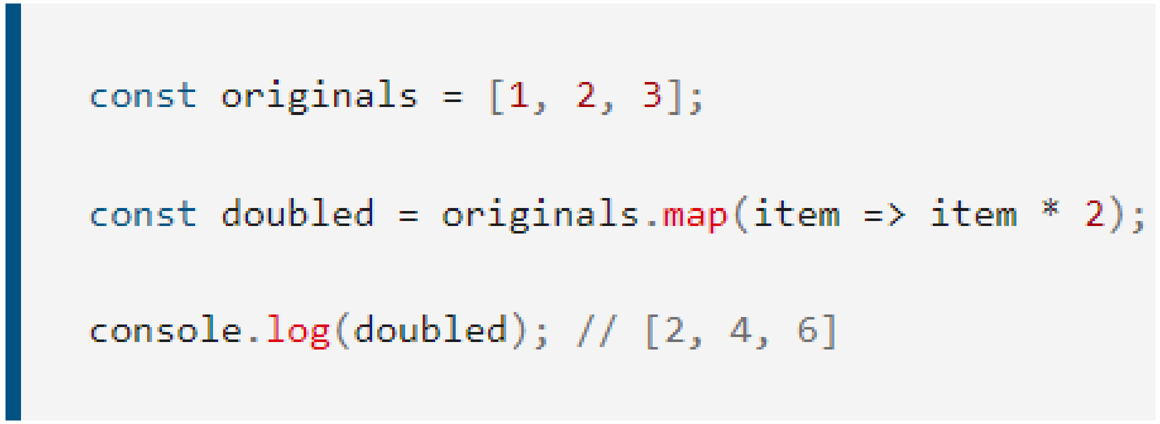
Arrow Function

```
textBox.addEventListener('keydown', (event) => {
  console.log(`You pressed "${event.key}".`);
});
```

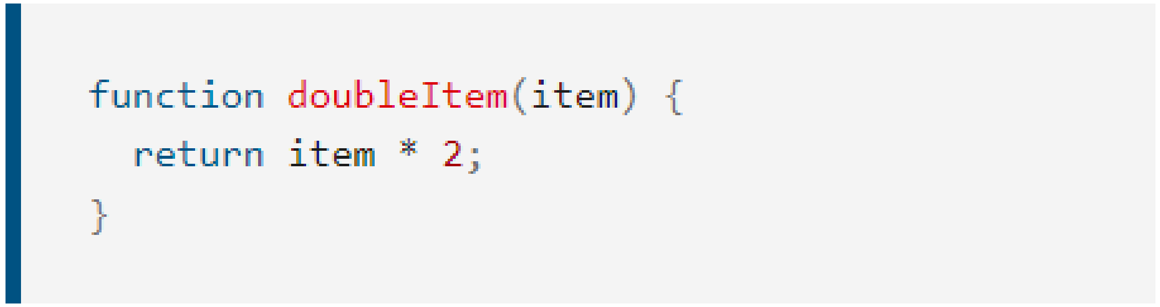
```
textBox.addEventListener('keydown', (event) => console.log(`You pressed "${event.key}".`));
```

```
textBox.addEventListener('keydown', event => console.log(`You pressed "${event.key}".`));
```

Arrow Function



```
const originals = [1, 2, 3];

const doubled = originals.map(item => item * 2);

console.log(doubled); // [2, 4, 6]
```



```
function doubleItem(item) {
  return item * 2;
}
```

Arrow Function

```
const originals = [1, 2, 3];

const doubled = originals.map(item => item * 2);

console.log(doubled); // [2, 4, 6]
```

```
function doubleItem(item) {
  return item * 2;
}
```

Javascript Events - addEventListener

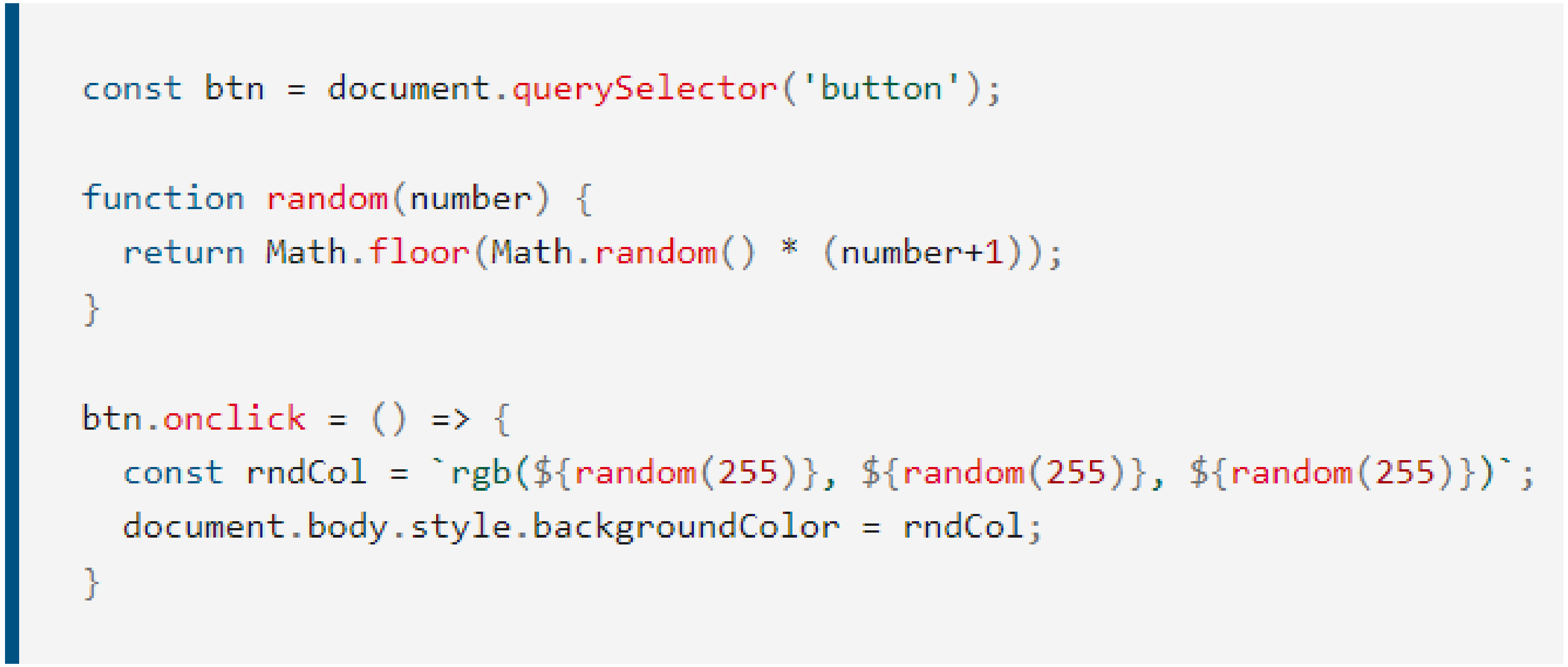
```html
<button>Change color</button>
```

```javascript
const btn = document.querySelector('button');

function random(number) {
  return Math.floor(Math.random() * (number+1));
}

btn.addEventListener('click', () => {
  const rndCol = `rgb(${random(255)}, ${random(255)}, ${random(255)})`;
  document.body.style.backgroundColor = rndCol;
});
```

Javascript Events - addEventListener

```javascript
const btn = document.querySelector('button');

function random(number) {
  return Math.floor(Math.random() * (number+1));
}

function changeBackground() {
  const rndCol = `rgb(${random(255)}, ${random(255)}, ${random(255)})`;
  document.body.style.backgroundColor = rndCol;
}

btn.addEventListener('click', changeBackground);
```

Javascript Events - Event Property



```
const btn = document.querySelector('button');

function random(number) {
  return Math.floor(Math.random() * (number+1));
}

btn.onclick = () => {
  const rndCol = `rgb(${random(255)}, ${random(255)}, ${random(255)})`;
  document.body.style.backgroundColor = rndCol;
}
```

Javascript Events - Event Property

```
const btn = document.querySelector('button');

function random(number) {
  return Math.floor(Math.random() * (number+1));
}

function bgChange() {
  const rndCol = `rgb(${random(255)}, ${random(255)}, ${random(255)})`;
  document.body.style.backgroundColor = rndCol;
}

btn.onclick = bgChange;
```

addEventListener vs Event Property

```
element.addEventListener('click', function1);
element.addEventListener('click', function2);
```

```
element.onclick = function1;
element.onclick = function2;
```

❌

Javascript Events - Inline Handler

```html
<button onclick="bgChange()">Press me</button>
```

```javascript
function bgChange() {
  const rndCol = `rgb(${random(255)}, ${random(255)}, ${random(255)})`;
  document.body.style.backgroundColor = rndCol;
}
```

```html
<button onclick="alert('Hello, this is my old-fashioned event handler!');">Press me</button>
```

Event Objects

```javascript
const btn = document.querySelector('button');

function random(number) {
  return Math.floor(Math.random() * (number+1));
}

function bgChange(e) {
  const rndCol = `rgb(${random(255)}, ${random(255)}, ${random(255)})`;
  e.target.style.backgroundColor = rndCol;
  console.log(e);
}

btn.addEventListener('click', bgChange);
```

Event Objects

```html
<input id="textBox" type="text"></input>
<div id="output"></div>
```

```javascript
const textBox = document.querySelector("#textBox");
const output = document.querySelector("#output");
textBox.addEventListener('keydown', event => output.textContent = `You pressed "${event.key}".`);
```

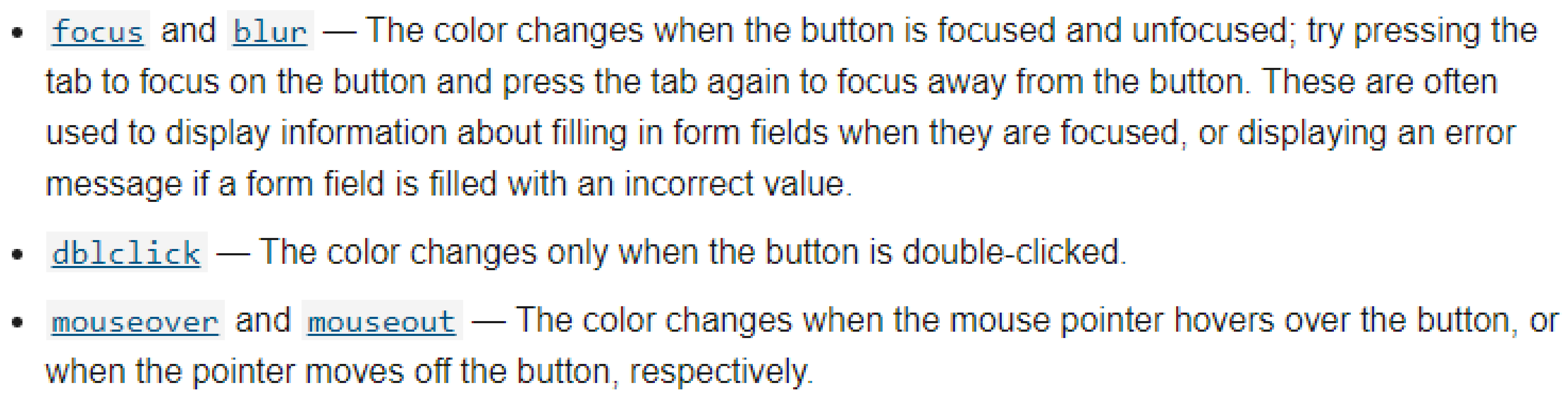
Event Objects

```html
<div id="container">
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
  <div class="tile"></div>
</div>
```

```javascript
function random(number) {
  return Math.floor(Math.random()*number);
}

function bgChange() {
  const rndCol = `rgb(${random(255)}, ${random(255)}, ${random(255)})`;
  return rndCol;
}

const container = document.querySelector('#container');

container.addEventListener('click', event => event.target.style.backgroundColor = bgChange());
```

Note: In this example we're using `event.target` to get the element that was the target of the event (that is, the innermost element). If we wanted to access the element that fired this event (in this case the container) we could use `event.currentTarget`.

Javascript Events



- `focus` and `blur` — The color changes when the button is focused and unfocused; try pressing the tab to focus on the button and press the tab again to focus away from the button. These are often used to display information about filling in form fields when they are focused, or displaying an error message if a form field is filled with an incorrect value.
- `dblclick` — The color changes only when the button is double-clicked.
- `mouseover` and `mouseout` — The color changes when the mouse pointer hovers over the button, or when the pointer moves off the button, respectively.

Removing Event Listener

```
btn.removeEventListener('click', changeBackground);
```

```
const controller = new AbortController();

btn.addEventListener('click', () => {
  const rndCol = `rgb(${random(255)}, ${random(255)}, ${random(255)})`;
  document.body.style.backgroundColor = rndCol;
}, { signal: controller.signal }); // pass an AbortSignal to this handler
```

```
controller.abort(); // removes any/all event handlers associated with this controller
```

Preventing Default Behavior

```html
<form>
  <div>
    <label for="fname">First name: </label>
    <input id="fname" type="text">
  </div>
  <div>
    <label for="lname">Last name: </label>
    <input id="lname" type="text">
  </div>
  <div>
     <input id="submit" type="submit">
  </div>
</form>
<p></p>
```

```javascript
const form = document.querySelector('form');
const fname = document.getElementById('fname');
const lname = document.getElementById('lname');
const para = document.querySelector('p');

form.addEventListener('submit', e => {
  if (fname.value === '' || lname.value === '') {
    e.preventDefault();
    para.textContent = 'You need to fill in both names!';
  }
});
```

Event Bubbling

```html
<body>
  <div id="container">
    <button>Click me!</button>
  </div>
  <pre id="output"></pre>
</body>
```

```javascript
const output = document.querySelector('#output');
function handleClick(e) {
  output.textContent += `You clicked on a ${e.currentTarget.tagName} element\n`;
}

const container = document.querySelector('#container');
const button = document.querySelector('button');

document.body.addEventListener('click', handleClick);
container.addEventListener('click', handleClick);
button.addEventListener('click', handleClick);
```

Event Bubbling

```html
<button>Display video</button>

<div class="hidden">
  <video>
    <source src="https://raw.gith
    <source src="https://raw.gith
    <p>Your browser doesn't suppo
  </video>
</div>
```

```javascript
const btn = document.querySelector('button');
const videoBox = document.querySelector('div');

function displayVideo() {
  if (videoBox.getAttribute('class') === 'hidden') {
    videoBox.setAttribute('class','showing');
  }
}

btn.addEventListener('click', displayVideo);
```

```javascript
videoBox.addEventListener('click', () => videoBox.setAttribute('class', 'hidden'));

const video = document.querySelector('video');

video.addEventListener('click', () => video.play());
```

Event Bubbling

```html
<button>Display video</button>

<div class="hidden">
  <video>
    <source src="https://raw.gith
    <source src="https://raw.gith
    <p>Your browser doesn't suppo
  </video>
</div>
```

```js
const btn = document.querySelector('button');
const videoBox = document.querySelector('div');

function displayVideo() {
  if (videoBox.getAttribute('class') === 'hidden') {
    videoBox.setAttribute('class','showing');
  }
}

btn.addEventListener('click', displayVideo);
```

```js
videoBox.addEventListener('click', () => videoBox.setAttribute('class', 'hidden'));

const video = document.querySelector('video');

video.addEventListener('click', e => {
  e.stopPropagation();
  video.play();
});
```

Event Bubbling

```html
<button>Display video</button>

<div class="hidden">
  <video>
    <source src="https://raw.gith
    <source src="https://raw.gith
    <p>Your browser doesn't suppo
  </video>
</div>
```

```javascript
const btn = document.querySelector('button');
const videoBox = document.querySelector('div');

function displayVideo() {
  if (videoBox.getAttribute('class') === 'hidden') {
    videoBox.setAttribute('class','showing');
  }
}

btn.addEventListener('click', displayVideo);
```

```javascript
videoBox.addEventListener('click', () => videoBox.setAttribute('class', 'hidden'));

const video = document.querySelector('video');

video.addEventListener('click', e => {
  e.stopPropagation();
  video.play();
});
```