Inline event handlers — don't use these

You might also see a pattern like this in your code:

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

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

The earliest method of registering event handlers found on the Web involved *event handler HTML attributes* (or *inline event handlers*) like the one shown above — the attribute value is literally the JavaScript code you want to run when the event occurs. The above example invokes a function defined inside a `<script>` element on the same page, but you could also insert JavaScript directly inside the attribute, for example:

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

You can find HTML attribute equivalents for many of the event handler properties; however, you shouldn't use these — they are considered bad practice. It might seem easy to use an event handler attribute if you are doing something really quick, but they quickly become unmanageable and inefficient.

For a start, it is not a good idea to mix up your HTML and your JavaScript, as it becomes hard to read. Keeping your JavaScript separate is a good practice, and if it is in a separate file you can apply it to multiple HTML documents.

Even in a single file, inline event handlers are not a good idea. One button is OK, but what if you had 100 buttons? You'd have to add 100 attributes to the file; it would quickly turn into a maintenance nightmare. With JavaScript, you could easily add an event handler function to all the buttons on the page no matter how many there were, using something like this:

```
const buttons = document.querySelectorAll('button');

for (const button of buttons) {
  button.addEventListener('click', bgChange);
}
```

Finally, many common server configurations will disallow inline JavaScript, as a security measure.

You should never use the HTML event handler attributes — those are outdated, and using them is bad practice.