Example: Basic Infinite Scrolling

Let's have fun with the `scroll` event and create a list which you can scroll infinitely (explanations below)!

You can run this code snippet on any page - just make sure that **you can scroll vertically** (either by adding enough dummy content, by adding some styles that add a lot of `height` to some elements or by shrinking the browser window vertically).

```
let curElementNumber = 0;

function scrollHandler() {
   const distanceToBottom =
document.body.getBoundingClientRect().bottom;

   if (distanceToBottom < document.documentElement.clientHeight +
150) {
       const newDataElement = document.createElement('div');
       curElementNumber++;
       newDataElement.innerHTML = `<p>Element
${curElementNumber}</p>`;
       document.body.append(newDataElement);
   }
}

window.addEventListener('scroll', scrollHandler);
```

So what's happening here?

At the very bottom, we register the `scrollHandler` function as a handler for the `'scroll'` event on our window object.

Inside that function, we first of all **measure the total distance** between our viewport (top left corner of what we currently see) and the end of the page (**not** just the end of our currently visible area) => Stored in `distanceToBottom`.

For example, if our browser window has a height of `500px`, then distanceToBottom could be `684px`, assuming that we got some content we can scroll to.

Next, we **compare the distance** to the bottom of our overall content (`distanceToBottom`) **to the window height + a certain threshold** (in this example `150px`).
`document.documentElement.clientHeight` is preferable to `window.innerHeight` because it respects potential scroll bars.

If we have **less than `150px` to the end of our page content**, we make it into the if-block (where we append new data).

Inside of the if-statement, we then create a new `<div>` element and populate it with a `<p>` element which in turn outputs an incrementing counter value.