



🏠 → [Browser: Document, Events, Interfaces](#) → [UI Events](#)

📅 2nd December 2019

# Scrolling

The `scroll` event allows to react on a page or element scrolling. There are quite a few good things we can do here.

For instance:

- Show/hide additional controls or information depending on where in the document the user is.
- Load more data when the user scrolls down till the end of the page.

Here's a small function to show the current scroll:

```
1 window.addEventListener('scroll', function() {  
2   document.getElementById('showScroll').innerHTML = window.pageYOffset + 'px'  
3 });
```

In action:

Current scroll = **scroll the window**

The `scroll` event works both on the `window` and on scrollable elements.

## Prevent scrolling

How do we make something unscrollable?

We can't prevent scrolling by using `event.preventDefault()` in `onscroll` listener, because it triggers *after* the scroll has already happened.

But we can prevent scrolling by `event.preventDefault()` on an event that causes the scroll, for instance `keydown` event for `pageUp` and `pageDown`.

If we add an event handler to these events and `event.preventDefault()` in it, then the scroll won't start.

There are many ways to initiate a scroll, so it's more reliable to use CSS, `overflow` property.

Here are few tasks that you can solve or look through to see the applications on `onscroll`.

## ✓ Tasks

Endless page

importance: 5

Create an endless page. When a visitor scrolls it to the end, it auto-appends current date-time to the text (so that a visitor can scroll more).

Like this:

## Scroll me

Date: Fri Feb 14 2020 13:18:47 GMT+0530 (India Standard Time)

Date: Fri Feb 14 2020 13:18:47 GMT+0530 (India Standard Time)

Date: Fri Feb 14 2020 13:18:47 GMT+0530 (India Standard Time)

Date: Fri Feb 14 2020 13:18:47 GMT+0530 (India Standard Time)

Please note two important features of the scroll:

1. **The scroll is “elastic”.** We can scroll a little beyond the document start or end in some browsers/devices (empty space below is shown, and then the document will automatically “bounces back” to normal).
2. **The scroll is imprecise.** When we scroll to page end, then we may be in fact like 0-50px away from the real document bottom.

So, “scrolling to the end” should mean that the visitor is no more than 100px away from the document end.

P.S. In real life we may want to show “more messages” or “more goods”.

[Open a sandbox for the task.](#)

solution

---

## Up/down button

importance: 5

Create a “to the top” button to help with page scrolling.

It should work like this:

- While the page is not scrolled down at least for the window height – it’s invisible.
- When the page is scrolled down more than the window height – there appears an “upwards” arrow in the left-top corner. If the page is scrolled back, it disappears.
- When the arrow is clicked, the page scrolls to the top.

Like this (top-left corner, scroll to see):

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83
84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102
103 104 105 106 107 108 109 110 111 112 113 114 115 116
117 118 119 120 121 122 123 124 125 126 127 128 129 130
131 132 133 134 135 136 137 138 139 140 141 142 143 144
145 146 147 148 149 150 151 152 153 154 155 156 157 158
159 160 161 162 163 164 165 166 167 168 169 170 171 172
173 174 175 176 177 178 179 180 181 182 183 184 185 186
```



[Open a sandbox for the task.](#)

solution

## Load visible images

importance: 4

Let's say we have a slow-speed client and want to save their mobile traffic.

For that purpose we decide not to show images immediately, but rather replace them with placeholders, like this:

```
1 
```

So, initially all images are `placeholder.svg`. When the page scrolls to the position where the user can see the image – we change `src` to the one in `data-src`, and so the image loads.

Here's an example in `iframe` :

Text and pictures are from <https://wikipedia.org>.

**All images with `data-src` load when become visible.**

## Solar system

The Solar System is the gravitationally bound system comprising the Sun and the objects that orbit it, either directly or indirectly. Of those objects that orbit the Sun directly, the largest eight are the planets, with the remainder being significantly smaller objects, such as dwarf planets and small Solar System bodies. Of the objects that orbit the Sun indirectly, the moons, two are larger than the smallest planet, Mercury.

The Solar System formed 4.6 billion years ago from the gravitational

Scroll it to see images load “on-demand”.

Requirements:

- When the page loads, those images that are on-screen should load immediately, prior to any scrolling.
- Some images may be regular, without `data-src`. The code should not touch them.
- Once an image is loaded, it should not reload any more when scrolled in/out.

P.S. If you can, make a more advanced solution that would “preload” images that are one page below/after the current position.

P.P.S. Only vertical scroll is to be handled, no horizontal scrolling.

[Open a sandbox for the task.](#)

solution



Previous lesson

Next lesson



Share



[Tutorial map](#)

## Comments

- If you have suggestions what to improve - please [submit a GitHub issue](#) or a pull request instead of commenting.
- If you can't understand something in the article – please elaborate.
- To insert a few words of code, use the `<code>` tag, for several lines – use `<pre>`, for more than 10 lines – use a sandbox ([plnkr](#), [JSBin](#), [codepen](#)...)