



Rev. 08Aug2019

Exercise 41

ARRAYS AND EVENT LISTENERS

Here's what we'll be building:

Please read each paragraph carefully and check each one off when done.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. ☐ Done

Epsum factorial non deposit quid pro quo hic escorol. Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum. Defacto lingo est igpay atinlay. Marquee selectus non provisio incongruous feline nolo contendere. Gratuitous octopus niacin, sodium glutimate. Quote meon an estimate et non interruptus stadium. Sic tempus fugit esperanto hiccup estrogen. Glorious baklava ex librus hup hey ad infinitum. Non sequitur condominium facile et geranium incognito. Epsum factorial non deposit quid pro quo hic escorol. Marquee selectus non provisio incongruous feline nolo contendere Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum. ☐ Done

A couple of things to note:

1. The "**Please read each...**" `<div>` at the top will remain *fixed*, while the text below it will *scroll*.
2. Each paragraph (8 in total) has a checkbox with a **Done** label next to it. When the checkbox is clicked, the paragraph will be removed from the page.

Open `index.html` to examine the HTML. You won't be changing this page at all, but notice this code that appears at the end of each paragraph: `<input type="checkbox" class="checkbox" />Done`

Those two notes above tell us *what* we need to make happen. Let's think about *how* we're going to accomplish them.

Note that I said 'a good way' — not 'the right way'. The fact is that different programmers may choose different methods to accomplish the same task.

This isn't a straight-forward process. As a programmer, you'll draw on your knowledge and your past experience to try to determine a good way of moving forward.

Here are some notes I might make as I try to formulate a plan:

- Probably a good idea to create a CSS class selector for `hidden` and have a rule that will set `display` to `none`
- I'll probably want to set event listeners on the checkboxes. I can get to those using `querySelectorAll('.checkbox')`
- I'll need an event-listener function that, when called, will add the class, `hidden`, not to the checkbox, but to the parent element of the checkbox, which includes the greeking text, the checkbox, and the **Done** label
- The `querySelectorAll` will produce a *odelist* that I'll have to turn into an array
- Once I have that array, I can *loop over* it adding an event listener to each individual checkbox
- I'll need to manipulate the CSS position element somehow to prevent the top `<div>` from scrolling with the rest of the page
- Looks like I'll need to do some simple CSS rules to change the `font-family` and `color` to make my page match the mockup

What to do first? Again, different programmers will have different takes on this. I'll usually tackle the JavaScript first, so I'll formulate a JavaScript plan and then provide little instructions in the form of JavaScript comments.

Seems like the first thing to do is to get all the checkboxes.

```
// get all the checkboxes
```

```
// convert the nodelist into an array
```

Now, to create a function that will be associated with the event listener. Its purpose is to create a `class` attribute called `hidden` and attach it to the `event.target`.

```
// create function that accepts an event
```

```
// get the event target's parent element
```

```
// create a class attribute
```

```
// set it to 'hidden'
```

```
// attach class attribute to parent element
```

```
// call forEach method on the array, setting a click event listener on each array element
```

Here's a little secret: many (I suspect most) programmers don't bother to formulate a plan. They just start coding. That can work initially, but as the application gets larger and more complex, it leads to significant levels of frustration and slows down ongoing development. The best time to make good habits is when you're first starting out...

With the JavaScript out of the way, we just have the CSS to fix. CSS is very different from HTML or JavaScript. Conceptually, CSS is very simple: you choose selectors that match the HTML elements you want to style, and then you create property rules for that selector.

But there are *so many* properties! And so much of CSS is trial-and-error. So, instead of walking you through the CSS, I simply present it to you and hope that you'll spend time seeing how I solved the problems. The best method for this is to change the values on the properties and see how the page is affected.

Hint: check out

`event.target.parentElement`

And, yes, once again I did
a lot of trial-and-error
to get it the way I
wanted it