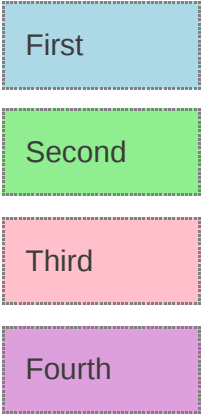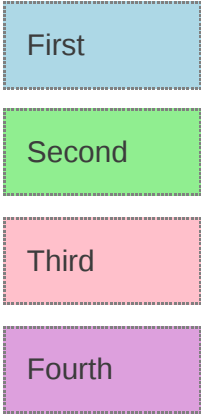# 'z-index' (OPTIONAL)

**Note**: this section is optional material included for the curious. It will not appear on any graded question.

In the previous sections, we named four positioning properties:`left`, `top`, `right` and `bottom`. With these properties, we can govern the placement of positioned elements in both the x and y axis. But there is a fifth positioning property: `z-index`.

```
z-index: 3;
```

Like the other positioning properties, `z-index` only applies to positioned elements (elements that have their position property set to `relative`, `absolute` or `fixed`, but not `static` ). With the `z-index` you can control overlapping - whether or not an element is in front of or behind other *sibling* positioned elements . The z-index can be an integer 0 or higher. The higher the number, the more "topmost" or "overlapping" the element will be.

In the sample below, we have two lists with relatively positioned list items and background colors. We are forcing them to overlap by making them position relative and using negative margins ( `margin-bottom:-20px;` ). The list on the left has no z-index values set, so the later elements overlap the earlier ones. But on the right, we govern the overlapping with the z-index property.

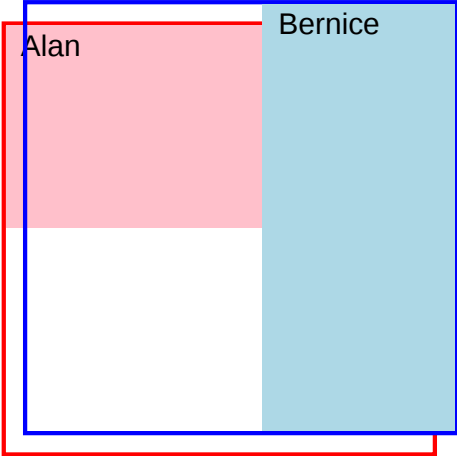| CSS | Result | CSS | Result |
|-----|--------|-----|--------|
| `/* no z-index set */` | First<br><br>Second<br><br>Third<br><br>Fourth | `.first { z-index: 4; }`<br>`.second { z-index: 3; }`<br>`.third { z-index: 2; }`<br>`.fourth { z-index: 1; }` | First<br><br>Second<br><br>Third<br><br>Fourth |

- z-index has no effect on position:static (the default) elements

- if it is not set, siblings that appear later in the HTML document overlap (are "higher than") earlier siblings.

- z-index is relative between siblings, not any arbitrary elements

## SIBLINGS AND NESTING

It is entirely possible that one element with z-index:100 could appear**below** another element with z-index:1;

This can happen because the z-index is used to figure out which sibling is higher than another. But if two elements are not siblings, then the z-index of their respective sibling ancestors will need to be calculated to figure out which is higher.

Below is a simple example: there are two overlapping sibling divs, "Albert" and "Betty". Albert has a red border and is  z-index:1.  Betty has a blue border, and is z-index:2. Therefore, Betty and her child Bernice are *higher* than Albert and his child Alan. Albert's child Alan has a z-index of 100, which is the highest z-index of any of them, but because Alan's parent, Albert is lower than Betty, Alan remains behind. Alan's high z-index is only relevant to his siblings, not to cousins further out in the document.

| HTML | CSS | Result |
|------|-----|--------|
| ```<div class="albert"> <div class="alan">Alan</div> </div> <div class="betty"> <div class="bernice"> Bernice</div> </div>``` | ```.albert { z-index: 1; } .betty { z-index: 2; } .alan { z-index: 100; } .bernice { z-index:0; }``` |  |

---

## 'Z-INDEX' KNOWLEDGE CHECKS

# 1. Will the z-index property work...

Will the z-index property work on an item with `position:relative;` ?

- ○ yes
- ○ no

CHECK

# 2. On an item with 'position:static;' ...

Will the z-index property work on an item with `position:static;` ?

- ○ no
- ○ yes

CHECK