

I want a css rule that will cause one section element to go to the right side of the screen without affecting the rest of the section elements. I have gotten the effect I want with `margin-right: -40px`, but this is altering the size of the container div and causing the printed output to be narrower on the page. How do I remove the relevant section elements out of the document flow such that adding this margin displays as it currently does and the rest of the document is not affected



Thought for a few seconds >

Use a class on only the section you want to shift, and use `transform`, not negative margin.

</> CSS



```
section.shift-right {  
  transform: translateX(40px);  
}
```

Then:

</> HTML



```
<section class="shift-right">  
  ...  
</section>
```

`transform: translateX(40px)` visually moves the section to the right **without changing the layout box**. That means it will appear shifted, but it will not widen or shrink the parent container, and it will not affect the printed width of the rest of the document.

This is probably the cleanest replacement for:

</> CSS



```
margin-right: -40px;
```

because margins participate in layout, while transforms do not.

If you truly want the element removed from document flow, you could use absolute positioning:

</> CSS



```
.section-wrapper {  
  position: relative;  
}  
  
section.shift-right {  
  position: absolute;  
  right: -40px;  
}
```

But that usually creates a new problem: the section no longer takes up vertical space, so following content may overlap it. You would then need a wrapper with a fixed/minimum height.

For your case, use this:

</> CSS



```
section.shift-right {  
  transform: translateX(40px);  
}
```

Or, if you want it only for print:

</> CSS



```
@media print {  
  section.shift-right {  
    transform: translateX(40px);  
  }  
}
```



...

it is still affecting the size of the document

Thought for a couple of seconds >

That means the browser is still counting the **visual overflow** of the shifted section when calculating the printable/page area. `transform` does not affect normal layout, but it *can* still affect the document's visual bounding box, especially in print.

Use absolute positioning inside a wrapper that clips overflow.

</> CSS



```
.section-shell {  
  position: relative;  
  overflow: visible;  
}  
  
.section-shell.no-layout-spill {  
  overflow: clip;  
  /* fallback if needed */  
  overflow: hidden;  
}  
  
.section-shell.no-layout-spill > section.shift-right {  
  position: relative;  
  left: 40px;  
}
```

HTML:

</> HTML



```
<div class="section-shell no-layout-spill">  
  <section class="shift-right">  
    ...  
  </section>  
</div>
```

But the more direct print-safe version is this:

</> CSS



```
section.shift-right {  
  position: relative;  
  left: 40px;  
  width: calc(100% - 40px);  
}
```

That moves the section right while keeping its right edge from expanding the page/document width.

So instead of:

</> CSS



```
section.shift-right {  
  margin-right: -40px;  
}
```

use:

</> CSS



```
section.shift-right {  
  position: relative;  
  left: 40px;  
  width: calc(100% - 40px);  
}
```

If the goal is “move right 40px but do not make the printable document wider,” the missing piece is usually reducing the width by the same amount:

</> CSS



```
section.shift-right {  
  margin-left: 40px;  
  width: calc(100% - 40px);  
}
```

That one may be the safest. It keeps the element in flow, shifts it right, and prevents overflow from changing the printed page width.



...

how do I increase the size of everything that 1) is not  
.shift-right and 2) is not a child of .shift-right

Use `:not()` plus a descendant exclusion.

For example, to increase font size for everything **except** `.shift-right` and anything inside `.shift-right`:

⌵ CSS



```
body *:not(.shift-right):not(.shift-right *) {  
  font-size: 110%;  
}
```

That means:

⌵ CSS



```
:not(.shift-right)
```

excludes the `.shift-right` element itself.

⌵ CSS



```
:not(.shift-right *)
```

excludes every descendant inside `.shift-right`.

A cleaner version is usually to apply the larger size to the normal document wrapper, then reset `.shift-right`:

⌵ CSS



```
.document-body {  
  font-size: 110%;  
}  
  
.shift-right {  
  font-size: initial;  
}
```

But if you specifically need the selector-only approach, use:



</> CSS



```
body *:not(.shift-right):not(.shift-right *) {  
  font-size: 110%;  
}
```

If you mean “increase the visual size of whole elements,” not just text, use `zoom` or `transform`, but for print/layout work, `font-size` is usually safer than scaling the entire element.



...

I want width to be width + 40pxx

Use `calc()` :

`</>` CSS



```
.your-element {  
  width: calc(100% + 40px);  
}
```

For your shifted section, it may be:

`</>` CSS



```
section.shift-right {  
  width: calc(100% + 40px);  
}
```

But note: if you add `40px` to the width, it can create overflow and may again affect print/document sizing.

If you want it **40px wider while shifted right**, you probably want the extra width to extend left or right specifically.

Extra width to the **right**:

`</>` CSS



```
section.shift-right {  
  width: calc(100% + 40px);  
}
```

Extra width to the **left**, while keeping the right edge aligned:

`</>` CSS



```
section.shift-right {  
  margin-left: -40px;  
  width: calc(100% + 40px);  
}
```

Extra width split evenly:

</> CSS



```
section.shift-right {  
  width: calc(100% + 40px);  
  margin-left: -20px;  
}
```

The basic answer is:

</> CSS



```
width: calc(100% + 40px);
```



...