

TEACHING · 2026-09

Three small lines that made a teaching page easier to use

The September mobile fix to Think Like Python is a reminder to test the lesson where people actually read it.



By Lyra

Published Oct 14, 2026 · 4 min read

September 2026 retrospective · Story context 2026-09-25



Side-by-side examples are great when you're teaching a familiar idea in a new language. You can put an ABAP example next to Python, walk through the differences, and let the reader compare without scrolling back and forth.

Then someone opens the lesson on a phone, where each narrow column competes for the same few inches of screen.

The September 25 change to ProgreTech's [Think Like Python](#) page was small: add the viewport declaration and stack the comparison columns inside the existing mobile breakpoint. Three lines

addressed a very visible reading problem. It's a useful example of how the quality of teaching depends on the page around the explanation.

Tell the browser which viewport you mean

Mobile browsers can use a wider layout viewport to display pages designed for desktop screens. Without the intended viewport metadata, your carefully chosen breakpoints may behave differently from what you expect on the device.

The page added:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

That gives a responsive layout a sensible starting point. It still needs responsive rules. A wide two-column section can remain awkward even when the browser uses the device width.

The existing small-screen media query gained two declarations:

```
@media (max-width: 700px) {  
  .split { flex-direction: column; }  
  .split div { width: 100%; }  
}
```

These names reflect that specific page. In a larger component system, target the direct children you intend to resize so nested containers don't inherit an overly broad rule. The useful idea is to change the reading order deliberately at the width where comparison becomes cramped.

Preserve the relationship between examples

Stacking columns solves width, but it can weaken the connection between two code samples. Keep labels close to their examples. Use matching headings and explain the specific concept each pair demonstrates.

For a short lesson about iteration, the comparison might answer: Where does the loop start? Which variable changes? What ends the loop? A reader should be able to follow those questions in either layout.

Avoid relying on position alone. "The code on the right" becomes unhelpful once the page stacks vertically. "The Python example" survives the layout change, a printed copy, and a screen reader's reading order.

This is also a good moment to check code blocks. Let a long line scroll inside its own block when necessary. If it forces the entire page wider than the phone, navigation and prose become harder to use together.

A practical review pass

Take one real teaching page and review it in this order:

1. Open it at a narrow phone width with the actual article content.
2. Read the headings without the body and check whether the progression makes sense.
3. Follow each example label to the code it describes.
4. Increase the text size and look for clipped controls or overlapping labels.
5. Use the keyboard to reach navigation, examples, and download links.
6. Check the page in print preview if it is intended as study material.

You can automate the presence of the viewport tag or a screenshot at a chosen width. Still read the rendered lesson. A technically stacked layout can have a confusing explanation order, an oversized illustration, or a code panel that swallows the whole screen.

Teach with the reader's working context in mind

A developer learning Python might be reading between tasks, using a phone beside another computer, or saving a page for later. Those situations deserve an understandable page without special instructions.

There is more accessibility work beyond this small fix. For this page, though, the next step was clear: open the lesson at the width where reading became awkward, correct the layout, and read it again. You can make that same pass on one of your own pages in a few minutes.

When the layout gets out of the way, the reader has more attention left for the code. That's a worthwhile teaching improvement, even when the patch is tiny.

Sources & further reading

Checked 2026-10-14 · Story context 2026-09-25

1. [Think Like Python](#) (2026-10-14)
2. [MDN viewport metadata](#) (2026-10-14)

[Read the current article online →](#)

Support the workshop

Explore software and learning offers at [StackSocial](#). Affiliate link: ProgreTech may earn a commission from purchases. Check current price, license and renewal terms before buying.

Find our projects at progretech.com.

ProgreTech LLC · Build / Learn / Improve / Together

This file contains its text, illustrations and fonts. No account, scripts or network connection required to read. External reference links need internet access.