

Recently I wrote about the concept (and potential uses) of [Shell Components](#) in Figma based design systems (not that the concept should only be used in Figma alone, it simply works with the ability to swap component instances from the tools panel). I found that it does seem to be quite a useful way to consider building out more versatile and complex components for many people, not just myself. If you haven't already, go have a read of that introduction, then come back here. Done?

This strategy/way of thinking about componentry is of course not my own original idea, I simply arrived at these conclusions of my own volition and realised it has a multitude of valid applications. It is also a similar idea to that of [Compound Components](#) in react, so knowing that I have created similar strategies for designing which developers have been working with for some time is reassuring.

Again, this technique is also utilised for other types of components, take this file from the Figma community outlining how you can create components with fixed ratio sections.

I have decided to push this a little further for two reasons;

1. Examples breed ideas

To start flexing those brains of yours, I wanted to help give a few more ideas relating to how this could be implemented in *your* work. I have taken the opportunity to provide more examples of how this

new type of component can be built to house various types of content.

2. Doing solidifies new knowledge

It's all well and good to read a Medium article about *what* these components are, but you may find it useful to actually build these yourself. Once you have had the experience of putting them together for one application, more ideas may come to fruition that may apply to design systems you work on yourself. This is really the end goal.

Wrapping up

So please, check out the file via the big beautiful blue button bellow, explore some different possibilities with shell components, and Tweet me at [Rhys Webster](#) with your implementations. I'd love to keep developing the applications of these components. Happy Designing!

The previous module introduced you to many of the available programming tools and explained when shell scripts are the best tool to solve a problem.

This module provides more information to help you understand the commands and tools you will be using to create your first shell script. By the end of this module, you will be able to:

1. Explain situations in which a shell script is most useful
2. Define external and built-in commands

3. List commands used to communicate with users
4. Identify how tests are used within shell scripts
5. Define the different variable types used in shell scripts
6. Describe how control structures determine how a script operates