

ASSIGNMENT: #03 TOPIC: CSS LAYOUTS

Download and extract the zip archive provided with this assignment. The archive contains an html document ([webpage.html](#)) and an image of the great master Fu-Tzu ([pic.jpg](#)), generated by the DALL-E model. Set up your web development environment and visit the html document using your browser of choice.

The provided html document does not include any authored style, and it shows! In this assignment, you will properly style the web page and apply different layouts.

EXERCISE 1. OLD SCHOOL LAYOUTS

In the first exercise, we'll pretend it's the early 2000s and fixed-width design are still a decent choice. You are asked to write a stylesheet ([fixed-width.css](#)) so that the web page has the following characteristics:

1. The width of the fixed-width layout is approximately **800px**;
2. The **<main>** element has a two-column layout, with the main **<section>** (on the left) having spanning approximately 65% of the layout width and the **<aside>** element (on the right) occupying 30% of the width;
3. In this exercise, you are **not allowed** to use flexbox or grid layouts.
4. The final web page should look something like the picture reported in the next page. You are free to customize the appearance according to your own taste, as long as points 1. and 2. are satisfied.

When you are done, experiment with the browser's Responsive Design Mode and observe how your fixed-width layout behaves in viewports that are larger/smaller of the defined width.

When you are done, create a new stylesheet named [fluid.css](#) and include it in the web page in place of [fixed-width.css](#). Now, edit the [fluid.css](#) stylesheet so that the layout is **fluid** and is always as large as the entire viewport.

Again, when you are done, experiment with the browser's Responsive Design Mode and observe how your fixed-width layout behaves in larger/smaller viewports.

ASSIGNMENT: #03 TOPIC: CSS LAYOUTS

The Book of Programming

Fragments from an ancient tome of wisdom.

[Home](#) [Data/Control](#) [Wisdom](#) [Web](#)

Web Wisdom

On the nature of HTML and CSS

Fu-Tzu once said: *HTML*, the *passive* substance, forms the *skeleton* upon which the web's flesh is draped. *CSS*, the *active* substance, lends character to the web's form. From the *plain* and *straightforward*, the *ornate* and *engaging* emerges.

On vertically centering divs

In a prior age of web development, before the introduction of *flexbox* and *grid* layouts, a student was trying to *horizontally and vertically center* a div contained inside another div. After looking at his efforts for a while, Fu-Tzu said to the student: *Worry not, young one, for one does not simply center a div.*

On JavaScript

A student was learning JavaScript. One day, he went crying to his master. "Master, if evaluate the JavaScript expression `2+2-2` the interpreter computes 2 as a result". "That seems correct", replied Fu-Tzu. "But when I evaluate the expression `"2"+"2"-2` I get 20 as a result". Fu-Tzu said: "These are the wonders of JavaScript automatic type coercion, where even the simplest statement could lead to a dance of unexpected outcomes."



Fu-Tzu: The Master

With a long beard and robes adorned with clever coding jokes, Fu-Tzu was a legend among programmers.

It is said his first word was `printf`. Some of his wisdom was transcribed in the Book of Programming.

ASSIGNMENT: #03 TOPIC: CSS LAYOUTS

EXERCISE 2. MODERN LAYOUTS: FLEXBOX AND GRID

Drawing from the stylesheets you designed in **EXERCISE 1. OLD SCHOOL LAYOUTS**, create a new stylesheet named **flexbox.css**, and include it in the web page.

Replicate the web page layout using **flexbox**. In this exercise, you are **not allowed to use floats or grid layouts**. The **maximum width** of the content in the flexbox layout should not exceed **800px**, but the content is allowed to shrink if the viewport is smaller. Moreover, upon scrolling, the navbar should remain on top of the page and not exit the viewport.

When you are done, experiment with the browser's Responsive Design Mode and observe how your flexbox layout behaves when changing the size of the viewport.

When the flexbox layout is done, create a new stylesheet (it'll be the last one for this assignment!) named **grid.css**. Then, replicate the layout for the last time using a **grid**. In this exercise, you are **not allowed to use floats**, but you may use flexbox to align content within the header, the navbar, or the footer). For the grid layout, It might be a good idea to change the webpage html and move the **<aside>** element, placing it outside the **<main>**, as its subsequent sibling. This would allow you to declare the body as a grid container, and place its grid-item children (the **<header>**, **<nav>**, **<main>**, **<aside>** and **<footer>** elements) directly in grid cells.

Once your grid layout is complete, don't forget to experiment with the browser's Responsive Design Mode and observe how it behaves when changing the size of the viewport.

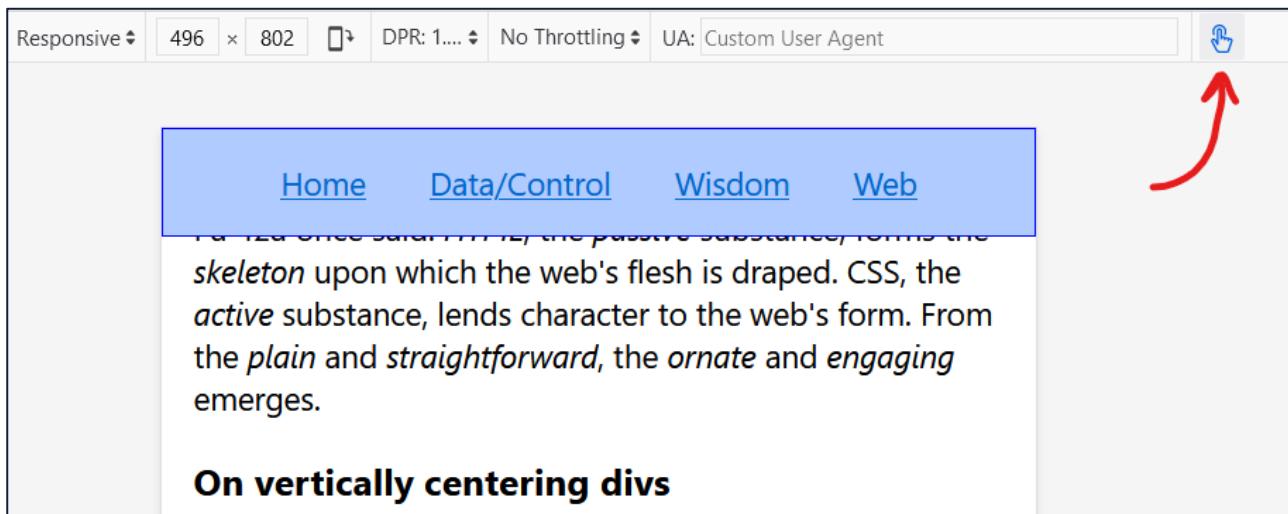
EXERCISE 3. MAKING OUR WEB PAGE RESPONSIVE

As you've noticed, in small viewports our web page does not look too good, mostly because there is not enough space to display the sidebar on the side. As a last task for this assignment, you are asked to edit the flexbox and grid layouts you have developed so far to make them **responsive**.

In particular, when the screen is smaller than **600px**, the sidebar should collapse and appear after (below) the main section.

Edit the stylesheets accordingly and, when you are done, check that the layout is behaving as expected using the Responsive Design Tool in your browser of choice. **Make sure not to forget the viewport meta tag, and enable the touch simulation mode to simulate a mobile browser, as shown in the screen capture below!**

ASSIGNMENT: #03 TOPIC: CSS LAYOUTS



EXERCISE 4. MAKING OUR FIRST WEBSITE RESPONSIVE

Remember the website you developed for the *Book of Programming* back in [ASSIGNMENT #01?](#) Leverage the knowledge you acquired with the previous exercises, and include a stylesheet in all the web pages to make the website look good and be fully responsive. You are free to change the HTML structure as well, to help with the implementation of the layout. Be sure to check that the website renders properly on all web pages, at different viewport sizes.