

Key points of coding case study

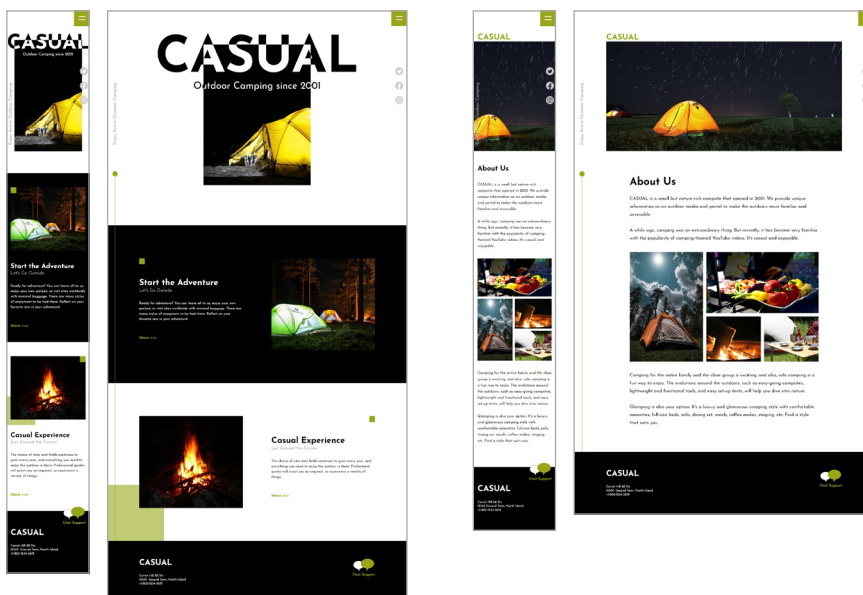


We have summarized the key points and techniques used in this 'HTML and CSS Coding Practice Book 5' web development.

Recap

The web design for this project has many decorative parts. It is required to fit the size with other ones, adjust the stacking order and position. It doesn't look easy to achieve the layout with responsiveness.

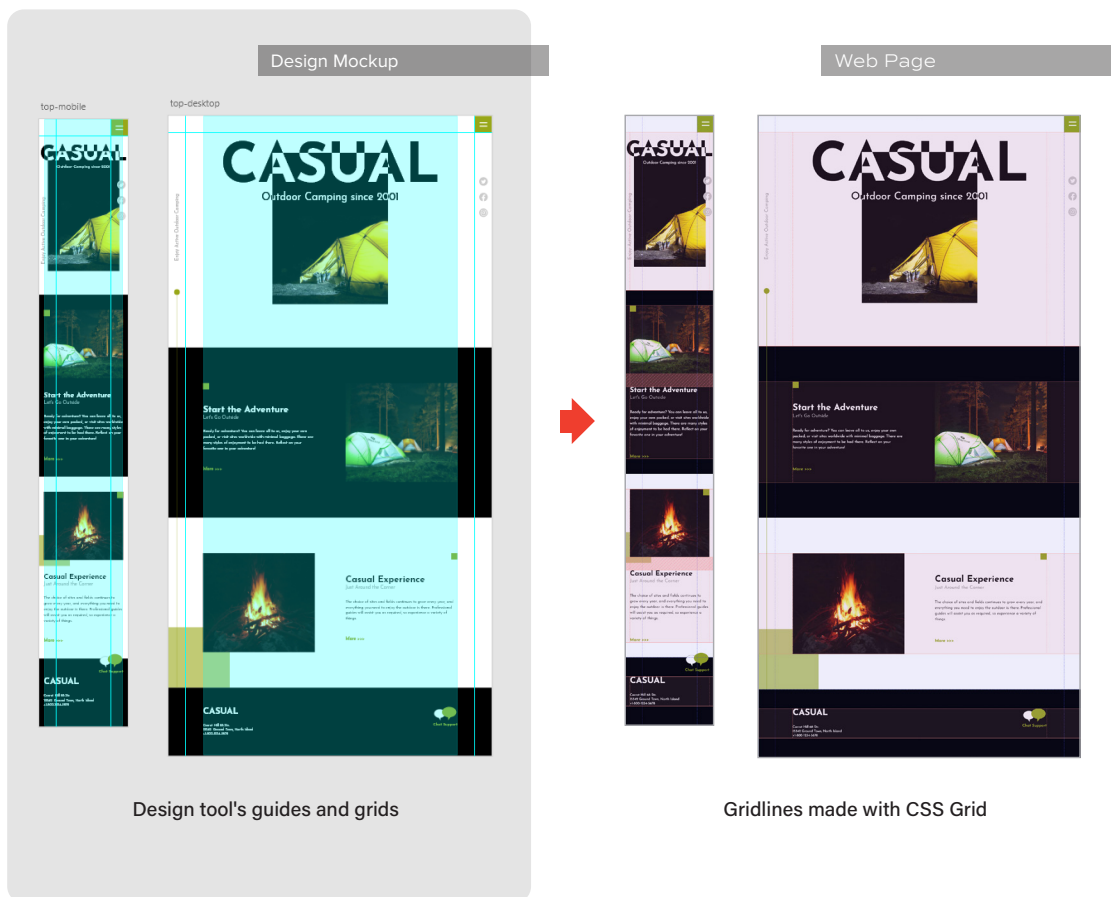
However, we have CSS Grid. So the key point is how to use it to make the web design.



Using design tool's guides and grids with CSS Grids

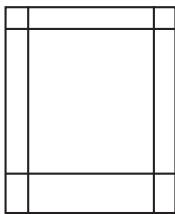
In CSS Grid, it takes time to determine the grid structure to control the layout. But it'll be easier if the design has some grids, such as guides and layout grids of XD and Figma.

The design tool's guides and grids are for placing and aligning parts. So make the same structure guidelines with CSS Grid and place parts. It helps to make the web layout efficient.

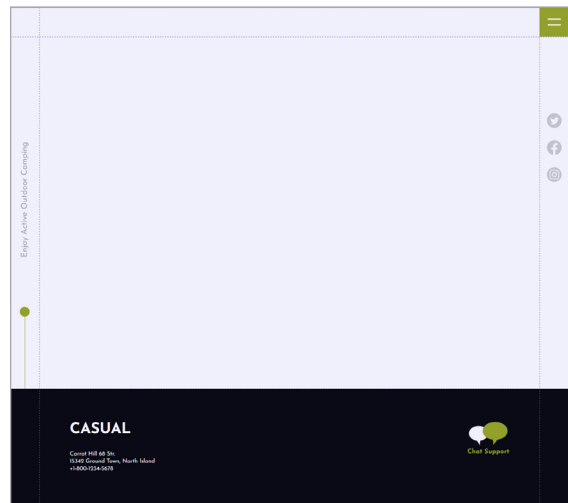
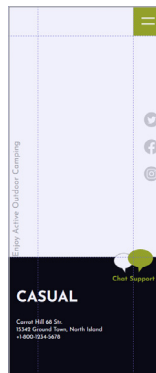


Don't control all layouts with a single grid and make the grid as simple as possible

If you try to control all layouts with a single grid, the grid structure will be complex and not easy to manage. To simplify, use two simple grids. Control the layout of the page frame with Page Grid and the main content with Parts Grid.



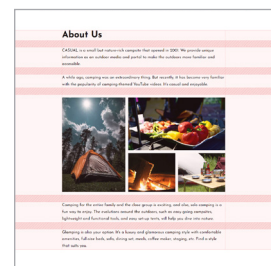
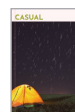
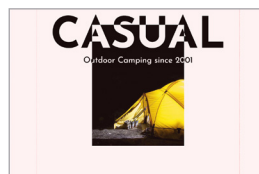
Page Grid



The layout of the page frame with Page Grid



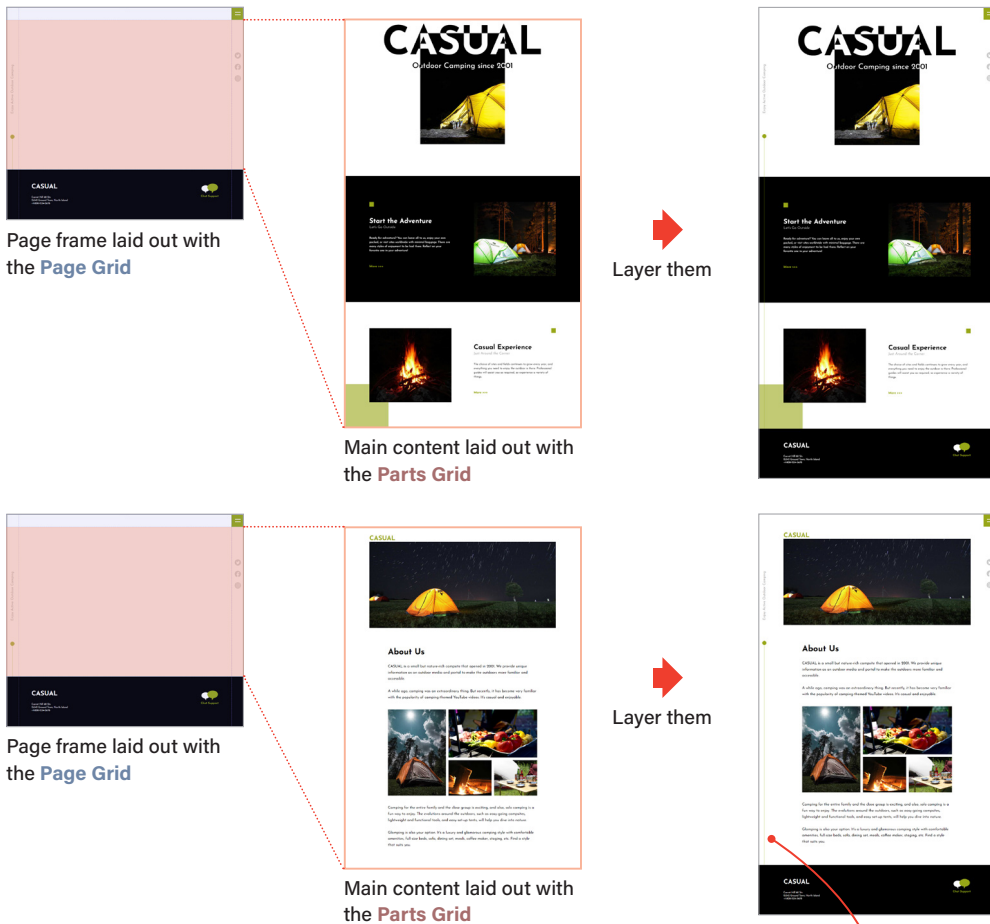
Parts Grid



The layout of the main content with Parts Grid

Layered multiple CSS Grids

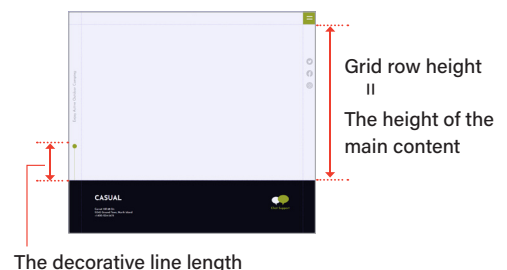
We layer two CSS Grids to layer the page frame on top of the main content. We achieve this by placing the main content laid out with the Parts Grid into the Page Grid. It feels like the layer function of a graphic tool like Adobe Photoshop.



Fitting the decorative line length to the main content

We need to fit and stretch the decorative line to the main content height. Layering two grids as above solves this layout because it places them on the same row of the Page Grid.

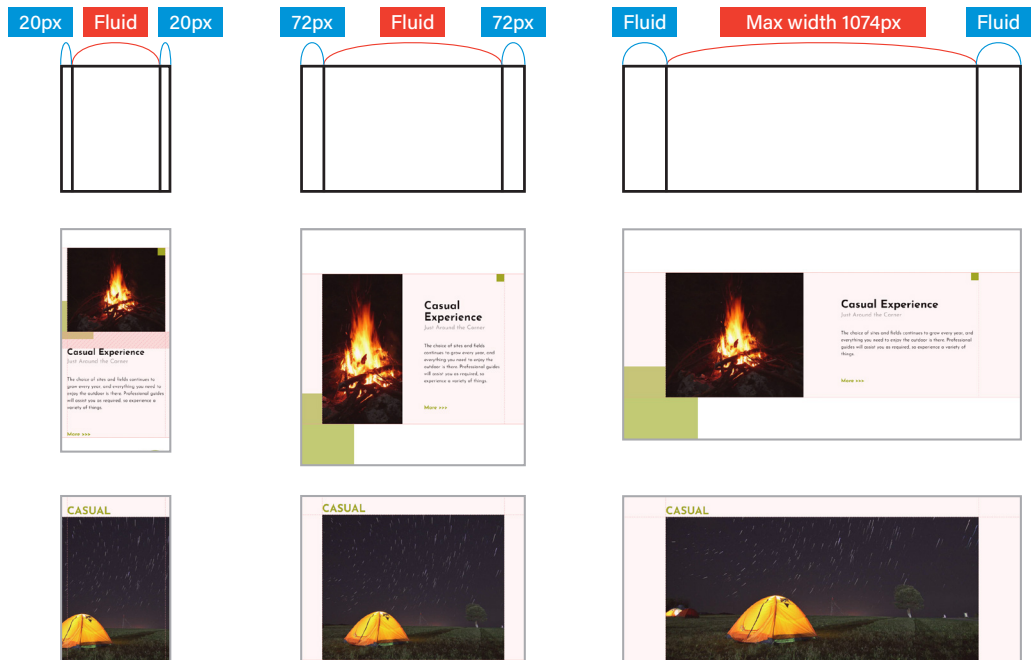
We only need to set the grid row height to auto-fit the main content and the decorative line length to auto-fit the row height.



Controle the content width and left/right space with CSS Grid

It's case-by-case whether or not to use CSS Grid to control the content width and left/right space. Each part is simple, and some elements need to be placed in the space area, so we use CSS Grid in this case study.

We create the three columns grid as Parts Grid. Each column width changes in three steps along with screen size.



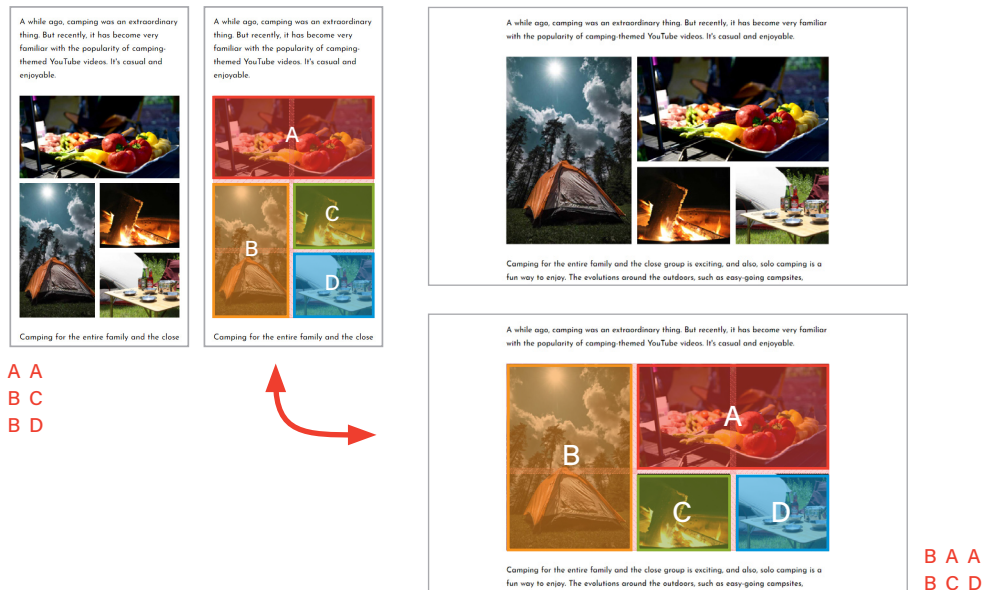
Fluid layout while pinned to the left edge

The green rectangle is pinned to the left edge and layer its upper right corner under the image. To keep this layout and make it fluid, place it into the first column of Parts Grid. The rectangle width will change along with the column width.



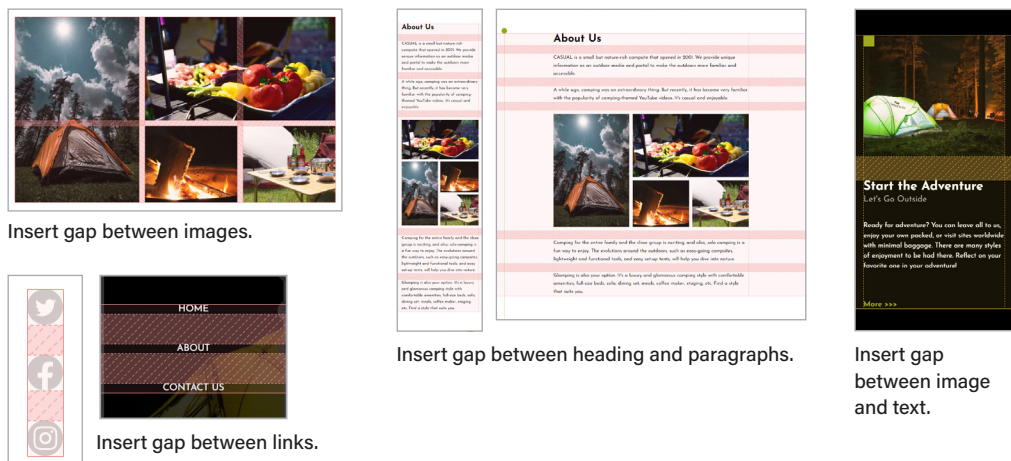
Mosaic layout with grid areas

The gallery needs to change the photo placement in responsive but doesn't need to overlap them. So we use the grid area, one of the CSS Grid functions suitable for a mosaic layout.



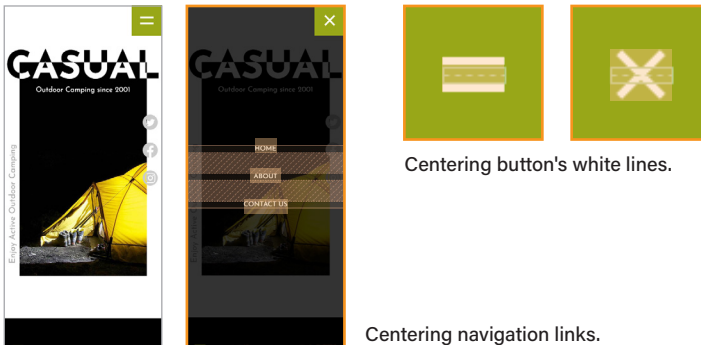
Spacing with CSS Grid gap

We use the CSS Grid gap to adjust spacing for images and texts.



Centering vertically and horizontally

We use the CSS Grid also for centering vertically and horizontally.



Inverting black and white

We make the text and image for the hero part overlap and invert the overlapped area color with CSS mix-blend-mode.



Making the scroll bar always visible on the desktop browser

We keep the scroll bar visible on the desktop browser to prevent changing the page width when the menu opens and closes.

We use the media query's pointer and hover and apply this setting only to the desktop.

