

Outline

1. **Customizing Outline Styles:**
Design a box with an outline that has a dashed style, a width of 4px, and a color of red. Include a hover effect that changes the outline style to solid and the color to green.
 2. **Outline vs. Border:**
Create two identical boxes. Use the `outline` property for one and the `border` property for the other. Write observations about how the outline and border behave differently in terms of spacing and rendering.
-

Font-Family in Other Languages

4. **Font for Multilingual Text:**
Write a paragraph containing text in English, Chinese, and Arabic. Use the `font-family` property to assign suitable fonts for each language segment. Ensure proper fallbacks in case specific fonts are unavailable.
 5. **Custom Web Fonts for Non-Latin Languages:**
Import a Google Font (e.g., Noto Sans for Japanese or Amiri for Arabic). Apply this font to a specific paragraph and compare the appearance with the browser's default font.
 6. **Fallback Fonts for Missing Characters:**
Create a heading with text that includes special characters from different languages. Use a font stack with fallbacks like `Arial`, `sans-serif`, `serif` and observe how the browser handles characters unavailable in the primary font.
-

Font Size

7. **Responsive Font Sizes:**
Create a responsive webpage where headings and paragraph text adjust their size using relative units like `em`, `rem`, or percentages. Test the design on different screen sizes.
 8. **Scaling Text with Media Queries:**
Use media queries to change the font size of a heading to:
 - 24px on small screens (below 600px)
 - 36px on medium screens (601px–1024px)
 - 48px on large screens (above 1024px).
 9. **Relative vs. Absolute Font Sizes:**
Create two blocks of text: one using relative font sizes (`em`, `rem`) and another using absolute sizes (`px`). Write a short analysis comparing the flexibility of the two approaches.
-

Display Grid

10. Basic Grid Layout:

Create a webpage section using `display: grid` with three rows and three columns. Assign different colors to each cell.

11. Grid Template Areas:

Design a webpage with a header, sidebar, main content, and footer using `grid-template-areas`. Assign appropriate areas and adjust their sizes using `grid-template-rows` and `grid-template-columns`.

12. Responsive Grid Design:

Create a photo gallery with a grid layout that displays:

- 1 column on small screens (below 600px),
- 2 columns on medium screens (601px–1024px),
- 3 columns on large screens (above 1024px).

13. Aligning Grid Items:

Create a grid container with four items. Use properties like `align-items`, `justify-items`, `align-self`, and `justify-self` to position the items in different ways.

BONUS: Combining Topics

14. Complete Card Layout:

Build a card layout that includes:

- A grid layout for the card structure.
 - Multilingual text using different font families.
 - Custom font sizes for different sections of the card.
 - A focusable button with a styled outline.
-