**Beginner Level CSS Exercises**

1. **Change Text Color**: Use CSS to change the text color of all <h1> elements to blue.
2. **Set Background Color**: Set the background color of a webpage to light gray using CSS.
3. **Style Paragraphs**: Use CSS to change the font size of all paragraphs to 16px and the font family to Arial.
4. **Add Padding and Margin**: Add padding of 10px and margin of 20px to all <div> elements.
5. **Center Align Text**: Center-align the text of all <h2> elements.
6. **Create a Border**: Add a 2px solid black border to all images on a webpage.
7. **Use a CSS Class**: Create a CSS class named .highlight that makes text bold and apply it to a specific paragraph.
8. **Style a Button**: Change the background color of a button to green and text color to white when the user hovers over it.
9. **Use External Stylesheet**: Link an external CSS file to an HTML document and apply a style to change the body’s font color.
10. **Change Font Weight**: Make the text of all <h3> elements bold using CSS.
11. **Apply a Box Shadow**: Add a box shadow to an image with horizontal offset 5px, vertical offset 5px, blur radius 10px, and color gray.
12. **Use the Universal Selector**: Use the universal selector \* to apply a border-box sizing to all elements.
13. **Create a Fixed Navigation Bar**: Make a navigation bar fixed at the top of the page.
14. **Use the :hover Pseudo-Class**: Change the text color of a link when the user hovers over it.
15. **Add Line Height**: Increase the line height of all paragraphs to 1.6 for better readability.
16. **Apply Text Decoration**: Remove the underline from all links using text-decoration.
17. **Set Maximum Width**: Limit the width of an image to 100% of its container.
18. **Use Multiple Selectors**: Style both <h1> and <h2> elements with the same color and font size.
19. **Set the Display Property**: Use the display: block; property to make all images display as block elements.
20. **Change the Cursor**: Change the cursor to a pointer when hovering over a button.

**Intermediate Level CSS Exercises**

1. **Create a Responsive Grid Layout**: Use CSS Grid to create a 3-column layout that adjusts to 1 column on smaller screens.
2. **Style a Form**: Style a form with rounded input fields, a different color on focus, and a styled submit button.
3. **Implement a Flexbox Layout**: Create a flexbox-based navigation bar with evenly spaced links that adapt to different screen sizes.
4. **Use CSS Variables**: Define CSS variables for primary and secondary colors and apply them throughout the webpage.
5. **Create a CSS Tooltip**: Create a tooltip that appears when the user hovers over a word or icon.
6. **Use nth-child Selector**: Style every other row in a table with a different background color using nth-child selector.
7. **Design a Card Component**: Create a card component with a shadow, padding, rounded corners, and a hover effect.
8. **Create a Responsive Image Gallery**: Use CSS Flexbox to create a responsive image gallery with a gap between images.
9. **Animate a Button**: Use CSS animations to make a button change color and scale slightly when hovered over.
10. **Create a Sticky Sidebar**: Develop a sticky sidebar that remains fixed while scrolling through the page content.
11. **Use the :before and :after Pseudo-Elements**: Create a custom list item style using the :before pseudo-element.
12. **Implement a CSS Accordion**: Use CSS and HTML to create a simple accordion component.
13. **Create a CSS Overlay**: Implement an overlay effect that displays additional information when hovering over an image.
14. **Style a Dropdown Menu**: Create a dropdown menu with smooth transitions and styling.
15. **Use CSS Transformations**: Rotate an element by 45 degrees when the user hovers over it.
16. **Create a Progress Bar**: Implement a simple progress bar using CSS transitions.
17. **Add a CSS Mask**: Use a CSS mask to display an image with a custom shape.
18. **Build a Simple CSS Animation**: Create an animation that moves an element from left to right over 5 seconds.
19. **Use Media Queries for Typography**: Adjust font size and line height for different screen sizes using media queries.
20. **Create a Custom Checkbox**: Style a custom checkbox with a different color and size.

**Advanced Level CSS Exercises**

1. **Build a Complex Grid Layout**: Use CSS Grid to create a complex layout with nested grids, including a header, footer, sidebar, and main content area.
2. **Implement a CSS Parallax Effect**: Create a parallax scrolling effect using CSS only.
3. **Design a Full-Screen Responsive Menu**: Create a full-screen overlay navigation menu that appears when a hamburger icon is clicked.
4. **Create a Custom CSS Loading Spinner**: Design a custom loading spinner using CSS animations.
5. **Develop a Responsive CSS Framework**: Build a small custom CSS framework with grid system, typography, and utility classes.
6. **Create a Complex CSS Animation**: Design a multi-step CSS animation that changes multiple properties, such as color, position, and opacity.
7. **Implement a Dark Mode Toggle**: Create a dark mode toggle switch using CSS variables and JavaScript.
8. **Design a Complete Dashboard Layout**: Use Flexbox and Grid to design a fully responsive admin dashboard with cards, tables, and charts.
9. **Create an Accessible Custom Modal**: Design an accessible modal with focus trapping and keyboard navigation.
10. **Build a Multi-column Layout with Flexbox**: Create a multi-column layout using Flexbox that adapts to different content sizes.
11. **Design a CSS-only Dropdown Search Filter**: Implement a dropdown search filter using CSS and HTML.
12. **Create a CSS-based Star Rating System**: Design a star rating system where stars change color on hover.
13. **Develop a CSS Timeline**: Create a vertical or horizontal timeline using CSS.
14. **Implement a CSS Image Carousel**: Build an image carousel with sliding animations using only CSS.
15. **Create a Multi-layer Parallax Effect**: Use multiple background images and CSS properties to create a multi-layer parallax effect.
16. **Design a Flip Card Animation**: Create a flip card effect using CSS transitions and 3D transforms.
17. **Implement CSS Scroll Snap**: Use CSS Scroll Snap to create a smooth scrolling experience for a slideshow.
18. **Build a Custom Range Slider**: Design a custom range slider with styled track, thumb, and fill using CSS.
19. **Create an Interactive Infographic with CSS**: Design an interactive infographic that changes appearance based on user interactions.
20. **Use Advanced CSS Selectors**: Use advanced selectors like :not(), :nth-of-type(), and attribute selectors to style complex components.

These exercises cover a wide range of CSS topics and difficulty levels, helping you to progressively improve your CSS skills.