

CSS:

13. Pseudo-classes and Pseudo-elements:

Pseudo-classes:

Pseudo-classes are used to select and style elements based on their state or position in the document. For example, `:hover` is a pseudo-class that selects and styles an element when the user hovers over it.

Code Example:

```
```html
<style>
 /* Style the anchor link when hovered */
 a:hover {
 color: red;
 }

 /* Style the first child of a list */
 li:first-child {
 font-weight: bold;
 }
</style>

 Item 1
 Item 2
 Item 3

```
```

Pseudo-elements:

Pseudo-elements allow you to style parts of an element. For example, `::before` creates a pseudo-element that inserts content before the actual content of an element.

Code Example:

```
```html
<style>
 /* Add a double arrow before each paragraph */
 p::before {
 content: "⇒ ";
 }

 /* Style the first line of each paragraph */
 p::first-line {
```

```
 font-weight: bold;
 }
</style>
```

```
<p>This is a sample paragraph.</p>
...

```

### ### 14. Selectors Combinators:

Selectors combinators allow you to combine multiple selectors to target specific elements. There are three main types:

- **\*\*Descendant Selector (whitespace):\*\*** Selects all elements that are descendants of a specified element.

**\*\*Code Example:\*\***

```
```html
<style>
  /* Select all <em> elements inside a <p> element */
  p em {
    color: blue;
  }
</style>

```

```
<p>This is a <em>sample</em> paragraph.</p>
...

```

- ****Child Selector (>):**** Selects all direct children of a specified element.

****Code Example:****

```
```html
<style>
 /* Select all direct children elements of */
 ul > li {
 list-style-type: square;
 }
</style>

```

```

 Item 1
 Item 2

...

```

- **Adjacent Sibling Selector (+):** Selects an element that is directly preceded by a specified element.

**Code Example:**

```
```html
<style>
  /* Select the <span> that directly follows an <h2> */
  h2 + span {
    color: green;
  }
</style>

<h2>Title</h2>
<span>This is a span.</span>
```
```

### 15. Responsive Images:

Responsive images adapt to different screen sizes to provide a better user experience. The `max-width: 100%` style ensures that the image doesn't exceed its container width.

**Code Example:**

```
```html
<style>
  /* Responsive image styling */
  img {
    max-width: 100%;
    height: auto;
  }
</style>


```
```

By using the `max-width: 100%` style, the image will shrink to fit its container while maintaining its aspect ratio, ensuring it doesn't overflow or become distorted on smaller screens. This is a simple example, and in a real-world scenario, you might want to use media queries for more sophisticated responsive design.