|  |  |
| --- | --- |
| **Function** | **Description** |
| setTimeout() | Executes a function after a specified time interval |
| setInterval() | Calls a function at specified intervals (repeatedly) |
| Math.random() | Generates a random number between 0 and 1 |
| Math.floor() | Rounds a number down to the nearest integer |
| Math.ceil() | Rounds a number up to the nearest integer |
| Math.round() | Rounds a number to the nearest integer |
| Math.max() | Returns the largest of zero or more numbers |
| Math.min() | Returns the smallest of zero or more numbers |
| typeof() | Returns the type of a variable |
| isNaN() | Checks whether a value is NaN (Not-a-Number) |
| parseFloat() | Parses a string and returns a floating-point number |
| parseInt() | Parses a string and returns an integer |
| decodeURI() | Decodes a Uniform Resource Identifier (URI) component |
| encodeURI() | Encodes a Uniform Resource Identifier (URI) component |
| Date() | Creates a new date object representing a date and time |

JAVASCRIPT FUNCTIONS

|  |  |
| --- | --- |
|  |  |
| setTimeout() | Executes a function after a specified time interval |
| setInterval() | Calls a function at specified intervals (repeatedly) |
| Math.random() | Generates a random number between 0 and 1 |
| Math.floor() | Rounds a number down to the nearest integer |
| Math.ceil() | Rounds a number up to the nearest integer |
| Math.round() | Rounds a number to the nearest integer |
| Math.max() | Returns the largest of zero or more numbers |
| Math.min() | Returns the smallest of zero or more numbers |
| typeof() | Returns the type of a variable |
| isNaN() | Checks whether a value is NaN (Not-a-Number) |
| parseFloat() | Parses a string and returns a floating-point number |
| parseInt() | Parses a string and returns an integer |
| decodeURI() | Decodes a Uniform Resource Identifier (URI) component |
| encodeURI() | Encodes a Uniform Resource Identifier (URI) component |
| Date() | Creates a new date object representing a date and time |

CSS FUNCTIONS

|  |  |
| --- | --- |
| **Function** | Description |
| clip() | Clips an absolutely positioned element |
| **counter-reset()** | Resets the value of a counter |
| **counter-increment()** | Increments the value of a counter |
| **attr()** | Retrieves the value of an attribute of an element |
| **transform-origin()** | Specifies the origin of transformations |
| **cursor()** | Specifies the type of cursor to be displayed |
| **outline()** | Sets the outline properties for an element |
| **text-overflow()** | Specifies how overflowed content in a block should be signaled to the user |
| **word-wrap()** | Allows long words to be able to be broken and wrap onto the next line |
| **pointer-events()** | Specifies how the mouse events should interact with an element |
| **will-change()** | Informs the browser ahead of time of what kinds of changes you are likely to make to an element |
| **perspective()** | Defines the perspective from which all child elements are viewed |
| **image-rendering()** | Sets the image rendering method |
| **font-feature-settings()** | Allows control over advanced typographic features in OpenType fonts |

|  |  |
| --- | --- |
| **Function** | Description |
| rgb() | Specifies colors using red, green, and blue values |
| **rgba()** | Specifies colors using red, green, blue, and alpha values |
| **hsl()** | Specifies colors using hue, saturation, and lightness values |
| **hsla()** | Specifies colors using hue, saturation, lightness, and alpha |
| **url()** | Specifies a URL |
| **calc()** | Performs calculations in CSS values |
| **var()** | Defines a variable in CSS |
| **attr()** | Retrieves the value of an attribute of an element |
| **linear-gradient()** | Creates a linear gradient |
| **radial-gradient()** | Creates a radial gradient |
| **transform()** | Applies a 2D or 3D transformation to an element |
| **transition()** | Specifies how CSS properties should change over time |
| **box-shadow()** | Adds shadow effects around an element |
| **text-shadow()** | Adds shadow effects to text |
| **@keyframes** | Defines animations |
| **grid-template()** | Defines the structure of the grid in grid layouts |
| **flex()** | Specifies the initial sizes of flex items |
| **clip-path()** | Clips an element to a basic shape or polygon |
| **filter()** | Applies visual effects like blurring or color shifting |

HTML TAGS

|  |  |
| --- | --- |
| **Tag** | Description |
| <html> | Root element |
| **<head>** | Contains meta-information about the document |
| **<title>** | Sets the title of the document |
| **<body>** | Contains the content visible to users |
| **<h1> - <h6>** | Headings of various levels |
| **<p>** | Paragraph |
| **<a>** | Anchor for creating hyperlinks |
| **<img>** | Embeds an image |
| **<ul>** | Unordered list |
| **<ol>** | Ordered list |
| **<li>** | List item |
| **<div>** | Division or section in a document |
| **<span>** | Inline container |
| **<table>** | Defines a table |
| **<tr>** | Table row |
| **<td>** | Table cell |
| **<th>** | Table header cell |
| **<form>** | Creates an HTML form for user input |
| **<input>** | Input field for forms |
| **<select>** | Drop-down list |
| **<option>** | Options for select elements |
| **<textarea>** | Multi-line text input |
| **<button>** | Creates a clickable button |
| **<iframe>** | Embeds an external resource in a document |
| **<audio>** | Embeds audio content |
| **<video>** | Embeds video content |
| **<header>** | Represents introductory content |
| **<footer>** | Represents the footer of a document |
| **<nav>** | Defines navigation links |
| **<section>** | Defines a section in a document |
| **<article>** | Represents independent, self-contained content |
| **<aside>** | Defines content aside from the content (sidebar) |
| **<main>** | Contains the main content of the document |
| **<details>** | Defines additional details that the user can view or hide |
| **<summary>** | Defines a visible heading for a **<details>** element |
| **<canvas>** | Used to draw graphics, on the fly, via scripting |
| **<svg>** | Scalable Vector Graphics |
| **<progress>** | Represents the progress of a task |
| **<meter>** | Represents a scalar measurement within a known range |