CSS Tutorial - <https://www.w3schools.com/css/css_syntax.asp>

Here is a list of the most important things to learn in CSS:

1. **Selectors**: <https://www.w3schools.com/css/css_selectors.asp>

* Simple selectors (select elements based on name, id, class)
* [Combinator selectors](https://www.w3schools.com/css/css_combinators.asp) (select elements based on a specific relationship between them)
* [Pseudo-class selectors](https://www.w3schools.com/css/css_pseudo_classes.asp) (select elements based on a certain state)
* [Pseudo-elements selectors](https://www.w3schools.com/css/css_pseudo_elements.asp) (select and style a part of an element)
* [Attribute selectors](https://www.w3schools.com/css/css_attribute_selectors.asp) (select elements based on an attribute or attribute value)

1. **Box Model**: The box model is a core concept in CSS that defines how elements are rendered on the page. It includes properties like **width, height, padding, border,** and **margin**. <https://www.w3schools.com/css/css_boxmodel.asp>
2. **Typography**: CSS provides a wide range of properties to control the appearance of text, such as **font-family, font-size, font-weight, text-align, line-height,** etc
3. **Colors and Backgrounds**:

This includes properties like **color, background-color, background-image,** and **background-size.**

1. **CSS Box Shadows and Borders**:   
   This includes properties like **box-shadow, border, border-radius, and outline**.
2. **Layout and Positioning**:   
   This includes properties like **position, display, float, flexbox,** and **grid**.
3. **Responsive Design**:   
   This includes **media queries, fluid layouts**, and responsive units like **rem** and **em**.
4. CSS **Flexbox**: Flexbox is a powerful layout module that simplifies the creation of flexible and responsive layouts. Understanding how to use properties like display: flex, flex-direction, justify-content, and align-items is essential.
5. CSS **Grid**:CSS Grid is another layout module that allows for more complex and grid-based layouts. Learning how to use properties like display: grid, grid-template-columns, grid-template-rows, and grid-gap can greatly enhance your layout capabilities.
6. CSS **Specificity and Cascade**: Understanding how CSS rules are applied and resolved is important to avoid conflicts and ensure the desired styles are applied. This includes understanding specificity, inheritance, and the cascade.
7. CSS Transforms: Transforms allow you to manipulate the position, size, and shape of elements. This includes properties like translate, rotate, scale, and skew.
8. CSS Variables: CSS variables, also known as custom properties, allow you to define reusable values that can be used throughout your CSS code. They provide flexibility and ease of maintenance.
9. CSS Transitions and Animations: Knowing how to create smooth transitions and animations can add interactivity to your web pages. This includes properties like transition, animation, and keyframes.

FlexBox: <https://www.youtube.com/watch?v=phWxA89Dy94>

Positioning: <https://www.youtube.com/watch?v=MxEtxo_AaZ4>