

Guia Definitivo de CSS e Javascript

CSS [Cascading Style Sheets] é usado para estilizar e layoutar páginas HTML. Aqui está uma lista de propriedades e seletores essenciais:

Seletores CSS

Seletores são usados para "selecionar" elementos HTML e aplicar estilos a eles.

1. Seletores Básicos

- **Seletor de Tag:** Aplica estilos a todos os elementos de um tipo.

```
p {
  color: blue;
}
```

- **Seletor de Classe:** Aplica estilos a elementos com uma classe específica.

```
.destaque {
  font-weight: bold;
}
```

- **Seletor de ID:** Aplica estilos a um único elemento com um ID específico.

```
#cabecalho {
  background-color: yellow;
}
```

2. Seletores Compostos

- **Seletor Descendente:** Aplica estilos a elementos dentro de outros elementos.

```
div p {
  font-size: 18px;
}
```

- **Seletor de Filho Direto:** Aplica estilos apenas a filhos diretos.

```
ul > li {
  list-style-type: square;
}
```

- **Seletor de Irmão Adjacente:** Aplica estilos ao irmão imediatamente seguinte.

```
h1 + p {
  margin-top: 0;
}
```

3. Pseudo-classes e Pseudo-elementos

- **Pseudo-classes:** Estilizam elementos em um estado específico.

```
a:hover {
  color: red;
}
```

- **Pseudo-elementos:** Estilizam partes específicas de um elemento.

```
p::first-line {  
    font-weight: bold;  
}
```

Propriedades CSS

1. Cores e Fundos

- **Cor do Texto:**

```
color: #ff0000; /* Hexadecimal */  
color: rgb(255, 0, 0); /* RGB */  
color: rgba(255, 0, 0, 0.5); /* RGBA com transparência */
```

- **Cor de Fundo:**

```
background-color: #00ff00;
```

- **Imagem de Fundo:**

```
background-image: url('imagem.jpg');  
background-size: cover;  
background-position: center;
```

2. Texto

- **Tamanho da Fonte:**

```
font-size: 16px;
```

- **Família da Fonte:**

```
font-family: Arial, sans-serif;
```

- **Espaçamento entre Linhas:**

```
line-height: 1.5;
```

- **Alinhamento de Texto:**

```
text-align: center;
```

3. Layout

- **Margem:**

```
margin: 10px; /* Todas as direções */  
margin: 10px 20px; /* Superior/Inferior e Esquerda/Direita */  
margin: 10px 20px 30px 40px; /* Superior, Direita, Inferior, Esquerda */
```

- **Preenchimento (Padding):**

```
padding: 10px;
```

- **Borda:**

```
border: 1px solid #000;  
border-radius: 5px; /* Bordas arredondadas */
```

- **Display:**

```
display: block; /* Exibe como bloco */  
display: inline; /* Exibe em linha */  
display: flex; /* Layout flexível */
```

- **Flexbox:**

```
display: flex;  
justify-content: center; /* Alinha horizontalmente */  
align-items: center; /* Alinha verticalmente */
```

- **Grid:**

```
display: grid;  
grid-template-columns: 1fr 1fr 1fr; /* Três colunas de tamanho igual */  
gap: 10px; /* Espaço entre células */
```

4. Posicionamento

- **Posição:**

```
position: relative; /* Relativo ao seu local original */  
position: absolute; /* Relativo ao ancestral posicionado mais próximo */  
position: fixed; /* Fixo na tela */
```

- **Top, Right, Bottom, Left:**

```
top: 10px;  
left: 20px;
```

- **Z-Index:**

```
z-index: 10; /* Controla a ordem de sobreposição */
```

Guia Definitivo de JavaScript

JavaScript é usado para adicionar interatividade e dinamismo aos sites. Aqui está uma lista de conceitos e métodos essenciais:

Conceitos Básicos

1. Variáveis

- **let e const:**

```
let nome = "João";  
const idade = 25;
```

2. Funções

- **Função Declarada:**

```
function saudacao(nome) {  
    return "Olá, " + nome + "!";  
}
```

- **Função Anônima:**

```
const saudacao = function(nome) {  
    return "Olá, " + nome + "!";  
};
```

- **Arrow Function:**

```
const saudacao = (nome) => "Olá, " + nome + "!";
```

3. Estruturas de Controle

- **Condicionais (if, else):**

```
if (idade >= 18) {  
    console.log("Maior de idade");  
} else {  
    console.log("Menor de idade");  
}
```

- **Loops (for, while):**

```
for (let i = 0; i < 5; i++) {  
    console.log(i);  
}
```

Manipulação do DOM

1. Selecionar Elementos

- **querySelector:**

```
const elemento = document.querySelector('.classe');
```

- **querySelectorAll:**

```
const elementos = document.querySelectorAll('p');
```

2. Modificar Elementos

- **Alterar Texto:**

```
elemento.textContent = "Novo texto";
```

- **Alterar HTML:**

```
elemento.innerHTML = "<strong>Novo HTML</strong>";
```

- **Alterar Estilos:**

```
elemento.style.color = "red";
```

3. Eventos

- **Adicionar Evento:**

```
elemento.addEventListener('click', function() {  
    alert("Clicado!");  
});
```

Métodos Úteis

1. Arrays

- **push e pop:**

```
const frutas = ["Maçã", "Banana"];  
frutas.push("Laranja"); // Adiciona ao final  
frutas.pop(); // Remove o último
```

- **map:**

```
const numeros = [1, 2, 3];  
const dobrados = numeros.map(num => num * 2);
```

2. Objetos

- **Acessar Propriedades:**

```
const pessoa = { nome: "João", idade: 25 };  
console.log(pessoa.nome);
```

- **Adicionar Propriedades:**

```
pessoa.profissao = "Desenvolvedor";
```

Próximos Passos

- **CSS:** Explore animações [**@keyframes**], media queries (design responsivo) e frameworks como Bootstrap.
- **JavaScript:** Aprenda sobre APIs, AJAX, e frameworks como React ou Vue.js.