Para estilizar uma class de nome subtitle no css use:  
.subtitle{

}

Para estilizar um id de nome title no css use:  
#title{

}

Em uma página web cada elemento é representado como uma box retangular com as seguintes características:

Diagrama

Descrição gerada automaticamente

A margin é o espaçamento entre elementos html.

A border é uma borda que envolve o padding.

O padding é transparente e é o espaçamento entre a borda e o conteúdo.

O content é o conteúdo em si.

**Estilizando Elementos**

**Padding e Margin:** Temos três formas de estilizar esses elementos.

A primeira é colocando um valor para as partes superior e inferior e depois para os lados esquerdo e direito.

.teste{

padding: 10px 5px

}

O valor de 10 pixels se refere ao eixo Y, ou partes superior e inferior, e os 5 pixels se referem aos lados esquerdo e direito.

A segunda forma é dando valores para cada lado do box.

.teste{

padding: 15px 10px 5px 0

}

Então começamos pelo topo com 15 pixels, passamos o lado direito com 10 pixels, depois para a parte inferior com 5 pixels e por último o lado esquerdo com 0, e sempre nessa ordem. Uma boa dica também é que quando o valor for 0 não precisamos não precisamos colocar a unidade.

A terceira forma é com as propriedades específicas para cada lado, até agora tínhamos visto atalhos para essas propriedades.

.teste{

padding-top: 15px

padding-right: 10px

padding-bottom: 5px

padding-left: 0

}

Essa opção é mais usada quando temos o mesmo valor para 3 lados, e o quarto precisa ter um valor diferente, então usamos o padding com apenas um valor e uma dessas opções para representar o lado diferente.

**Background:** Por enquanto veremos apenas como mudar a cor de fundo. E aqui temos 3 formas de colocar uma cor de fundo, e ainda existem outras.

A primeira é pelo nome da cor em inglês, a segunda é pelo código hexadecimal e a terceira é usando apenas o atalho background.

.teste{

background-color: green

background-color: #000800

background: green

}

**Border:** Vimos que a propriedade border pode ter 3 valores: a largura, a cor e o estilo, mas existem algumas particularidades nisso. A largura pode ser usada com várias unidades, como px, em e mm. A cor pode ser atribuída pelo nome ou por um código hexadecimal, assim como fizemos com o background, e o estilo é representada por palavras-chave, vamos ver algumas delas:

**solid**: mostra uma borda simples e reta.

**dotted**: são bolhinhas com um espaçamento entre elas.

**dashed**: mostra um traçado.

.teste{

border: 10px solid green

}

E se você não quiser usar a propriedade border existem as propriedades específicas para cada aspecto de uma borda, são elas border-width para a largura, border-color para a cor e border-style para o estilo.

.teste{

border-width: 10px

border-color: green

border-style: solid

}

E a última propriedade é o border-radius, ele permite arredondar os cantos de um elemento. Podemos usar várias unidades, mas as mais comuns são os pixels e a porcentagem. Colocando apenas um valor mudamos todos os cantos do elemento, mas seguindo aquela mesma ordem que vimos no padding e margin - topo, direita, inferior e esquerda - conseguimos alterar cada canto separadamente.

.teste{

border-radius: 10px

border-radius: 50%

}

**Estilizando Textos**

1. **font-family 🡪** altera a fonte da escrita.

**font-family: Verdana;**

1. **font-size 🡪** altera o tamanho do texto.

**font-size: 30px;**

1. **font-style 🡪** altera a aparência do texto. Existem alguns tipos: normal, italic.

**font-style: italic;**

1. **font-weight 🡪** altera o peso do texto. Existem alguns tipos: normal, bold.

**font-weight: bold;**

1. **text-transform 🡪** altera o texto entre maiúsculos e minúsculos. Existem alguns tipos: uppercase, lowercase e capitalize.

**text-transform: uppercase;**

1. **text-decoration 🡪** usado para dar destaque. Existem alguns tipos: underline, overline, linethrough.

**text-decoration: underline;**

**Estilizando listas**

1. **list-style-type 🡪** altera o marcador das listas. Existem alguns tipos que são diferentes para listas ordenadas ou não ordenadas: square, “\1F44D”, upper-roman.
2. **list-style-image 🡪** usa uma imagem de marcador.

**list-style-image: url(“rocket.png”);**

**Dimensionamento e Alinhamento**

Width

Height

Max-width

Max-height

Margin

Text-align

**Flexbox**

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**- Introdução ao Flexbox:**

O flexbox foi projetado como um modelo unidimensional e como um modelo de distribuição de espaço entre itens e alinhamento.

1. **Flex Container:** é a tag que envolve os itens. Será nela que usaremos a propriedade “display: flex”, que transformará os seus itens filhos em flex itens.

Desenho de homem e texto

Descrição gerada automaticamente com confiança baixa

- Propriedades relacionadas:

* display 🡪 inicializa o flex no container;
* flex-direction 🡪 faz o direcionamento dos itens (linha ou coluna);
* flex-wrap 🡪 faz a quebra de linha;
* flex-flow 🡪 é uma junção do flex-direction e flex-wrap;
* justify-content 🡪 alinha os itens de acordo com a direção do container;
* align-itens 🡪 alinha os itens de acordo com o eixo do container;
* align-content 🡪 alinha os itens de acordo com a linha do container;

1. **Flex-item:** São os elementos filhos do container. E podem se tornar flex container, caso os flex-itens também tenham elementos filhos.

- Propriedades relacionadas:

* flex-grow 🡪 define o fator de crescimento;
* flex-basis 🡪 define o tamanho inicial do item antes da distribuição do espaço restante no container;
* flex-shrink 🡪 define a capacidade de redução;
* flex 🡪 é uma junção das três propriedades anteriores;
* order 🡪 define a ordem de distribuição desses itens;
* align-self 🡪 define o alinhamento de um item específico do container;