**HTML5 e CSS3**

**Html (1991)**

**Estrutura Básica**



**Semântica**

<section> - Lista de Artigo

<header> - Cabeçalho

<article> - Conteúdo Relevante

<aside> - Conteúdo Relacionado ao conteúdo principal

<footer> - Rodapé

<h1>...<h6>

**Textos e Links**

<p> Conteúdo do artigo </p>

<a></a> Âncora

<a href=”link”> </a>

<a href=”mailto:email”> </a>

<a target=”\_blank”> </a> - Abrir em nova aba

**Imagens**

<img>

<img src=”img/”> - guarda o caminho da imagem

<img alt=”Desc”> - Descrição da foto

**Listas**

<ul> Lista Não Ordenada

<ol> Lista Ordenada

<li> Item da Lista

**CSS3 (1996)**

Seletores – formado por elementos html

Declarações – formado por propriedades e valores

ID x Classe

.header (class)

#header (id) - pode ser usado só 1 vez

Para adicionar css ao html devemos linkar a pagina html com a pagina css “style.css”:

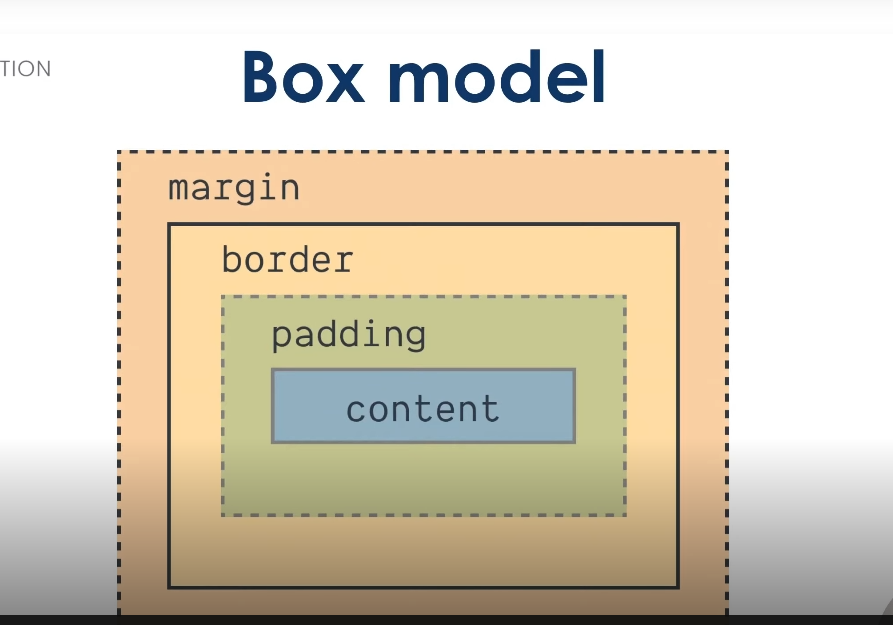
<head>

<meta charset=”utf-8”>

<link rel=”stylesheet” href=”style.css”>

</head>

Conceitos Básicos



Margin – espaçamento entre elementos

Border – ficam em torno do padding e conteúdo

Padding - espaçamento entre a borda e conteúdo

Content – texto, imagem, vídeo...

**Padding e Margin**

Anteriormente usamos o *padding* e o *margin* da forma mais básica, com apenas um valor, mas eles são mais poderosos que isso. Se quisermos atribuir tamanhos diferentes para cada lado do *box* nós podemos, e vamos ver três formas de fazer isso.

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

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*.

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.

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

A propriedade background também é um atalho para várias propriedades, mas isso vocês podem absorver aos poucos, e uma boa opção de leitura é a documentação do MDN.

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.

### 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 bolinhas com um pequeno espaçamento entre elas;

**dashed**: forma uma linha tracejada.

E aproveitando que mostrei esse código temos que falar sobre como separar a estilização dos lados de uma borda.

E se você não quiser usar a propriedade border existem as propriedade 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.

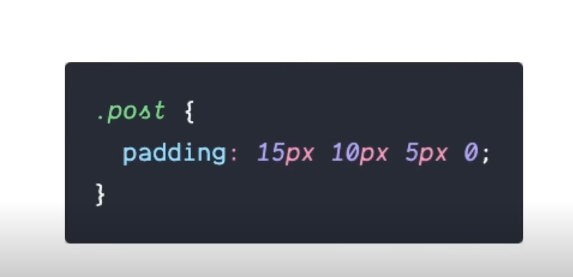
Aqui temos o mesmo código anterior de duas formas diferentes, a primeira com o atalho border e a segunda com cada propriedade específica.

E depois disso podemos juntar os lados com os aspectos de uma borda e criar uma regra mais específica ainda.

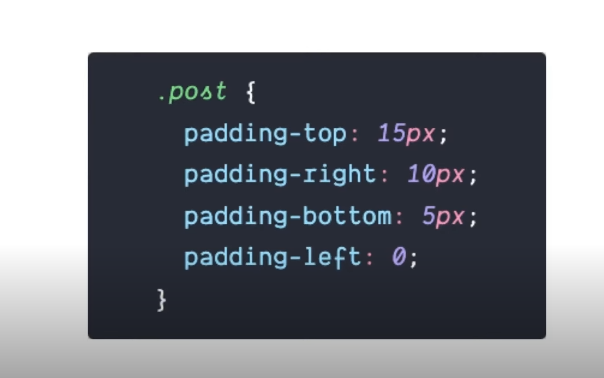
Border-radius

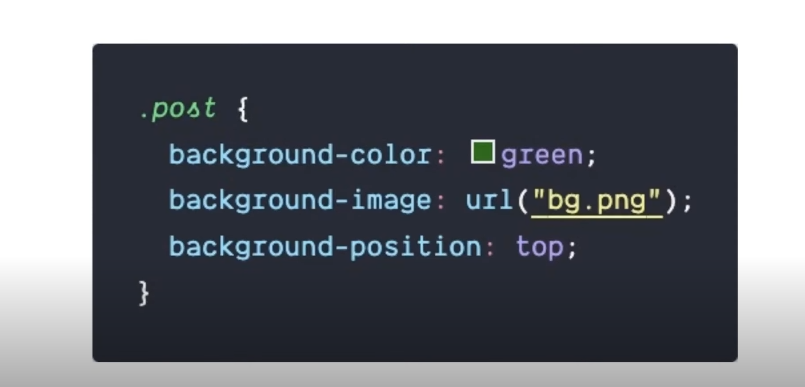
E a última propriedade é o border-radius, ele permite arredondar os cantos de um elemento. Podemos usar várias unidades, mas as mais comuns são os pixels e a porcentagem.

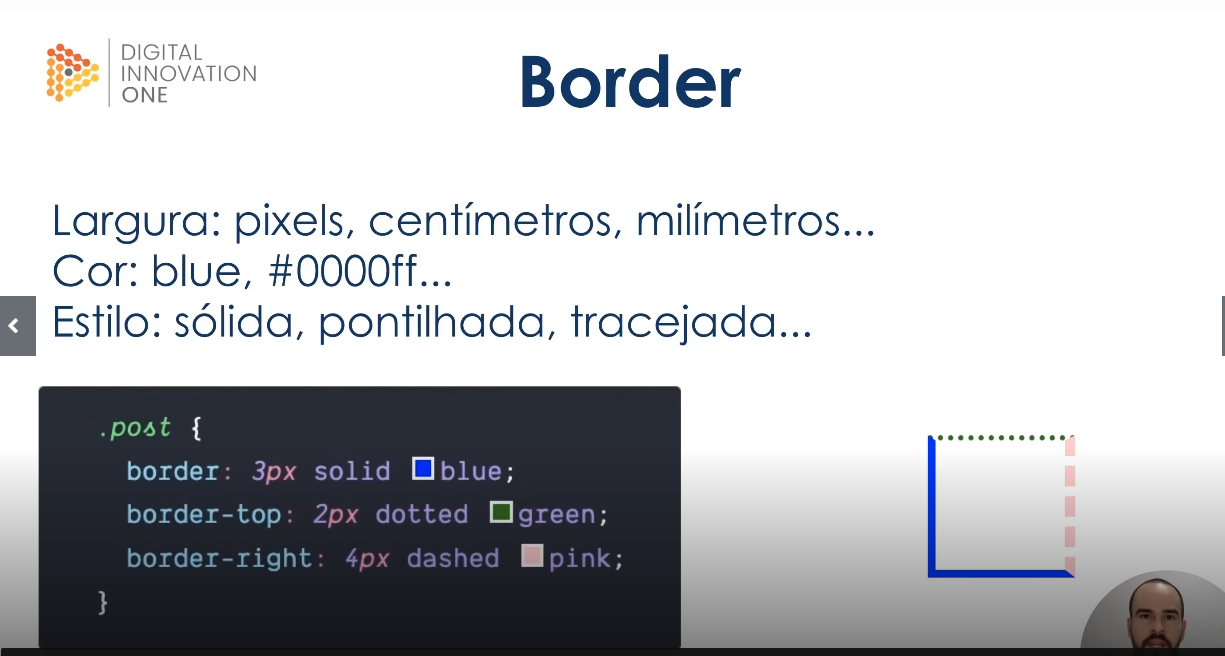
Colocando apenas um valor mudamos todos os cantos do elemento, mas seguindo aquela mesma ordem que vimos no padding e margin - topo, direita, inferior e esquerda -  conseguimos alterar cada canto separadamente.

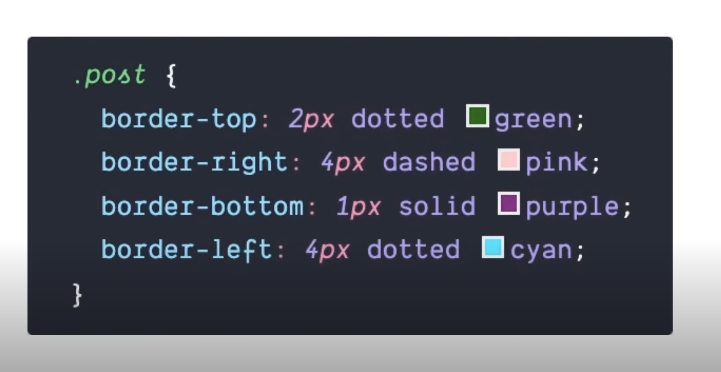


**Padding y,x inferior, esquerdas**











**Fontes**

Font-family – a fonte

Font-size – tamanho

Font-style – estilo (normal,italic)

Font-weight – (normal,bold)

Text-transform – maiúsculas e minúsculas (uppercase, lowercase, capitalize)

Text decoration – (underline,overline,line-through)

**Listas**

List-style-type – tipo de lista

ul (square)

ol (Upper roman)

list-style-image: url(“ img.png”); - mudar para imagem

**Dimensão e Alinhamento**

Width - Largura

Height - Altura

Max-width – maxima largura (responsivo)

Mas-height – maxima altura (responsivo\_

Margin - margem

Text-align – alinhamento de texto

**CSS Flexbox**

Criação de layouts responsivos.

Projeto: Landing Page

**Flex Container**

- É a tag que envolve os itens e nele é aplicado a propriedade display: flex que transforma todos os seus itens filhos em flex itens.

Propriedades relacionadas

- display – inicia container

- flex-direction – direção

- flex-wrap – quebra de linha (um item abaixo do outro)

- flex-flow – abreviação para wrap e Direction

- justify-content – alinha os itens do container de acordo com sua direção

Obs: caso os itens ocupem 100% de todo container, isso não se aplica.

- align-itens – alinha os itens de acordo com o eixo do container

- align-content – alinha as linhas do container

**Flex item**

- São os itens do container

Propriedades:

flex (grow,basis,shrink)

(flex,order,align-self)

Grow – fator de crescimento

Basis – tamanho inicial

Shrink - capacidade de redução

Flex – abrevia os 3 acima

Order- ordem de distribuição

Align-self – alinhamento de um item especifico do container.

Justify content (variações)

Flex-start: inicio do container

Flex-end: final do container

Center: ao centro do container

Space-between: cria um espaçamento igual entre os elementos.

Space-around: os espaçamentos do meio são duas vezes maiores que o inicial e final.

Align-item (tipos)

Center – itens ao centro

Stretch: padrão, flex itens crescem igualmente

Flex-start – alinhar itens no inicio

Flex-end – alinhar itens no final

Baseline – alinhamento de acordo com a linha base da tipografia dos itens