CSS

Herança é uma característica de outro lugar

Agrupamos todas as declarações CSS de um mesmo elemento dentro de um seletor

SELETORES:

.class1.class2 = ambos os elementos

.class1 .class2 = o elemento .class2 que está herdando do class1

div > p = indica que o que será configurado será o p que está dentro de deiv, se houver outro p em um article dentro dessa div, não valerá a configuração.

Div p = TODOS os p que estiverem dentro da div serão configurados, mesmo se houver algum article no caminho

REGRAS

Começam com @, sempre devem estar no começo da folha de css separada ou logo abaixo da tag style

@charset = indica sobre acentuação

@import = um arquivo importado através de um link

@font-face = para fontes baixadas “nome da pasta”/”nome da fonte” = fonte da pasta

Ex:

@font-face {

    font-family: 'idroid.fonte';

    src: url('../fontes/idroid.otf');

}

Há 3 formas de usar as CSS

1º forma: estilo Inline: (apenas use se for muito específica e pontual)

Ex:

<body style=“backgroung-color:’’ lightskyblue; font-family Arial, Helvetica, sans-serif font-size: 20px;’’>

<h1 style=”color: mediumblue; font-size: 2em’’>Capítulo 1</h1>

<p style=’’text-align: justify;’’> Lorem </p>

Vantagens = simples, porta de entrada para os outros estilos

Desvantagens = código poluído, dificuldade se quiser mudar várias partes da página

2º forma: estilo Interno: (use para estilos pequenos)

Ex:

<style>

        body {

            background-color: darkcyan;

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

            font-size: 20px;

            font-style: normal;

        }

        h1 {

            background-color: aquamarine;

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

            font-size: 2em;

        }

        h2 {

            background-color: cadetblue;

        }

    </style>

Isso, dentro da área head

Vantagens = mais organizado

Desvantagens = nada prático se quiser mudar um site que tenha 2 pág Ou mais

3º forma: estilos externos (use mais esse)

Ex:

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="shortcut icon" href="bts buxa.png" type="image/x-icon">

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

    <title>Estilos Externos</title>

</head>

Nesse método, cria-se uma outra página focada somente em CSS, e, posterior a isso, é posto o link, como no exemplo acima.

É possível usar mais de um link

Tem como usar mais de um dos métodos

***PROPRIEDADES CSS***

/\* “qlqr coisa” \*/ = comentário

style = muda o estilo do texto

style + background color + cor de sua escolha = muda a cor da marca texto

na área head, style + { } background color + cor qualquer = muda a cor de todos os marcadores

site para ver as media types existentes iana.com

type na tag ol/ul mostra como a lista vai ser ordenada start = de onde vai começar

1 = com números A = com letras maiúsculas a = com letras minúsculas

I = com números romanos maiúsculos i = com números romanos minúsculos

ul : disc = bola cheia circle = círculo square = quadrado cheio

list-style-position: (normalmente é outside)

outside = pra fora da caixa inside: pra dentro da caixa

list-style-type:

‘\código numérico de emoji’= emoji

1 = com números A = com letras maiúsculas a = com letras minúsculas

I = com números romanos maiúsculos i = com números romanos minúsculos

ul : disc = bola cheia circle = círculo square = quadrado cheio

\0020 ou \00A0 = espaço em branco

start na segunda lista + número subsequente ao do último da primeira ordena a lista como se fosse uma continuação

background-color = cor de fundo font-family = fonte font-size = tamanho da letra

font-style = estilo da letra text-align = alinhamento do texto

background-attachment = deixa as cores fixas na parte baixa da tela

width = largura ou tamanho padding = acolchoamento ( espaço extra praticamente )

border-radius = deixa as bordas mais arredondadas

box-shadow = 1º(->) 2º(baixo) 3º(espalhamento) 4º(cor)

font-style; : estilo da fonte text-decoration: underline; : sublinhado

font-weight; :

Tamanhos literais: lighter, normal, bold, bolder. Aceitos pela maioria das fontes

Tamanhos numéricos: 100 – 900 (de 100 em 100)

Nem todas as fontes permitem essas 3 configurações em CSS3

Text-align = alinhamento de texto

Text-ident = da recuo para os parágrafos, você que escolhe o tanto de recuo

Text-decoration = “muda os apetrechos” do texto text-shadow: sombra, funciona como uma box

Line-height = espaço vertical entre linhas, medida em EM

display: none = mostra nada

content = conteúdo, texto

border:

podemos na cor usar ‘transparent’

solid = linha normal dashed = em riscos dotted = pontilhado

double = linha dupla groove = linha 3D

shorthand pra border: tamanho estilo cor, size style color

padding: top = cima right = direita bottom = baixo left = esquerda

shorthand pra padding e margin: sentido horário

as configurações do outline é da mesma forma que as do border

para centralizarmos uma margin usamos: margin: auto;

se todas as configurações de uma shorthand forem iguais, devemos por somente uma vez a medida

se tivermos posto apenas 2 parâmetros, estaremos com o primeiro configurando a parte de cima e de baixo, e na segunda, a da esquerda e direita

podemos pôr auto em uma shorthand

para calcular o tamanho total de um elemento, deve calcular:

conteúdo + preenchimento + borda + margem

por estarmos calculando as medidas de esquerda e direita juntas, deveremos espelhar a equação, ficando assim:

margem + borda + preenchimento + conteúdo + preenchimento + borda + margem

display:

block = transforma em box-level inline = transforma em inline-level

inlineblock = inline-level com características de box-level, configuração de altura e largura.

Box-shadow: sombra da caixa. A lógica das sombras é igual uma reta numérica, ou seja se a contagem estiver negativa, a sombra está na esquerda, positiva na direita, isso no caso da sombra horizontal

Shorthand: horizontal vertical espalhamento cor

Podemos também criar uma sombra a partir do devtools, vamos na parte roxa antes da declaração, clicamos nela e moldamos a sombra.

Se quisermos mudar algo na sombra em si, isso deve ser a transparência, não a cor em si

Border-radius = arredondamento da borda

Shorthand: cima esquerda, cima direita, baixo direita, baixo esquerda

Se for posto apenas dois parâmetros, será na prática em formato de X, ou seja, cima esquerda baixo direita e cima direita e baixo esquerda

Podemos usar porcentagens para transformar objetos quadrados em redondos

Para colocar bordas á uma caixa temos de definir sua largura e usar margin: auto; pra centralizar

Position:

É usada para definir como os elementos estarão visíveis na nossa página

Vai estar por padrão como static

Elementos posicionados sem usar o static tendem a ficar sobre elementos não posicionados

Static:

Ignora as propriedades auxiliares: top, right, bottom e left

Tem o canto superior esquerdo como referência

Segue o fluxo dos outros elementos da página normalmente, um após o outro

A ordem que aparece no navegador é a mesma que aparece no código

Fixed:

Fica fixo em relação á pagina

Vai ser posicionado de maneira relativa ao tamanho da janela em que o site é exibido

Mesmo com rolamento o elemento não se move na tela

Continua no mesmo lugar independentemente da rolagem vertical

Ficará onde está independente do tamanho da tela

Sticky:

Fica fixo ao rolamento da tela.

Precisa ser posto as propriedades auxiliares, porque precisa de um posicionamento como referência

Se estiver dentro de uma div só vai acompanhar o rolamento da página até o fim dessa div

Relative:

Tem como referência sua posição original, e é relativa a ela.

Não afeta as tags que estão abaixo dela

Absolute:

Há dois tipos de comportamento, sendo eles:

1-elemento pai sendo diferente de static:

Vai ter o position do pai como referência

2-sem elemento pai ou este sendo static:

Vai ignorar os elementos acima e vai se posicionar no canto superior esquerdo da tela, podendo até mesmo sobrepor a outros elementos

***Bordas personalizadas:***

Deve haver uma borda normal configurada para que esta seja então personalizada

Border-image-source: url (‘nome da imagem’) = qual será a imagem

Border-image-slice = a divisão dos pedaços da borda (contagem em números)

Border-image-repeat = como vão estar distribuídos os pedaços da borda

Shorthand para border-image:

Border-image = source slice repeat

HERANÇA:

É um elemento passar aos elementos de baixo suas características

PSEUDO-CLASSES:

Mudam o estilo do conteúdo de acordo com ações do usuário.

:hover = o que ocorre ao passar o cursor por cima do elemento.

podemos configurar um parágrafo, e mostra-alo através do :hover, da seguinte maneira

p {  
  display: none;  
  background-color: yellow;  
  padding: 20px;  
}  
  
div:hover p {  
  display: block;  
}

repare que, primeiro configuramos o parágrafo, e depois o :hover de uma div mostrando o parágrafo.

Podemos usar a propriedade transition-duration para definirmos em quanto tempo as configurações do hover (e de outras pseudo-classes) aparecerão e desaparecerão, deve ser configurado não no hover, mas, sim no elemento original. Só é aceito valor de tempo em segundo

:visited = o que ocorre quando o usuário já visitou esta parte.

:link = link não visitado

:active = link ativo, muda quando o usuário está com o site aberto

**Note:** a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective! a:active MUST come after a:hover in the CSS definition in order to be effective! Pseudo-class names are not case-sensitive.

PSEUDO-ELEMENTOS

Mudam especificamente o elemento.

::first-line = primeira linha

**Note:** The ::first-line pseudo-element can only be applied to block-level elements.

The following properties apply to the ::first-line pseudo-element:

* font properties
* color properties
* background properties
* word-spacing
* letter-spacing
* text-decoration
* vertical-align
* text-transform
* line-height
* clear

\* { /\* Configurações Globais das CSS \*/

            height: 100%;

        }

*REPRESENTAÇÕES CORES EM CSS*

<!--Representação por nomes-->

    <h2 style="background-color: black; color: cyan;" >Exemplo de cores</h2>

    <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Itaque dicta enim vitae autem repellendus delectus soluta aliquid, odio ipsa nostrum impedit amet reprehenderit consectetur doloremque debitis. Reprehenderit repellendus vero dicta?</p>

    <!--Representação hexadecimal

    A escala é de 0 a F, ou seja 0 1 2 3 4 5 6 7 8 9 A B C D E F, conforme sua intensidade. Uma cor é composta de 3 partes, sendo, vermelha, verde e azul. Os números e letras, vão representar o quanto de cada uma dessas cores irá formar uma outra.-->

    <h2 style="background-color: #490640; color: #c2c2c28c;">Exemplo de cores</h2>

    <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Itaque dicta enim vitae autem repellendus delectus soluta aliquid, odio ipsa nostrum impedit amet reprehenderit consectetur doloremque debitis. Reprehenderit repellendus vero dicta?</p>

    <!--Essa contagem é feita de 0 a 255, com a mesma lógica dos hexadecimais. É possível também, moldar cores com rgb, ao passarmos, nosso mouse por cima da cor posta com o nome. Á esquerda escolhemos a tonalidade, á direita a cor.-->

    <h2 style="background-color: rgb(255, 82, 2); color: rgb(240, 230, 00);">Exemplo de cores</h2>

    <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Itaque dicta enim vitae autem repellendus delectus soluta aliquid, odio ipsa nostrum impedit amet reprehenderit consectetur doloremque debitis. Reprehenderit repellendus vero dicta?</p>

    <!--Representção por HSL

    Hue = Matiz   Saturation = Saturação   Luminosity = Luminosidade

A = Alfa (Transparência)-->

    <h2 style="background-color: hsl(277, 76%, 8%); color: rgb(92, 92, 255);">Exemplo de cores</h2>

    <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Itaque dicta enim vitae autem repellendus delectus soluta aliquid, odio ipsa nostrum impedit amet reprehenderit consectetur doloremque debitis. Reprehenderit repellendus vero dicta?</p>

***Degradê em CSS***

body {

            background-image: linear-gradient(180deg, #4E28E0, #C728E0);

        }

deg = graus

to right/left = direita, esquerda to top = p/ cima to bottom = p/ baixo

radial-gradient + circle = faz o degradê ser em círculo.

porcentagem depois da cor indica o quanto de espaço essa cor vai usar.

Podemos na cor colocar transparent

*REPRESENTAÇÃO FONTES EM CSS*

/\*

        Medidas Absolutas

        cm, mm, in (polegada), px (recomendada pela w3c e mais usada para fontes), pt (não é boa de se usar, ponto é só para documentos que vao ser impressos), pc. Não são recomendadas, a medida mostrada na tela do seu usuário pode alterar.

        Mediadas Relativas

        em (recomendada pela w3c. Relativo ao tamanho da fonte atual, geralmente é 16px, ou seja, 1em = 16px. Normalmente é a altura do m maiúsculo)

        ex (relativo à altura x)

        rem (relativo ao tamanho da fonte que está configurada no body ou \*)

        vw (relativo à largura da sua tela, porcentagem da largura)

        vh (relativo à altura da tela, porcentagem da altura)

        %

        \*/

shorthand é uma compactação de uma declaração, que é feita na seguinte ordem:

font: font-style font-weight font-size font-family

caso não haja nenhuma configuração em uma dessas áreas, será usada a padrão

para usar fontes importadas, devemos ter o @import dela, para que assim tenhamos o acesso a ela

a parte que importa do @import, é o nome da fonte que vem após o “=”

SELETORES PERSONALIZADOS

Id = identificador, é posto dentro da tag html, com um nome, onde esse nome vai ser usado nas configurações css sem a necessidade de pôr qual é a tag, usando “#” antecedendo o nome não é preciso reconfigurar de acordo com a tag genérica, o id é apenas configurações adicionais

Em HTML id = CSS é # Em HTML class = CSS é .

Class é como se fosse um id, porém, este sendo usado para agrupar configurações, em outras palavras, pode ter em mais de 1 elemento num documento HTML

Dar nomes às classes e id com base em suas funções

É possível agrupar mais de uma classe dentro de um parâmetro class, sem precisar de vírgula

É possível usar um id e um class na mesma tag, porém o id sobrepõe

Id se sobrepõe sobre o class e configurações genéricas

Tanto class como id herdarão características da camada acima (não o código, mas sim a configuração da tag que veio anteriormente)

VARIÁVEIS

Servem para guardar valores que serão usados futuramente

Usamos a pseudo classe root, seguindo o exemplo:

COMPACTANDO VÍDEOS:

Usamos de artifícios para isso, já que o iframe já vem do site de origem com algumas configurações

Colocamos o vídeo dentro de uma div com uma classe e então configuramos essa classe

Nessa classe configuraremos o position: relative

Depois configuraremos o iframe que está dentro da div com position: absolute e configuramos também as propriedades auxiliares top e right. Além claro de width e heigt.

div.video{

    background-color: var(--cor7);

    margin: 20px -20px 20px -20px;

    padding: 20px;

    padding-bottom: 55%;

    position: relative;

}

div.video > iframe{

    position: absolute;

    top: 5%;

    left: 5%;

    width: 90%;

    height: 90%;

}