

MATERIAL DE APOIO:

CSS

por: Koding

SUMÁRIO

CSS	3
CAPÍTULO 1: CSS – INTRODUÇÃO	3
Aula 00 Fundamentos do CSS	3
Aula 01 – Estilização de Layout com CSS	4
Aula 02: Estilização Avançada com CSS	7
CAPÍTULO 2: CSS – ANIMAÇÕES E TRANSIÇÕES	11
Aula 00 – Conceitos Básicos de Animações em CSS	11
Aula 01 – Transições em CSS para Efeitos de Hover	12
Aula 02 – Animações Avançadas em CSS	13
CAPÍTULO 3: CSS – LAYOUTS FLEXÍVEIS	14
Aula 00 – Introdução a Layouts Flexíveis em CSS	14
Aula 01 – Layouts de Grade em CSS	15
Aula 02 – Layouts Flexíveis Avançados e Combinação de Técnicas	15
GABARITO DO JOGO	16
REFERÊNCIAS	17

CSS

CSS (Cascading Style Sheets ou Folhas de Estilo em Cascata) é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML ou em XML. O CSS descreve como elementos são mostrados na tela, no papel, na fala ou em outras mídias.

CAPÍTULO 1: CSS – INTRODUÇÃO

Neste capítulo introdutório sobre CSS, seu filho explorará os fundamentos e estilização essenciais para a criação de páginas web.

Aula 00 Fundamentos do CSS

O QUE É CSS:

CSS, ou Cascading Style Sheets, é uma linguagem de estilo usada para descrever a apresentação de um documento HTML. Ele define como os elementos HTML devem ser exibidos na tela, incluindo layout, cores, fontes e outros aspectos visuais.

SINTAXE BÁSICA:

O CSS utiliza uma sintaxe simples de seletores e propriedades.

```
1  /* Seleciona todos os parágrafos e define a cor
2  do texto para azul */
3  p {
4    color: blue;
5  }
```

Aula 01 – Estilização de Layout com CSS

PRINCIPAIS SELETORES:

Seletor De Tag

Dentre os seletores do CSS possuímos o seletor de tag, neste seletor usamos a própria tag para estilizar o documento.

```
1 p {  
2   color: blue;  
3 }
```

Neste exemplo utiliza-se a tag <p> para exemplificar o uso do seletor de tag do CSS, porém se houver mais de uma tag <p> a estilização será aplicada em todos os elementos de parágrafo do documento.

```
1 h1, p, a {  
2   color: blue;  
3 }
```

O exemplo acima mostra que é possível selecionar múltiplos elementos para que seja aplicada uma mesma estilização nos mesmos, para isso basta que separemos os elementos que queremos aplicar a estilização com a vírgula.

Seletor De Id

Neste próximo seletor básico, utiliza-se o próprio atributo global id, como identificador para estilizar o documento. Este seletor é utilizado, quando desejamos identificar um elemento específico no nosso documento, seu uso é aconselhado em elementos específicos e únicos. Para sua representação no CSS atribuímos o “#” (cerquilha).

```
1 #id{  
2   color: blue;  
3 }
```

Seletor De Classe

Neste último seletor, é usado o próprio atributo global class do HTML para estilizar o documento. Estas classes permitem ao CSS selecionar múltiplos elementos ao mesmo tempo, desde que eles possuam a mesma classe. Para sua representação no CSS atribuímos o “.” (ponto final).

```
1 .classe{  
2   color: blue;  
3 }
```

ESTILIZAÇÃO DE TEXTO:

Text-Align

Esta propriedade destina-se a definir o alinhamento horizontal do texto.

Os principais valores para esta propriedade são.

Left - Alinha o texto à esquerda do box;

Right - Alinha o texto à direita do box;

Center - Alinha o texto no centro do box;

Justify - Alinha o texto de modo que cada linha se estenda da esquerda à direita do box.

Text-Transform

Esta propriedade destina-se a definir os efeitos de capitalização (maiúsculas, minúsculas) do texto.

Os principais valores para esta propriedade são.

Capitalize - Causa o efeito de transformar as primeiras letras das palavras do texto em maiúscula;

Uppercase - Causa o efeito de transformar todas as letras das palavras do texto em maiúscula;

Lowercase - Causa o efeito de transformar todas as letras das palavras do texto em minúsculas.

Text-Decoration

Esta propriedade destina-se a definir uma linha decorativa para o texto ou fazer com que o texto "pisque".

Os principais valores para esta propriedade são.

Underline - Causa o efeito sublinhado. A linha decorativa é desenhada sob o texto;

Overline - Causa o efeito superlinhado. A linha decorativa é desenhada sobre o texto;

line-through - Causa o efeito riscado. A linha decorativa é desenhada riscando o texto;

blink - Causa o efeito de texto "piscante". Critérios de acessibilidade fazem com que a maioria dos navegadores não implementem este efeito.

Font-Family

Define a fonte que será usada em determinada parte do documento.

Font-Size

Define o tamanho da fonte que será usado em determinada parte do documento.

ESTILIZAÇÃO DE CORES:

O CSS oferece várias formas de definir cores, seja por nome, código hexadecimal, RGB, ou HSL.

Aula 02: Estilização Avançada com CSS

POSICIONAMENTO:

Todo elemento estático é posicionado alinhado pelo canto superior esquerdo no corpo do documento. Para alterá-lo será necessário utilizar a propriedade “position”, sendo possível atribuir 5 valores, que são:

static - É estático, apenas segue o fluxo junto dos outros elementos da página;

fixed - Faz com que o elemento que recebeu esta propriedade não se mova na tela;

sticky - Ele fica fixo em relação ao rolamento da página;

relative - É usado quando queremos alterar a posição de um elemento tendo como referência a posição inicial dele;

absolute - Quando o elemento com essa propriedade possui um elemento pai de valor diferente de static, ele terá este elemento pai como referência para ser posicionado. Caso essa situação não ocorra, ele irá ignorar estes elementos e será posicionado a partir do canto superior esquerdo superior do documento.

Alterando o Position, pode-se utilizar outras quatro propriedades auxiliares que são: top, bottom, left e right. Apenas o position static ignora o valor dessas propriedades. Elas podem ser utilizadas com todos os outros tipos de positions possibilitando definir a posição de um elemento. As referências de topo, inferior, esquerda e direita podem mudar dependendo do tipo de position aplicado.

DISPLAY E FLOAT:

A propriedade `display` define os tipos de exibição internos e externos de um elemento.

block - O elemento gera uma caixa de elemento de bloco, gerando quebras de linha antes e depois do elemento quando no fluxo normal.

inline - O elemento gera uma ou mais caixas de elemento em linha que não geram quebras de linha antes ou depois de si mesmas.

flex - O elemento se comporta como um elemento de bloco e apresenta seu conteúdo de acordo com o modelo flexbox.

grid - O elemento se comporta como um elemento de bloco e apresenta seu conteúdo de acordo com o modelo de grade.

A propriedade `float` determina que um elemento deve ser retirado do seu fluxo normal e colocado ao longo do lado direito ou esquerdo do seu container, onde textos e elementos em linha irão se posicionar ao seu redor.

left - É uma palavra-chave que indica que o elemento deve flutuar à esquerda do bloco;

right - É uma palavra-chave que indica que o elemento deve flutuar à direita do bloco;

none - É uma palavra-chave que indica que o elemento não deve flutuar;

inline-start - É uma palavra-chave que indica que o elemento deve flutuar no lado inicial do seu bloco;

inline-end - É uma palavra-chave que indica que o elemento deve flutuar no lado final do seu bloco.

PSEUDO-CLASSES E PSEUDO-ELEMENTOS:

Uma pseudo-classe CSS é uma palavra-chave adicionada a seletores que especifica um estado especial do elemento selecionado. Por exemplo, `:hover` pode ser usado para alterar a cor de um botão quando o usuário passar o cursor sobre ele.

Um pseudo-elemento CSS é uma palavra-chave adicionada a um seletor que permite que você estilize uma parte específica do elemento selecionado. Por exemplo, o pseudo-elemento `::first-line` aplica o estilo apenas na primeira linha de um parágrafo.

CAPÍTULO 2: CSS – ANIMAÇÕES E TRANSIÇÕES

Neste capítulo seu filho irá conhecer os conceitos básicos de animações, transições e animações avançadas em CSS.

Aula 00 – Conceitos Básicos de Animações em CSS

KEYFRAMES

Keyframes são a base de Animações CSS. Eles definem como a animação será em cada estágio de sua linha do tempo. Em CSS, trabalha-se com keyframes através da diretiva `@keyframes`. As propriedades `animation-*` são utilizadas para referenciar os `@keyframes` a um (ou mais) seletor(es) CSS para trabalhar as características da animação.

animation-name - nome atribuído no `@keyframes`

animation-duration - O tempo de duração da animação

animation-timing-function - velocidade com que a animação é executada (easing)

animation-delay - o delay para iniciar a animação

animation-iteration-count - quantas vezes a animação será executada

animation-direction - altera a forma como a animação é executada (ex: do começo para o fim ou do fim para o começo)

animation-fill-mode - determina o que acontecerá ao término da animação (se retorna ao ponto original, se fica como terminou etc.)

animation-play-state – especifica se a animação está acontecendo ou se está pausada.

Aula 01 – Transições em CSS para Efeitos de Hover

As transições em CSS são uma técnica poderosa para criar efeitos visuais suaves e agradáveis durante interações do usuário. Quando aplicadas a elementos HTML, as transições permitem que você defina como as propriedades de estilo devem mudar ao longo do tempo, criando uma animação gradual.

As transições envolvem a definição de propriedades CSS que experimentarão uma mudança suavizada.

Normalmente, os efeitos de hover são aplicados a elementos interativos, como links, botões ou áreas específicas de uma página.

Ao combinar a seleção do elemento com a propriedade `transition`, é criado um efeito de hover suave. Isso faz com que a mudança de estilo ocorra gradualmente, proporcionando uma experiência visual mais agradável.

Além da mudança de cor, é possível aplicar transições a várias outras propriedades, como fundo, tamanho, posição, etc., dependendo do efeito desejado.

```
1  .elemento {  
2    color: #333; /* Cor padrão */  
3    transition: color 0.3s ease-in-out;  
4    /* Transição suave ao longo de 0.3 segundos  
5    */  
6  }  
7  
8  .elemento:hover {  
9    color: #ff5733;  
10   /* Nova cor quando em hover */  
11  }  
12
```

Aula 02 – Animações Avançadas em CSS

Animações CSS tornam possível animar transições de um estilo CSS para outro. Algumas de suas propriedades avançadas são:

animation-timing-function - Configura a sincronização da animação; que é, como a animação transita por keyframes, por estabilizar curvas de aceleração;

animation-fill-mode - Configura que valores são aplicados pela animação antes e depois de se executar;

animation-iteration-count - Configura o número de vezes que uma animação deveria se repetir; você pode especificar infinito para repetir a animação indefinidamente.

CAPÍTULO 3: CSS – LAYOUTS FLEXÍVEIS

Neste capítulo seu filho irá aprender sobre layouts flexíveis, em grade e combinação de técnicas.

Aula 00 – Introdução a Layouts Flexíveis em CSS

FLEXBOX:

O Flexbox é um módulo completo e não uma única propriedade; algumas delas devem ser declaradas no container (o elemento-pai, que chamamos de flex container), enquanto outras devem ser declaradas nos elementos-filhos (os flex itens). Algumas de suas propriedades são:

display – Esta propriedade define um flex container; inline ou block dependendo dos valores passados. Coloca todos os elementos-filhos diretos num contexto Flex;

flex-direction – Estabelece o eixo principal, definindo assim a direção em que os flex items são alinhados no flex container;

- **row**: esquerda para a direita em left to right, direita para a esquerda em right to left;
- **column**: mesmo que row, mas de cima para baixo.

flex-wrap – Com esta propriedade é possível permitir que os itens quebrem para uma linha seguinte conforme for necessário.

- **wrap**: os flex items vão quebrar em múltiplas linhas, de cima para baixo.

Aula 01 – Layouts de Grade em CSS

Grids podem ser usados para o design de layouts de grandes seções de uma webpage, assim como de pequenos elementos de interface.

O Grid Layout oferece controle preciso sobre o alinhamento e espaçamento de itens dentro das células da grade. Você pode ajustar a posição vertical e horizontal, bem como adicionar espaçamento entre as linhas e colunas.

Aula 02 – Layouts Flexíveis Avançados e Combinação de Técnicas

Uma abordagem poderosa para layouts flexíveis é combinar as capacidades do Flexbox e do Grid. Enquanto o Grid oferece uma estrutura de grade mais complexa, o Flexbox é útil para o alinhamento de itens em uma única dimensão (linha ou coluna).

GABARITO DO JOGO

FASE 1:

Situação: **Ajude a alinhar 3 (três) colunas usando uma combinação de flex-direction e flex-wrap.**

Resposta:

Flex-direction: column; Flex-wrap: wrap;

FASE 2:

Situação: **Ajude a alinhar 3 (três) colunas usando uma combinação de align-items e flex-wrap.**

Resposta:

Align-items: center; Flex-wrap: wrap;

REFERÊNCIAS

<https://developer.mozilla.org/pt-BR/docs/Web/CSS>

<https://www.treinaweb.com.br/blog/seletores-basicos-do-css>

<https://maujor.com/tutorial/propriedades-css-para-estilizacao-de-textos.php>

<https://www.alura.com.br/artigos/entenda-a-propriedade-position-css>

<https://developer.mozilla.org/pt-BR/docs/Web/CSS/float>

<https://developer.mozilla.org/pt-BR/docs/Web/CSS/display>

<https://developer.mozilla.org/pt-BR/docs/Web/CSS/Pseudo-classes>

<https://developer.mozilla.org/pt-BR/docs/Web/CSS/Pseudo-elements>

<https://desenvolvimentoparaweb.com/css/animacoes-css-introducao/>

https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_animations/Using_CSS_animations

<https://www.alura.com.br/artigos/css-guia-do-flexbox>

<https://developer.mozilla.org/pt->

BR/docs/Web/CSS/CSS_grid_layout/Basic_concepts_of_grid_layout