

CSS

LAYOUT PARA COMPUTADOR

CSS básico, Box Model, Flexbox e organização de páginas desktop

```
.pagina {  
  max-width: 1200px;  
  margin: 0 auto;  
  display: flex;  
}
```



Fundação Arco-Íris de Araputanga
FCARP
FACULDADE CATÓLICA
RAINHA DA PAZ

Professor Thiago Dias Negrão

Aula 04-1 2026



OBJETIVOS DA AULA

- Conectar o arquivo CSS ao documento HTML
- Aplicar seletores, cores, fontes e espaçamentos
- Compreender o Box Model
- Organizar cabeçalho, conteúdo e cartões com Flexbox
- Construir uma página completa para computador

Resultado esperado: uma página desktop organizada e pronta para receber responsividade.

HTML E CSS

HTML

- Organiza o conteúdo
- Define títulos, textos e imagens
- Cria a estrutura da página



CSS

- Controla cores e fontes
- Define tamanhos e espaçamentos
- Organizar o layout visual

FORMAS DE APLICAR CSS

<!-- Inline -->

```
<h1 style="color: blue;">Título</h1>
```

<!-- Interno -->

```
<style>
```

```
  h1 { color: blue; }
```

```
</style>
```

<!-- Externo -->

```
<link rel="stylesheet" href="estilo.css">
```

- Inline: dentro da tag
- Interno: dentro do head
- Externo: arquivo separado
- Para o projeto, use CSS externo

Teste cada alteração e observe o resultado no navegador.

LIGAÇÃO COM O ARQUIVO CSS

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>Minha empresa</title>
  <link rel="stylesheet"
href="estilo/estilo.css">
</head>
```

- O caminho precisa estar correto
- O arquivo pode ficar em uma pasta
- Use nomes sem espaços
- Salve antes de atualizar

Teste cada alteração e observe o resultado no navegador.

SELETORES CSS

```
p {  
  color: #333333;  
}
```

```
.destaque {  
  color: #0066cc;  
}
```

```
#titulo-principal {  
  font-size: 42px;  
}
```

- Elemento: todas as tags iguais
- Classe: pode repetir
- ID: identifica um elemento
- Prefira classes para estilos

CORES E FUNDOS

```
body {  
  color: #222222;  
  background-color: #f2f5f7;  
}  
  
header {  
  color: white;  
  background-color: #173b66;  
}
```

- color altera o texto
- background-color altera o fundo
- Hexadecimal usa #
- Mantenha contraste legível

FONTES E TEXTOS

```
body {  
  font-family: Arial, sans-serif;  
  font-size: 18px;  
  line-height: 1.6;  
}
```

```
h1 {  
  font-size: 48px;  
  text-align: center;  
}
```

- font-family escolhe a fonte
- font-size define o tamanho
- line-height melhora a leitura
- text-align controla o alinhamento

LARGURA E ALTURA

```
.container {  
  width: 90%;  
  max-width: 1200px;  
  min-height: 400px;  
  margin: 0 auto;  
}
```

- width pode usar pixels ou porcentagem
- max-width evita conteúdo largo demais
- min-height define uma altura mínima
- margin auto centraliza o container

Teste cada alteração e observe o resultado no navegador.

MARGIN E PADDING

```
.cartao {  
  margin: 20px;  
  padding: 24px;  
  background-color: white;  
}
```

/ fora = margin | dentro = padding */*

- Margin cria espaço externo
- Padding cria espaço interno
- Aceitam até quatro valores
- Use espaçamento consistente

BORDAS E CANTOS

```
.cartao {  
  border: 1px solid #d8e0e8;  
  border-radius: 12px;  
  box-shadow: 0 4px 12px rgba(0, 0,  
0, 0.12);  
}
```

- border define a borda
- border-radius arredonda os cantos
- box-shadow cria profundidade
- Evite sombras muito fortes

BOX MODEL

Camadas

- Conteúdo no centro
- Padding ao redor do conteúdo
- Borda envolve o padding
- Margin separa os elementos

Cálculo

- Largura final inclui todas as camadas
- box-sizing simplifica o cálculo
- Use border-box no projeto
- Inspecione no navegador

BOX SIZING

```
* {  
  box-sizing: border-box;  
}  
  
.caixa {  
  width: 300px;  
  padding: 20px;  
  border: 2px solid #173b66;  
}
```

- O width inclui conteúdo, padding e borda
- Evita caixas maiores que o esperado
- Aplique no seletor universal
- Facilita o controle do layout

DISPLAY

```
span { display: inline; }  
section { display: block; }  
.menu { display: flex; }  
.oculto { display: none; }
```

- block ocupa a linha disponível
- inline acompanha o texto
- flex organiza os elementos
- none remove da exibição

INTRODUÇÃO AO FLEXBOX

```
.linha {  
  display: flex;  
  gap: 24px;  
}
```

```
.linha > div {  
  flex: 1;  
}
```

- O elemento pai recebe display flex
- Os filhos ficam lado a lado
- gap cria espaço entre os filhos
- flex distribui o espaço

EIXOS DO FLEXBOX

```
.menu {  
  display: flex;  
  flex-direction: row;  
  justify-content: space-between;  
  align-items: center;  
}
```

- row cria o eixo horizontal
- justify-content atua no eixo principal
- align-items atua no eixo transversal
- space-between separa as extremidades

Teste cada alteração e observe o resultado no navegador.

CABEÇALHO PARA DESKTOP

```
header .container {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  min-height: 90px;  
}  
  
nav { display: flex; gap: 28px; }
```

- Logo à esquerda
- Menu à direita
- Itens alinhados ao centro
- Container limita a largura

ÁREA PRINCIPAL E CARTÕES

```
.hero {  
  display: flex;  
  align-items: center;  
  gap: 50px;  
}
```

```
.servicos {  
  display: flex;  
  gap: 24px;  
}
```

```
.cartao { flex: 1; }
```

- Texto e imagem lado a lado
- Três cartões na mesma linha
- Espaços definidos por gap
- Distribuição igual com flex

PROJETO DESKTOP

ATIVIDADE PRÁTICA

- Crie index.html e estilo/estilo.css
- Monte header, nav, main, section e footer
- Crie uma área principal com texto e imagem
- Adicione três cartões de serviços
- Use container, Box Model e Flexbox

ENTREGA

Uma página para computador com cabeçalho, conteúdo principal, três cartões e rodapé.

Salve, abra no navegador e confira antes de enviar.

REVISÃO DA AULA

RESPONDA SEM CONSULTAR

- Qual é a função do CSS?
- Qual é a diferença entre classe e ID?
- O que muda entre margin e padding?
- Quais camadas formam o Box Model?
- Como colocar elementos lado a lado com Flexbox?

Próxima etapa: adaptar o mesmo projeto para telas de celular.