

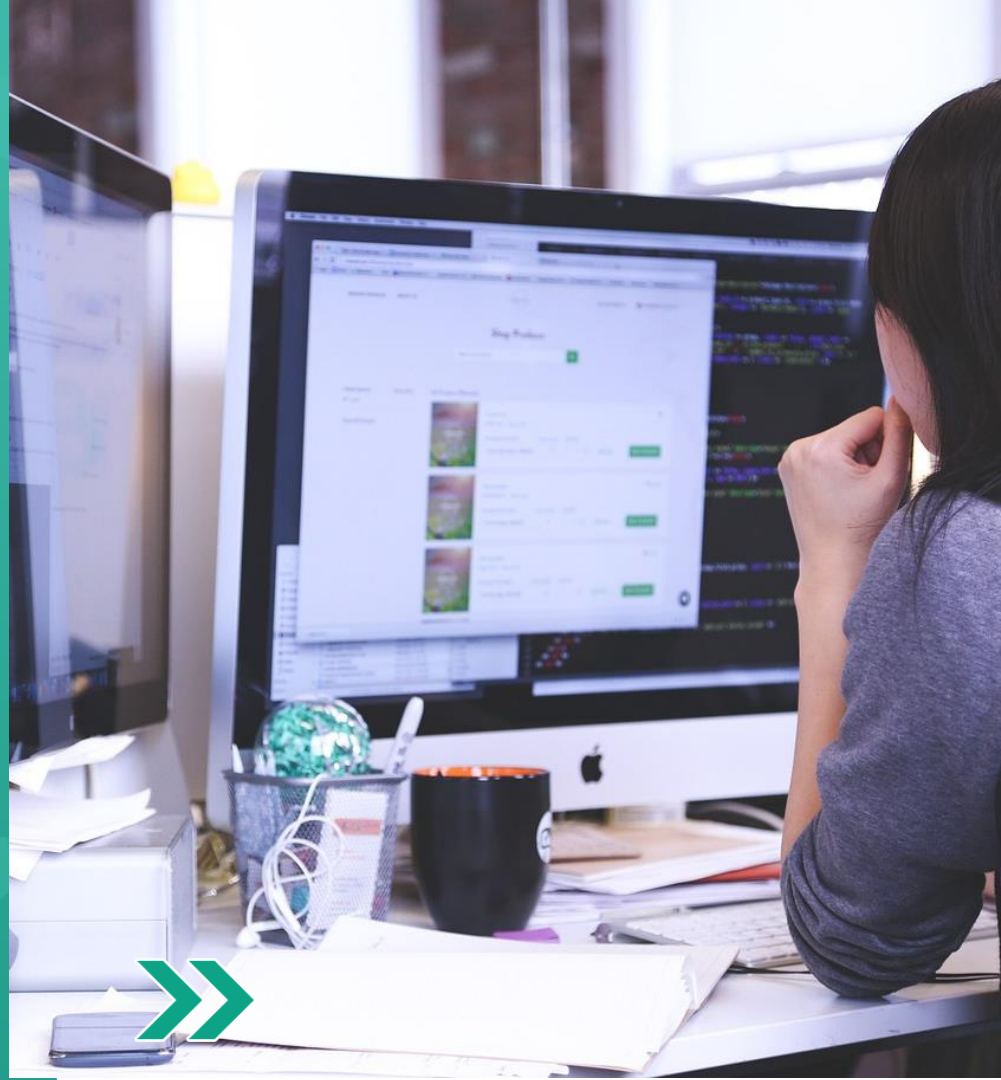


escola
britânica de
artes criativas
& tecnologia

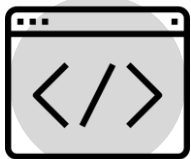
Profissão: Engenheiro Front-End



BOAS PRÁTICAS



Landing Page: página de um evento fictício



Confira boas práticas da comunidade de Front-End por assunto relacionado às aulas.

- **Inicie o desenvolvimento da Landing Page**
- **Insira a imagem de destaque**
- **Insira o botão de compra de ingresso**



Inicie o desenvolvimento da landing page

Acompanhe alguns motivos para usar a ferramenta Parcel em seu projeto.



- **Configuração zero:**
Parcel é conhecido por sua configuração zero. Isso significa que você pode começar a usá-lo imediatamente, sem precisar configurar manualmente um arquivo de configuração complexo. Parcel possui um conjunto de configurações padrão que funcionam bem para a maioria dos projetos, permitindo que você se concentre no desenvolvimento em vez de se preocupar com a configuração do ambiente.
- **Suporte para várias linguagens e frameworks:**
Parcel oferece suporte nativo para uma ampla variedade de linguagens e frameworks, como JavaScript, TypeScript, HTML, CSS, React, Vue.js e muitos outros. Independentemente da pilha tecnológica que você está utilizando, é provável que o Parcel possa lidar com ela de forma eficiente e simplificada.

Inicie o desenvolvimento da landing page

Acompanhe alguns motivos para usar a ferramenta Parcel em seu projeto.



- **Empacotamento rápido e eficiente:**
O Parcel é altamente elogiado por sua velocidade de empacotamento. Ele utiliza uma abordagem de empacotamento paralelo, que aproveita ao máximo os recursos do seu hardware, como múltiplos núcleos de processamento, para empacotar seu código de forma rápida e eficiente. Isso é especialmente benéfico durante o desenvolvimento, pois você obtém atualizações instantâneas e recarregamento automático do navegador (hot reloading) à medida que faz alterações no código.
- **Resolução automática de dependências:**
Parcel possui um sistema de resolução de dependências inteligente que analisa o seu código e identifica automaticamente as dependências necessárias. Ele lida com a importação de módulos e pacotes de forma transparente, permitindo que você se concentre no desenvolvimento em vez de gerenciar manualmente as dependências.

Inicie o desenvolvimento da landing page

Acompanhe alguns motivos para usar a ferramenta Parcel em seu projeto.



Suporte para otimização de produção:

Além de oferecer suporte para o desenvolvimento local, o Parcel também possui recursos avançados de otimização de produção. Ele pode criar pacotes otimizados e otimizar automaticamente o código para melhorar o desempenho e diminuir o tamanho dos arquivos finais. Isso é importante para garantir uma experiência de carregamento rápido para os usuários finais do seu aplicativo ou site.



Ecosistema de plugins:

Parcel possui um ecossistema de plugins ativo e expansível que permite estender suas funcionalidades. Você pode encontrar plugins para realizar tarefas específicas, como minificação de código, otimização de imagens, suporte a Sass, entre outros. Isso permite que você personalize o processo de empacotamento de acordo com as necessidades do seu projeto.



Inicie o desenvolvimento da landing page

Acompanhe alguns motivos para usar a ferramenta Parcel em seu projeto.



Comunidade ativa e suporte:

Parcel tem uma comunidade ativa de desenvolvedores e mantenedores. Isso significa que você pode encontrar suporte, documentação e exemplos disponíveis para ajudá-lo a tirar o máximo proveito da ferramenta. Além disso, Parcel é um projeto de código aberto, o que significa que você pode contribuir com melhorias e recursos, caso deseje.



Inicie o desenvolvimento da landing page

Acompanhe agora algumas dicas para uso da ferramenta.



- **Hot reloading:**
O Parcel habilita automaticamente o hot reloading por padrão. Isso significa que, sempre que você fizer alterações no seu código-fonte, o navegador será atualizado automaticamente para refletir as alterações. Não é necessário configurar nada adicionalmente para usufruir desse recurso.
- **Atenção aos erros de compilação:**
Se houver erros de compilação em seu código, o Parcel exibirá uma mensagem no terminal. As alterações não serão aplicadas até que os erros sejam corrigidos. Portanto, verifique sempre o terminal para garantir que não haja erros que impeçam o hot reloading de funcionar corretamente.
- **Recarregamento manual:**
Em alguns casos, o hot reloading automático pode não funcionar corretamente. Se você fizer uma alteração no código e não vir as atualizações no navegador, tente forçar um recarregamento manual da página.

Insira a imagem de destaque

Acompanhe as dicas para usar o gradiente linear no CSS.



- Escolha a direção correta:**
 A direção do gradiente linear pode ter um impacto significativo na aparência final. Pense sobre a disposição do elemento em relação ao gradiente e escolha a direção que melhor se ajusta à sua intenção. Experimente diferentes direções (por exemplo, de cima para baixo, da esquerda para a direita, diagonal) para encontrar a que mais lhe agrada.
- Utilize prefixos de navegador:**
 Para garantir a compatibilidade com diferentes navegadores, é recomendável usar prefixos de navegador ao definir o gradiente linear. Por exemplo, você pode usar `-webkit-linear-gradient` para navegadores baseados no WebKit (como o Chrome e o Safari), `-moz-linear-gradient` para o Firefox e assim por diante. Lembre-se de incluir a versão sem prefixo por último para fornecer suporte aos navegadores mais recentes.

Insira a imagem de destaque

Acompanhe as dicas para usar o gradiente linear no CSS.



- Experimente cores e paradas de cor:**
 O gradiente linear permite que você crie efeitos interessantes combinando diferentes cores. Experimente diferentes combinações de cores e adicione paradas de cor para criar transições suaves. Você pode usar valores hexadecimais, nomes de cores ou até mesmo a função `rgba()` para definir as cores.
- Use transparência:**
 Você pode adicionar transparência às cores do gradiente linear usando a função `rgba()` ou valores de transparência em `rgba` ou `hex`. Isso pode ser útil para criar efeitos de sobreposição ou transições sutis.
- Combine com outras propriedades CSS:**
 Além de `background-image`, você pode combinar gradientes lineares com outras propriedades CSS, como `background-color`, `border`, `text-fill-color`, `box-shadow`, entre outras. Isso permite a criação de designs mais complexos e a utilização de gradientes em diferentes elementos do seu layout.

Insira a imagem de destaque

Acompanhe as dicas para usar o gradiente linear no CSS.

- **Gradientes em texto:**
Você também pode aplicar gradientes lineares diretamente ao texto usando a propriedade `-webkit-background-clip: text;` combinada com um gradiente linear. Isso pode criar efeitos de texto colorido interessantes.
- **Explore gradientes múltiplos:**
O CSS permite a aplicação de múltiplos gradientes lineares a um elemento. Você pode usar a função `linear-gradient()` várias vezes, separando-as com vírgulas. Isso permite a criação de gradientes complexos com transições suaves entre eles.



Insira o botão de compra de ingresso

Acompanhe algumas dicas para usar a função `getTime()` de forma eficaz no JavaScript.



- Conversão para segundos ou outros formatos de tempo:**
 O valor retornado por `getTime()` é em milissegundos. Se você precisar do valor em segundos, divida o resultado por 1000. Você também pode converter para outros formatos de tempo, como minutos, horas ou dias, realizando cálculos adicionais com base em milissegundos.
- Comparação de datas:**
 Ao comparar duas datas usando `getTime()`, você pode obter dois timestamps e compará-los diretamente para determinar qual data é maior, menor ou se são iguais. Isso é útil ao classificar ou filtrar uma lista de datas.
- Cálculos de intervalo de tempo:**
 Você pode usar `getTime()` para calcular a diferença de tempo entre duas datas. Subtraia o timestamp de uma data pelo timestamp da outra para obter a diferença em milissegundos. Em seguida, você pode converter para outros formatos de tempo, se necessário.

Insira o botão de compra de ingresso

Acompanhe algumas dicas para usar a função `getTime()` de forma eficaz no JavaScript.



- Manipulação de datas:**
 Se você precisar adicionar ou subtrair um determinado período de tempo de uma data, pode fazer isso usando `getTime()`. Obtenha o timestamp da data original, execute os cálculos apropriados nos milissegundos correspondentes ao período de tempo e, em seguida, crie uma nova data a partir do novo timestamp usando `new Date()`.
- Trabalhando com APIs de tempo:**
 Muitas APIs de tempo e serviços externos fornecem valores de data e hora em formato de timestamp. Ao usar essas APIs, você pode aproveitar o `getTime()` para converter os timestamps recebidos em objetos `Date` do JavaScript.

Insira o botão de compra de ingresso

Acompanhe algumas dicas para usar a função `getTime()` de forma eficaz no JavaScript.



Testes de desempenho:

Em alguns casos, você pode querer medir o tempo decorrido entre duas partes do código para verificar o desempenho. Use `getTime()` para obter o timestamp antes e depois da seção do código que deseja testar e calcule a diferença para obter o tempo decorrido. Certifique-se de levar em consideração o fuso horário e a localização ao lidar com datas e timestamps em seu código.



Insira o botão de compra de ingresso

Acompanhe agora algumas dicas sobre o uso da função `setInterval` no JavaScript:

- Limite de uso:**
 Tenha cuidado ao usar o `setInterval` para evitar loops infinitos. Certifique-se de ter uma condição de parada apropriada dentro da função que está sendo executada. Caso contrário, o intervalo será executado continuamente, o que pode afetar o desempenho do seu aplicativo ou página web.
- Gerenciamento de memória:**
 Ao usar o `setInterval`, é importante ter em mente que a função passada como argumento é mantida na memória até que o intervalo seja cancelado usando `clearInterval`. Isso significa que se você atribuir o intervalo a uma variável e essa variável ficar fora de escopo ou não for mais necessária, você deve cancelar o intervalo para liberar a memória usando `clearInterval`.



Insira o botão de compra de ingresso

Acompanhe agora algumas dicas sobre o uso da função `setInterval` no JavaScript:



Ajuste de intervalo:

Escolha um intervalo adequado com base na necessidade da sua aplicação. Considere a frequência com que você precisa executar a função e o desempenho geral do seu aplicativo. Intervalos muito curtos podem sobrecarregar o navegador e resultar em uma experiência lenta para o usuário, enquanto intervalos muito longos podem fazer com que as atualizações pareçam lentas ou desatualizadas.



Lógica de sincronização:

Esteja ciente de que a função passada para `setInterval` será executada assincronamente, em intervalos regulares, mas não sincronizada com outras partes do seu código. Isso pode levar a problemas de sincronização se você estiver compartilhando dados ou executando operações simultâneas. Certifique-se de lidar com qualquer lógica de sincronização necessária adequadamente.

Insira o botão de compra de ingresso

Acompanhe agora algumas dicas sobre o uso da função `setInterval` no JavaScript:

- Utilização em animações:**
 O `setInterval` pode ser útil para criar animações simples, atualizando propriedades ou estilos em intervalos regulares. No entanto, para animações mais complexas e suaves, é recomendável considerar o uso de bibliotecas ou APIs dedicadas, como CSS Transitions ou a API de animação do JavaScript, que oferecem um melhor desempenho e controle mais refinado sobre a animação.
- Consideração de compatibilidade:**
 A função `setInterval` é amplamente suportada pelos navegadores modernos. No entanto, ao desenvolver para ambientes específicos ou navegadores mais antigos, verifique a compatibilidade com `setInterval` ou considere o uso de polyfills ou bibliotecas de terceiros que forneçam uma abstração para a criação de intervalos.



Insira o botão de compra de ingresso

Acompanhe agora algumas dicas sobre o uso da função `setInterval` no JavaScript:



Cancelando o intervalo:

Sempre lembre-se de chamar `clearInterval` passando o ID do intervalo retornado por `setInterval` quando você quiser parar a execução do intervalo. Isso é importante para evitar vazamentos de memória e garantir que o intervalo pare de ser executado quando não for mais necessário. Considere o contexto e as necessidades específicas do seu aplicativo ao decidir usar o `setInterval`.



Bons estudos!

