

Projetos JavaScript para Iniciantes

Este documento contém uma lista de projetos práticos para quem está começando a aprender JavaScript. Os projetos estão organizados do mais simples para o mais complexo, permitindo uma progressão natural no aprendizado.

☀️ Projetos Básicos (Conceitos Fundamentais)

1. Calculadora Simples

Conceitos: Variáveis, operadores, funções básicas

- Criar uma calculadora que realiza operações básicas (+, -, *, /)
- Interface simples com botões ou input do usuário
- Exibir resultado na tela

2. Conversor de Temperatura

Conceitos: Funções, condicionais, manipulação de DOM

- Converter entre Celsius, Fahrenheit e Kelvin
- Interface com campos de input e botões
- Validação de entrada do usuário

3. Gerador de Números Aleatórios

Conceitos: Math.random(), loops, arrays

- Gerar números aleatórios dentro de um intervalo
- Opção para gerar múltiplos números
- Histórico dos números gerados

4. Contador Simples

Conceitos: Eventos, manipulação de DOM, variáveis globais

- Botões para incrementar, decrementar e resetar
- Exibição do valor atual
- Limite mínimo e máximo opcional

5. Relógio Digital

Conceitos: Date(), setInterval(), formatação de strings

- Exibir hora atual em tempo real

- Formato 12h ou 24h
- Mostrar data completa

Projetos Intermediários Básicos

6. Lista de Tarefas (To-Do List)

Conceitos: Arrays, loops, eventos, localStorage (opcional)

- Adicionar, remover e marcar tarefas como concluídas
- Filtros (todas, pendentes, concluídas)
- Contador de tarefas

7. Jogo de Adivinhação

Conceitos: Loops, condicionais, validação, random

- Computador escolhe um número aleatório
- Usuário tenta adivinhar com dicas (maior/menor)
- Sistema de pontuação e tentativas

8. Validador de Formulário

Conceitos: Regex, eventos, validação, DOM

- Validar email, telefone, CPF
- Mensagens de erro personalizadas
- Feedback visual (cores, ícones)

9. Conversor de Moedas (valores fixos)

Conceitos: Objetos, funções, manipulação de dados

- Converter entre diferentes moedas
- Taxas de câmbio pré-definidas
- Interface intuitiva

10. Cronômetro

Conceitos: setInterval(), clearInterval(), formatação de tempo

- Iniciar, pausar, parar e resetar
- Exibir tempo em formato mm:ss:ms
- Histórico de tempos (opcional)

Projetos de Consolidação

11. Calculadora de IMC

Conceitos: Cálculos, validação, condicionais

- Calcular Índice de Massa Corporal
- Classificação do resultado (normal, sobrepeso, etc.)
- Dicas de saúde baseadas no resultado

12. Gerador de Senha

Conceitos: Strings, random, arrays, loops

- Opções de comprimento e caracteres
- Incluir/excluir números, símbolos, maiúsculas
- Indicador de força da senha

13. Quiz Interativo

Conceitos: Arrays de objetos, loops, condicionais, pontuação

- Perguntas com múltiplas alternativas
- Sistema de pontuação
- Resultado final com porcentagem

14. Calculadora de Idade

Conceitos: Date(), cálculos com datas, validação

- Calcular idade exata (anos, meses, dias)
- Próximo aniversário
- Validação de data de nascimento

15. Conversor de Unidades

Conceitos: Objetos, funções, múltiplas conversões

- Comprimento, peso, volume, área
- Interface com seleção de unidades
- Histórico de conversões

Projetos Lúdicos

16. Jogo da Velha (Tic-Tac-Toe)

Conceitos: Arrays bidimensionais, lógica de jogo, eventos

- Tabuleiro 3x3 interativo
- Detecção de vitória e empate
- Reiniciar jogo

17. Pedra, Papel, Tesoura

Conceitos: Random, condicionais, pontuação

- Jogar contra o computador
- Placar acumulativo
- Animações simples

18. Jogo de Memória Simples

Conceitos: Arrays, eventos, timeouts, lógica de jogo

- Cartas com números ou cores
- Virar cartas e encontrar pares
- Contador de tentativas

19. Simulador de Dados

Conceitos: Random, animações, arrays

- Simular lançamento de dados (1-6 faces)
- Múltiplos dados
- Estatísticas dos lançamentos

20. Gerador de Cores RGB

Conceitos: Math.random(), strings, DOM styling

- Gerar cores aleatórias
- Exibir código RGB e hexadecimal
- Aplicar cor como fundo da página

Dicas para Implementação

Estrutura Recomendada

```
projeto/  
├── index.html  
├── style.css  
├── script.js  
└── README.md
```

Boas Práticas

- Usar nomes descritivos para variáveis e funções
- Comentar o código para explicar a lógica
- Testar com diferentes entradas
- Implementar validação de dados
- Manter o código organizado e limpo

Progressão Sugerida

1. Comece com projetos que usam apenas `console.log()`
2. Evolua para manipulação básica do DOM
3. Adicione interatividade com eventos
4. Implemente validação e tratamento de erros
5. Explore armazenamento local (localStorage)

Recursos de Aprendizado

- **MDN Web Docs:** Documentação oficial do JavaScript
- **W3Schools:** Tutoriais e exemplos práticos
- **JavaScript.info:** Guia completo moderno
- **FreeCodeCamp:** Exercícios práticos gratuitos

Próximos Passos

Após dominar esses projetos básicos, você estará pronto para:

- Projetos com APIs externas
- Frameworks como React, Vue ou Angular
- Node.js para desenvolvimento backend
- Projetos mais complexos com bancos de dados

Dica Final: Não tenha pressa! Pratique cada projeto até se sentir confortável com os conceitos antes de avançar. A consistência é mais importante que a velocidade no aprendizado de programação.

