

Lista de Exercícios de Programação 02 - 2025.1

Professor: Henrique Pequeno

Monitor: Gustavo Nepomuceno Nogueira

Questões sobre P5.js

1. Crie uma animação interativa usando `p5.js`, onde uma "nuvem" de círculos se move pela tela. A posição e o movimento desses círculos devem ser repelidos com base na posição do mouse, de forma que os círculos se afastem do cursor conforme ele se move.
2. Crie uma animação interativa onde há três formas geométricas: **Triângulo**, **Quadrado** e **Círculo**. Quando o usuário clica em qualquer uma dessas formas, a cor dela deve mudar aleatoriamente.
3. Crie um jogo de corrida simples em que um número escalável de cavalos possa participar. Cada cavalo deve ter uma cor diferente gerada aleatoriamente e a sua velocidade deve mudar aleatoriamente a cada momento da corrida.
4. Crie um jogo onde um balão, gerado por uma função construtora, sobe da parte inferior da tela. O balão tem uma cor aleatória, uma velocidade aleatória e um valor de pontos aleatório, que é dado ao jogador ao ser clicado. Se o balão sair da tela sem ser clicado, o jogador perde 3 pontos. O jogo continua até o jogador atingir 50 pontos ou cair para -18 pontos, com a pontuação sendo atualizada conforme o balão é estourado ou perdido. Use apenas um balão por vez, sem listas, e altere a pontuação conforme os eventos de clique ou falha no clique.

5. Crie uma animação onde várias estrelas caem do topo da tela com tamanhos, posições e velocidades aleatórias. Use uma função construtora para gerar cada estrela com brilho e velocidade próprios. As estrelas devem se mover para baixo continuamente e, ao saírem da tela, voltarem ao topo com novas características.

Questões sobre Javascript Vanilla

6. Crie um código usando `for` que vá de 1 a 100. Para cada número, se for múltiplo de 2, mostre "Hello", se for múltiplo de 3, mostre "World"; se for múltiplo de ambos, mostre "HelloWorld"; e se não for múltiplo de nenhum, mostre "Nada". Evite repetir código e organize a lógica de forma eficiente.
7. Crie uma função `criarContador` que retorna um objeto com dois métodos: `incrementar()` e `zerar()`. O contador deve ser interno (não acessível de fora) e começar em 0. Cada chamada de `criarContador` deve gerar um contador independente. Use escopo para proteger a variável do contador.
8. Dado um array de produtos com nome e preço, crie uma função que filtre os produtos com preço acima de R\$ 100. Depois, converta os nomes desses produtos para maiúsculas e calcule o total dos preços. Use os métodos `filter`, `map` e `reduce`.
9. Crie uma função chamada `simularCaixaEletronico` que recebe dois parâmetros: `saldoInicial` e `operacoes`. A função deve aplicar as operações (saque ou depósito) ao saldo, uma por uma. Cada operação é um objeto com `{ tipo: 'saque' | 'deposito', valor: number }`. No final, exiba o saldo final. Implemente validação para impedir saque com saldo insuficiente.
10. Crie uma função chamada `validarSenha` que recebe uma string e verifica se ela atende aos seguintes critérios (caso prefira pode usar regex!):
 - Ter no mínimo 8 caracteres.

- Conter pelo menos uma letra maiúscula.
- Conter pelo menos uma letra minúscula.
- Conter pelo menos um número.
- Conter pelo menos um caractere especial (como `!@#$$%&*`).

A função deve retornar:

- `"Senha válida"` se a senha atender a todos os critérios.
- `"Senha inválida"` se não atender a algum critério.