**UNIVERSIDADE** **DO** **OESTE** **DE** **SANTA** **CATARINA**

**Campus** **de** **São** **Miguel** **do** **Oeste** **Curso:** **Ciência** **da** **Computação**

**Área:** **Ciências** **Exatas** **e** **Tecnológicas** **Disciplinas:** **Programação** **III**

**Professor:** **Roberson** **Junior** **Fernandes** **Alves**

**Carga** **horária:** **80** **h/a** **Semestre** **Letivo:** **2024/01**

LISTA DE EXERCÍCIOS DISPONIBILIZAR AS RESPOSTAS NO GITHUB

1. **(0,50)** Explique que vantagens e desvantagens existem no fato do JavaScript ser uma linguagem interpretada pelo navegador.

Vantagens:

Facilidade de uso: Como o JavaScript é interpretado pelo navegador do usuário, os desenvolvedores não precisam se preocupar em compilar o código antes de executá-lo. Isso simplifica o processo de desenvolvimento e permite uma iteração mais rápida.

Ampla compatibilidade: A maioria dos navegadores modernos suporta JavaScript, o que significa que os desenvolvedores podem criar aplicações web que funcionam em uma ampla variedade de dispositivos e plataformas.

Rápido ciclo de desenvolvimento: Como o código JavaScript é interpretado instantaneamente pelo navegador, os desenvolvedores podem ver imediatamente o resultado de suas alterações, tornando o processo de desenvolvimento mais eficiente.

Execução em tempo real: Como o JavaScript é executado no navegador do usuário, as interações com a página web podem ser processadas imediatamente, proporcionando uma experiência mais fluida e responsiva para o usuário final.

Desvantagens:

Desempenho limitado: A interpretação do JavaScript pode ser mais lenta do que a execução de código compilado, especialmente em casos de cálculos intensivos ou manipulações de grandes conjuntos de dados.

Segurança: Como o código JavaScript é executado no navegador do usuário, ele pode ser acessado e manipulado por terceiros maliciosos, representando potenciais vulnerabilidades de segurança, como ataques de injeção de código.

Dependência do navegador: Diferentes navegadores podem interpretar o JavaScript de maneira ligeiramente diferente, o que pode levar a inconsistências de comportamento entre plataformas, exigindo testes extensivos e trabalho adicional para garantir a compatibilidade cruzada.

Limitações de acesso ao sistema: JavaScript sendo executado no navegador tem acesso limitado ao sistema operacional do usuário, o que pode restringir certas funcionalidades avançadas que seriam possíveis em um ambiente de execução local.

**(1,00)** Associe os **eventos** com suas respectivas **funções**:

|  |  |
| --- | --- |
| **EVENTOS** | **DESCRIÇÃO** |
| 1) onclick | ( 8 ) mouse é movimentado sobre o elemento HTML. |
| 2) onkeypress | ( 4 ) elemento HTML recebe o foco. |
| 3) onblur | ( 3 ) elemento HTML perde o foco. |
| 4) onfocus | ( 1 ) elemento recebe um clique do mouse. |
| 5) onsubmit | ( 5 ) a página é carregada. |
| 6) onload | ( 6 ) elemento HTML tem seu valor alterado. |
| 7) onchange | ( 7 ) formulário é submetido. |
| 8) onmousemove | ( 2 ) uma tecla é pressionada e solta. |

1. **(1,00)** Crie um código em Javascript que receba os nomes de 5 pessoas através de campos de formulário(FORM HTML). Ao clicar em um botão Inverter, armazene-os em um array e apresente os valores nos mesmos campos do formulário, porém na ordem inversa da que foram inicialmente digitados.
2. **(1,00)** Crie uma função em JavaScript que receba 2 palavras através de campos de formulário e depois imprima-as intercaladas em um outro campo de formulário que deve está desabilitado (o usuário não deve conseguir digitar nenhum valor nesse campo). Ex.: ‘Joao’ e ‘Mary’; a impressão deve ser: ‘Jmoaaroy’.
3. **(1,50)** Implemente uma página HTML com um *babber* rotativo. Selecione 03 imagens,
4. uma para cada banner que deverão ser trocadas(exibidas) a cada 4 segundos. Use a função

setInterval do JavaScript para realizar a troca. Há várias formas de implementar essa troca de imagens. Uma sugestão é manter um array com os valores possíveis para a imagem e um inteiro que guarda qual é o banner atual:

**var** banners = ["img/banner1.png", "img/banner2.png", ...];

**var** bannerAtual = 0;

**function** trocaBanner() {

bannerAtual = (bannerAtual + 1) % 4; document.getElementById('imgBanner').src = banners[bannerAtual];

}

setInterval(trocaBanner, 4000);

Implemente ainda a possibilidade do usuário pausar a rotação das imagens no banner.

1. **(3,00)** Implemente uma página HTML que realize a validação e pesquisa de CEP, utilizando AJAX, nos webservices disponíveis em: <https://viacep.com.br/>. Permita que o usuário informa o CEP ou os dados da CIDADE a serem pesquisados. Os dados devem ser retornados no formato JSON e armazenados em um objeto JavaScript para então serem exibidos ao usuário. Permita ainda que o usuário faça a pesquisa de CEPs em um determinado intervalo, ou seja, ele pode digitar um CEP inicial e um CEP final.

**DATA** **PARA** **ENTREGA** **EM:** **14/03/0004.**