

## - Conhecimento geral

Explique com suas palavras:

### 1- Qual a diferença entre enviar dados via get ou post?

Pelo método GET, nos comunicamos com o servidor por variáveis enviadas diretamente pela URL requisitada no browser. Em um exemplo bem PERIGOSO temos <http://www.dominio.com/index.php?page=1&user=adriano&pass=123456>. Notem que dados importantes foram passados via GET e no caso de login e senha jamais ele deve ser utilizado pois um usuário um pouco mais avançado pode ter acesso à login, senha ou tentar burlar a navegação trocando o ID de uma página. Ou seja o método GET deve ser utilizado com cautela além disso conseguimos criar valores de variáveis que expiram ou não possuem uma sequência lógica tão clara. As vantagens do método GET é que conseguimos criar cache do conteúdo, a velocidade da requisição é mais rápida porém podemos mandar apenas ASCII por ela.

No método POST as variáveis de comunicação com o servidor são encapsuladas durante as trocas de pacotes com o servidor não ficando visível para o usuário, se bem que com um sniffer de rede é possível localizar e abrir estes pacotes na tentativa de encontrar alguma informação importante. Para evitar a visibilidade da informação neste caso é possível encriptar estas variáveis dificultando ainda mais o acesso à alguma informação sigilosa. Além disso pelo método POST as variáveis não precisam ser em ASCII e é possível enviar uma quantidade muito maior de dados que pelo método GET.

### 2- O que é uma constante?

Constante é uma variável que quando declarada recebe um valor de qualquer tipo, somente de leitura, não sendo possível modificá-lo durante a execução do aplicativo. Na matemática existem números constantes famosos e, para facilitar a programação e a execução do código, são implementados como propriedades de objetos globais exemplo o famoso Math.PI do Javascript. Existem também variáveis que podem ser acessadas pelas linguagens de programação que possuem valores somente de leitura ou seja constantes um exemplo é ao acessar a versão do sistema operacional onde a aplicação está sendo executada pelo OperatingSystem OSVersion do C#.

### 3- Descreva três tipos de variáveis.

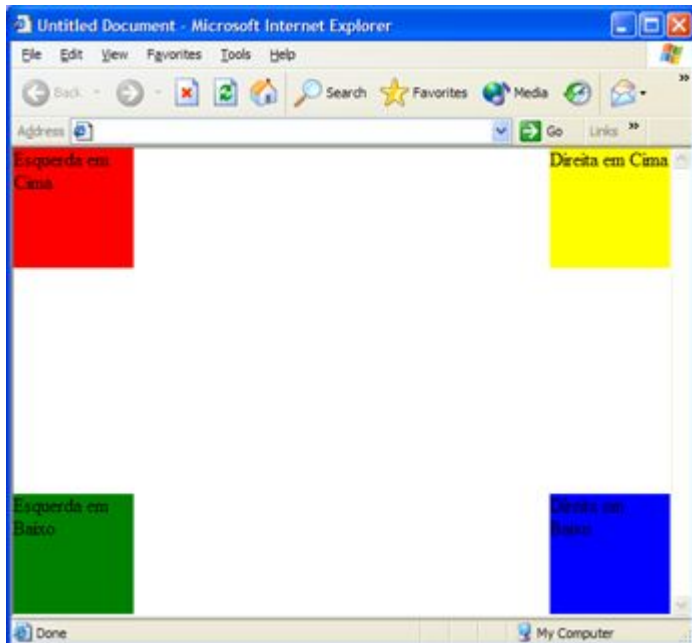
Usando o javascript como exemplo, onde não nos preocuparemos com alocação de memória ou troca de tipagem de variáveis, podemos ter dentre outros tipos variáveis, as variáveis booleanas (Boolean), que aceitam um valor binário de 0 e 1 ou *true* e *false* e representa um teste de verdadeiro ou falso (*true* e *false*). Existem variáveis *string* que aceitam uma cadeia de caracteres alfa-numéricos, exemplo o famoso termo "Olá mundo!". Existem variáveis numéricas onde apenas números podem ser atribuídos.

### 4- Qual a diferença entre "while" e "do while"?

No laço de iteração "while" e "do while" um bloco de programação é executado enquanto é verdadeiro um teste lógico atribuído à ele. A diferença entre os dois é que no "do while" o bloco de programação é executado e, após isto, repetido enquanto o teste lógico for verdadeiro, ou seja, executado ao menos uma vez. No "while" o bloco é executado somente se for verdadeira a condição atribuída.

## - Conhecimento CSS

5- Criar um css que tenha esse efeito



```
<body>
  <div class="ec">
    <p>Esquerda em Cima</p>
  </div>
  <div class="eb">
    <p>Esquerda em Baixo</p>
  </div>
  <div class="dc">
    <p>Direita em Cima</p>
  </div>
  <div class="db">
    <p>Direita em Baixo</p>
  </div>
</body>
```

Ver resposta em [https://github.com/caversan/teste\\_encrypta/blob/master/teste\\_frontend/teste\\_css.html](https://github.com/caversan/teste_encrypta/blob/master/teste_frontend/teste_css.html)

6- Faça um css que deixe o primeiro item azul e o restante vermelho

Ver resposta em

[https://github.com/caversan/teste\\_encrypta/blob/master/teste\\_frontend/teste\\_css.html](https://github.com/caversan/teste_encrypta/blob/master/teste_frontend/teste_css.html)

## - Conhecimento jquery

7- Usando JQuery, selecione todos os elementos <DIV> de uma página que contenham a classe CSS “okButton”

Veja a resposta em

[https://github.com/caversan/teste\\_encrypta/blob/master/teste\\_frontend/teste\\_jquery.html](https://github.com/caversan/teste_encrypta/blob/master/teste_frontend/teste_jquery.html)

8- Usando JQuery, adicione e remova a classe CSS de um elemento

Veja a resposta em

[https://github.com/caversan/teste\\_encrypta/blob/master/teste\\_frontend/teste\\_jquery.html](https://github.com/caversan/teste_encrypta/blob/master/teste_frontend/teste_jquery.html)

## - Conhecimento específico Javascript e lógica

9- Qual o resultado do loop abaixo?

```
for(x=0; x<=5; x++){  
    for(y=0; y<2; y++){  
        console.log(x + " : " + y);  
        x--;  
    }  
}
```

O resultado é uma recursão infinita do primeiro laço “for” pois estamos decrementando “x” no laço for(y=0 ; y<2; y++). A página irá congelar até deixar de responder. por consequência toda a aplicação que estiver rodando na página parará. Em computadores com poucos o browser e o próprio sistema operacional poderão deixar de responder.

10- Qual o resultado do código abaixo.

```
var text1 = "Hello";  
(function() {  
    var text2 = " World";  
    alert(text1 + text2);  
})();  
alert(text1 + text2);
```

Aparecerá apenas uma janela de alerta escrita “Hello World” do “alert” que está contido dentro da função anônima pois a variável “text2” está declarado apenas no escopo dela. A segunda janela de alerta não abrirá pois a variável text2 não existe no escopo global. Abaixo a função anônima resumida exibindo o mesmo resultado.

```
(function(text1, text2) {  
    alert(text1 + text2);  
}) ("Hello", " World");
```

11- Fazendo um teste de mesa para saber o resultado dessa função, consegue saber qual o propósito da rotina a seguir?

**O propósito é converter um número decimal declarado na variável “resultado” em um número hexadecimal**

(obs: código ascii 65 corresponde ao carácter A)

- A) Transformar número na base 10 para hexadecimal.
- B) Criar um hash de uma sequência de números.
- C) Nenhuma das alternativas anteriores (explique).

```
function converte(inteiro){  
    if(inteiro>=10){  
        return String.fromCharCode(65+(inteiro-10));  
    }else{  
        return inteiro;  
    }  
}  
  
function rotina(valor){  
    string = "";  
    do{  
        resto = converte(valor%16);  
        string = resto + string;  
        valor = Math.floor(valor/16);  
    }while(valor>0);  
  
    return string;  
}  
  
resultado = rotina([número qualquer]);
```

12- Faça uma função que recebe um número qualquer, adiciona esse número a um array, e ordena esse array em ordem crescente

```
numeros = [1, 8, 4, 5, 7, 4, 3, 2]  
insereNumeros = function(numero) {  
    numeros.push(numero);  
    numeros = numeros.sort(function(a, b) {  
        return a - b  
    });  
}  
  
insereNumeros(20)
```

13- Faça uma função, passando o horário de entrada e o horário de saída, que calcule o custo total do período em que seu carro ficou no estacionamento.

As regras do estacionamento são:

- A taxa de entrada custa 2
- A primeira hora (completa ou não) custa 3
- Cada hora adicional (completa ou não) custa 4

Os horários de entrada e saída são strings no formato "HH:MM" (horas de 00 a 23, e minutos de 00 a 59)

Pode assumir que o horário de entrada e saída sempre acontece no mesmo dia

```
calculaEstacionamento = function(horaEntrada, horaSaida) {  
    var entrada = new Date();  
    var saida = new Date();  
    var total = new Date();  
    var vlrEntrada = 2;  
    var vlrPrimeiraHora = 3;  
    var vlrHoraAdicional = 4;  
    var vlrTotal = 0;  
  
    entrada.setHours(Number(horaEntrada.split(":")[0]),  
Number(horaEntrada.split(":")[1]));  
    saida.setHours(Number(horaSaida.split(":")[0]),  
Number(horaSaida.split(":")[1]));  
    total = saida - entrada;  
  
    if (total > 0) {  
        vlrTotal = vlrTotal + vlrEntrada + vlrPrimeiraHora;  
    }  
    if (total > 3600000) {  
        vlrTotal = vlrTotal + (vlrHoraAdicional * Math.ceil((total -  
3600000) / 3600000))  
    }  
  
    console.log(vlrTotal)  
  
}  
calculaEstacionamento("10:30", "12:31")
```

14- Considerando as listas abaixo, faça um script que permita o usuário navegar pelos **boxes** utilizando o teclado, Por exemplo: Seleciona o elemento a direita quando pressionar, no teclado, a seta para direita, seleciona o elemento abaixo quando pressionar a seta para baixo... assim por diante

Veja a resposta em [https://github.com/caversan/teste\\_encrypta/blob/master/teste\\_frontend/teste\\_box\\_li.html](https://github.com/caversan/teste_encrypta/blob/master/teste_frontend/teste_box_li.html)

```
<style>
.box{
    width: 50px;
    height: 100px;
    background-color: #f00;
    display: inline-block;
}
.selected{
    border: 3px solid #00f;
}
</style>

<body>
    <div>Lista 1</div>
    <ul data-listindex="0">
        <li class="box selected"></li>
        <li class="box"></li>
        <li class="box"></li>
        <li class="box"></li>
    </ul>

    <div>Lista 2</div>
    <ul data-listindex="1">
        <li class="box"></li>
        <li class="box"></li>
        <li class="box"></li>
        <li class="box"></li>
    </ul>
</body>
```