

Exemplos e Trabalho Sobre Métodos e Laços

- 1) Criar um arquivo chamado Util.js
 - a. Criar a classe Util dentro do arquivo criado anteriormente.
 - b. Criar métodos estáticos para reduzirmos os seguintes comandos:
 - c. `prompt("Digite seu valor")`
 - d. `parseInt(prompt("Digite seu valor"))`
 - e. `parseFloat(prompt("Digite seu valor"))`
 - f. `alert("Mensagem qualquer")`
 - g. `document.getElementById("id_elemento")`
 - h. `document.getElementById("id_elemento").value`
 - i. `parseInt(document.getElementById("id_elemento").value)`
 - j. `parseFloat(document.getElementById("id_elemento").value)`
 - k. `document.getElementById("id_elemento").value = valor`
 - l. `document.getElementById("id_elemento").innerHTML = valor`
 - m. `document.getElementById("id_elemento").innerHTML += valor`
- 2) Um algoritmo que pergunte o nome, idade e altura. Ao final, informar se esta pessoa é maior de idade ou não e informar de sua altura está entre o 1.77 e 1.68.

Resolução:

HTML

```
6 <body>
7 <p>
8 <label>
9     Seu nome:
10    <input type="text" id="nome" >
11 </label>
12 </p>
13 <p>
14 <label>
15     Sua idade:
16    <input type="text" id="idade" >
17 </label>
18 </p>
19 <p>
20 <label>
21     Seu peso
22    <input type="text" id="peso" >
23 </label>
24 </p>
25 <p>
26     <button onclick="verificar()">Verificar</button>
27 </p>
28 <hr>
29 <div id="res"></div>
30 </body>
31 </html>
```

JAVASCRIPT SEM UTIL:

```
4 <script>
5     function verificar(){
6         var nome = document.getElementById("nome").value;
7         var idade = parseInt(document.getElementById("idade").value);
8         var altura = parseFloat(document.getElementById("altura").value);
9
10
11         if(nome==" " || isNaN(idade)==true || isNaN(altura)==true){
12             alert("Preencha corretamente os campos!");
13             return; //Saída forçada do método verificar
14         }
15
16         if(idade>=18){
17             document.getElementById("res").innerHTML = nome + " você é maior de idade.";
18         }else{
19             document.getElementById("res").innerHTML = nome + " você é menor de idade.";
20         }
21
22         if(altura>=1.68 && altura<=1.77){
23             document.getElementById("res").innerHTML += "<br>Você está na altura ideal.";
24         }else{
25             document.getElementById("res").innerHTML += "<br>Você está acima da altura";
26         }
27     }
28 </script>
```

JAVASCRIPT COM UTIL:

```
4 <script>
5     function verificar(){
6         var nome = Util.getValue("nome");
7         var idade = Util.getInt("idade");
8         var altura = Util.getFloat("altura");
9
10
11         if(Util.ehVazio(nome) || !Util.ehNumero(idade) || !Util.ehNumero(altura)){
12             Util.show("Preencha corretamente os campos!");
13             return; //Saída forçada do método verificar
14         }
15
16         if(idade>=18){
17             Util.html("res", nome + " você é maior de idade.");
18         }else{
19             Util.html("res", nome + " você é menor de idade.");
20         }
21
22         if(Util.estaIntervalo(altura, 1.68, 1.77)==true){
23             Util.addHTML("res", "<br>Você está na altura ideal.");
24         }else{
25             Util.addHTML("res", "<br>Você está acima da altura");
26         }
27     }
28 </script>
```

- 3) Criar um algoritmo em que o usuário possa informar o nome, valor, quantidade e percentual de desconto da compra de um produto. Ao final, mostrar o total a ser pago e quanto de desconto foi concedido em reais. Utilizar a classe Util para entrada e saída de dados. (Resolver com inputs).
- 4) Criar uma classe chamada Valida com métodos estáticos para validação conforme lista abaixo:
 - a. Verificar se um valor é negativo;
 - b. Verificar se um valor é positivo;
 - c. Verificar se um valor é neutro;

- d. Verificar se é par;
- e. Verificar se é ímpar;
- f. Verificar se um valor é numérico;
- g. Verificar se um valor é vazio;
- h. Verificar se é um número inteiro;
- i. Verificar se é um número real;
- j. Verificar se determinada variável do tipo caractere possui um determinado tamanho;
- k. Verificar se um número está em um dado intervalo numérico;

5) Refazer o exercício 3 perguntando em quantas vezes deseja parcelar o pagamento. Mostrar uma saída como o seguinte exemplo:

- Exemplo para compras no dia 21/11/2017:

- Total = 1000.00
- Percentual desconto = 10%
- Parcelas = 5
- totalPagar = 1100.00
- valorParc = 220.00
- Saída:

```
Parcela 1 - 21/11/2017 - R$ 220.00
Parcela 2 - 21/12/2017 - R$ 220.00
Parcela 3 - 21/01/2018 - R$ 220.00
Parcela 4 - 21/02/2018 - R$ 220.00
Parcela 5 - 21/03/2018 - R$ 220.00
```

○

Utilizar a classe Util e a classe Verifica para as seguintes validações:

- Nome não pode ser vazio e deve haver pelo menos 3 caracteres;
- Valor não pode ser negativo, não pode ser uma cadeia de caracteres do alfabeto e não pode estar vazio;
- Quantidade deve ser um número do tipo inteiro, não pode estar vazio, deve ser no mínimo 1 e no máximo 12;
- Percentual de desconto deve estar entre 0 e 100, não pode estar vazio e deve ser um número.

- 6) Criar uma classe chamada Lacos com alguns métodos estáticos:
- Retornar uma string com todos os números de 0 até n , onde n é definido pelo usuário.
 - Retornar o somatório dos números de 0 até n , onde n é definido pelo usuário.
 - Retornar a média dos números de m até n , onde m e n são definidos pelo usuário e são respectivamente o início e fim do intervalo numérico;
 - Retornar a contagem dos números pares e múltiplos de 6 ao mesmo tempo. O intervalo é definido pelo usuário e passado como parâmetro para dentro da função.
- 7) Criar uma interface para que o usuário possa interagir com o sistema e poder escolher utilizar um dos recursos disponibilizados pela classe Lacos. Utilizar a classe Util para entrada e saída de dados e a classe valida para validação da entrada dos dados.
- 8) Faça um programa em que o usuário possa informar um mês no formato numérico e imprimir o mês num formato de calendário. Exemplo de saída para a entrada de valor 11.

D	S	T	Q	Q	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		