Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
Prof. Felippe Scheidt – IFPR – Campus Foz do Iguaçu

Desenvolvimento Web I

JS | Math, Strings, Arrays e Date

Funções matemáticas

- Javascript possui uma biblioteca interna chamada Math que pode auxiliar bastante na realização de cálculos:

`Math.abs(x)`	Retorna o valor absoluto
`Math.cell(x)`	Arredonda o número para cima
`Math.cos(x)`	Retorna o cosseno de x
`Math.floor(x)`	Arredonda o número para baixo
`Math.max(x,y)`	Retorna o maior valor
`Math.min(x,y)`	Retorna o menor valor

Funções matemáticas

`Math.pow(x,y)`	Retorna x elevado ao expoente y
`Math.random()`	Retorna um número aleatório entre 0 e 1
`Math.sqrt(x)`	Retorna a raiz quadrada
`Math.sin(x)`	Retorna o seno de x
`Math.PI`	Retorna o número π

Para lista completa dos métodos do objeto Math consulte o site:
http://www.w3schools.com/js/js_math.asp

Trabalhando com Datas

O Javascript oferece o objeto Date() que permite manipular datas. Exemplos:

```
var hoje = new Date(); // retorna a data atual
hoje.getFullYear(); // retorna o ano
hoje.getMonth(); // retorna o mês (int)
hoje.getDate(); // retorna o dia do mês (int)
hoje.getDay(); // retorna o dia da semana (int)
hoje.getHours(); // retorna a hora
hoje.getMinutes(); // retorna o minuto
hoje.getSeconds(); // retorna o segundo
```

Trabalhando com datas

- Obter a data no padrão brasileiro:

```
Date data = new Date();
data.toLocaleDateString();
```

- Criar uma Data:

```
data = new Date(ano,mes,dia,hora,minutos,segundos);
ex:
data = new Date(2013,11,20);
```

Lembre-se que no javascript, janeiro é 0 e dezembro 11.

Trabalhando com datas

- Calculando a diferença entre datas.

```
Date data1 = new Date();
Date data2 = new Date();
var diff = (data1 - data2);
// porém o retorno é em milisegundos
// é necessário dividir pela quantidade de milissegundos em
// um dia para encontrar a quantidade de dias.
var milisecPorDia = 24 * 60 * 60 * 1000;
```

Trabalhando com Strings

- Variáveis do tipo String possuem diversos métodos para manipular caracteres. Por exemplo, seja a seguinte declaração:

```
var str = "Olá, mundo";
```

- Podemos executar os seguintes comandos nesta variável:

`str.length`	retorna o tamanho da string
`str.replace('m','M')`	substitui um conteúdo por outro
`str.toLowerCase()`	converte todos caracteres para minúsculo
`str.toUpperCase()`	converte todos caracteres para maiúsculo
`str.substring(0,3)`	remove uma parte da string
`str.split(',')`	quebra a string de acordo com um padrão

Vetores

- Vetores permitem armazenar mais de um valor. No javascript são dinâmicos, isto é, para utilizá-los não precisamos especificar seu tamanho. Além disso, ele pode ter seu tamanho modificado durante a execução do script.

```
/* exemplo de declaração de uma variável
do tipo vetor com tamanho zero */
var vetor = [];
```

Declaração vetores

- Exemplo de declaração de vetores:

```
/* declaração de um vetor de strings com
tamanho 4 */
var conceitos_web = ["A","B","A","C"];

/* acessando a posição de cada elemento: */
alert(conceitos_web[0]); // A
alert(conceitos_web[3]); // C
```

Percorrendo vetores

- Para percorrer um vetor, basta usar o atributo length e obter o tamanho do mesmo. Usar um for para imprimir cada posição:

```
var notas = [6,8,9,2,4,10,7,8,9,4];
for(i=0;i<notas.length;i++){
    // apertar F12 para visualizar
    console.log(notas[i]);
}
```

Métodos

- Todo array é na verdade um objeto. Objetos do tipo array possuem métodos para auxiliar na manipulação dos elementos do vetor:

```
var vetor = ["pera","banana","uva","abacate"];
// adiciona a fruta amora
vetor.push("amora");
// remove o último elemento (amora)
vetor.pop();
// remove o primeiro elemento (pera)
vetor.shift();
// adiciona laranja logo no início do vetor
vetor.unshift("laranja");
// ordena o vetor em ordem alfabética
vetor.sort();
```

Remover um determinado item

- Para remover um elemento em determinada posição do vetor, podemos usar a combinação da função **indexOf** para obter a posição do elemento dentro do vetor e a função **splice** que remove o elemento.

```
var alunos = ["julia","fernando","diego"];
// obter a posicao do elemento no vetor:
var posicao = alunos.indexOf("fernando");
// usar a função splice para remover o elemento
alunos.splice(posicao,1);
```

- O segundo parâmetro da função splice refere-se a quantidade de elementos que serão removidos.

Função setInterval()

- Função do javascript que permite executar um código ou função repetidamente.
- Vamos criar uma função contadora, que incrementa i e atualiza o conteúdo na div cont:

```
<script>
var i = 0;
function contador(){
  i++;
  document.getElementById("cont").innerHTML = i;
}
</script>
```

Função setInterval()

- Para chamar a função contador repetidamente usamos o setInterva() passando como parâmetro o nome da função que será executada e o intervalo de tempo em milissegundos que a função será chamada:

```
<body onload="setInterval(contador,1000);">
    <div id="cont">0</div>
</body>
```

Cancelando o setInterval

- Para isso, é necessário criar uma referência para a função setInterval(), no momento em que ela foi definida:

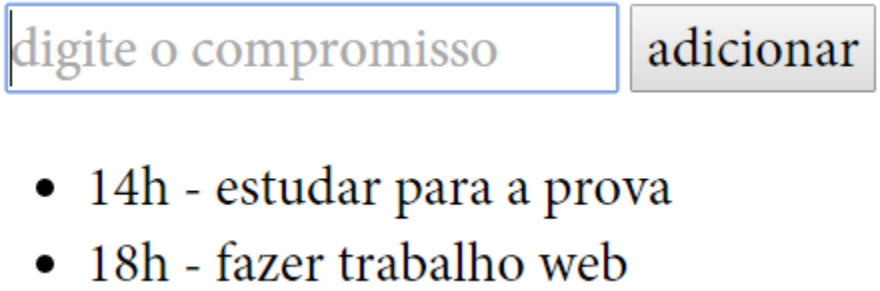
```
var interval = setInterval(someFunction, 1000);
```

- Com a variável interval, podemos usar a função clearInterval para cancelar a repetição:

```
clearInterval(interval);
```

Atividades

1) Faça um script que lê 4 notas digitadas em 4 inputs e armazena esses valores num vetor. Depois mostre a médias das notas percorrendo o vetor quando o usuário clica num botão exibir média.
2) Faça um script que sorteia quem irá fazer as compras do supermercado. Coloque 5 nomes num vetor e sorteie um número de 1 a 5 e mostre quem foi o escolhido.
3) Implemente um relógio no javascript que utiliza o setInterval para atualizar as horas, minutos e segundos na tela.
4) Faça uma agenda, que permite adicionar compromisso colocando-os numa lista



Atividades

5) Faça um script que gera 1000 números inteiros aleatoriamente dentro do intervalo de `0 a 100` e salva estes números num vetor. Verifique nesse array quantos números são ímpares e quantos são pares. Mostre o resultado na tela. Exemplo:

Pares: 493

Ímpares: 507