Instituto Federal de Educação Ciência e Tecnologia
Curso Técnico Integrado em Informática
Campus Currais Novos

Desenvolvimento Web

JavaScript – aula III –

Operadores, estruturas de decisão e de repetição

Professor: Bruno E. G. Gomes

2013

Introdução

- Na aula de hoje:
  - Como escrever expressões em *JavaScript*?
    - ***Operadores***
  - E se eu quiser desviar o fluxo de execução do script?
    - ***Estruturas de decisão e repetição***
  - Exercícios

Operadores

- Operadores são utilizados em expressões para se gerar algum resultado
- Operadores *JavaScript* são semelhantes aos operadores de C
- Tipos de Operadores
  - Aritméticos
  - Atribuição
  - Comparação
  - Lógicos

Operadores Aritméticos

- Operandos são expressões inteiras ou reais
- Resultado inteiro ou real, dependendo do tipo dos operandos

Operador	Descrição	Exemplo	Resultado
+	Adição	x = y + 2	
-	Subtração	x = x – 2	
*	Multiplicação	x = 3 * 2	
/	Divisão	x = y / 2	
%	Módulo (resto inteiro da divisão)	x = y % 2	
++	Incremento	x++	
--	Decremento	x = --y	

Operadores de Atribuição

Operador	Exemplo	Equivalente a
=	x = 1	
+=	x += 2	x = x + 2
– =	x –= y	x = x – y
*=	x *= (a + b)	x = x * (a + b)
/=	x /= (x + 1)	x = x / (x + 1)
%=	x %= 2	x = x % 2

Operadores de comparação (relacionais)

- Comparação de valores entre dois operandos (variáveis, constantes ou expressões)
- Retornam verdadeiro (*true*) ou falso (*false*)

Operador	Descrição	Exemplos (p/ x == 5)
==	Igual a	x == 8 (false)
===	Estritamente igual a (valor e tipo)	x === 5 (true) x === "5" (false)
!=	Diferente de	x != 10 (true)
>	Maior que	x > 8 (false)
<	Menor que	x < 8 (true)
>=	Maior ou igual a	x >= 2 (true)
<=	Menor ou igual a	x <= 5 (true)

Operadores Lógicos

- Operações fundamentais da lógica proposicional
  - OU, E, NÃO
- Resultado é o valor verdade verdadeiro (*true*) ou falso (*false*)

Operador	Descrição	Exemplo (p/ x = 6 e y = 3)
&&	E	(x < 10 && y > 1) é *true*
||	Ou	(x == 5) || (y == 5) é *false*
!	Não	!(x == 10) é *true*

Obtendo informações do ambiente do usuário

- Muitas vezes é útil sabermos detalhes do ambiente de navegação do usuário
  - Navegador, SO, etc.

- *Exemplo*: navegador pode exibir HTML/CSS um pouco diferente de outro
  - Correções tem que ser feitas para que a página seja exibida igualmente nos dois ambientes

- Utilizamos o objeto ***Navigator*** para ter acesso a informações sobre o ambiente de execução do usuário

Funções do objeto "Navigator"

```
document.write( "<p><strong>Código do navegador</strong>:
                 " + navigator.appCodeName + "</p>");

document.write("<p><strong>Nome do navegador: </strong>"
                + navigator.appName + "</p>");

document.write("<p><strong>Versão do navegador: </strong>"
                + navigator.appVersion + "</p>");

document.write("<p><strong>Cookies habilitados? </strong>"
                + navigator.cookieEnabled + "</p>");

document.write("<p><strong>Platforma: </strong>" +
                navigator.platform + "</p>");
document.write("<p><strong>Cabeçalho User-agent: </strong>"
                + navigator.userAgent + "</p>");
```

Estruturas de Decisão

- Sintaxe e semântica semelhante a C

- Estruturas
  - If
  - Switch

Estrutura IF

```
if (condition) {
    <código executado se a condição for verdadeira>
} else {
    <código executado se condição for falsa>
}
```

- <Condição> é uma expressão que resulta em um valor booleano (*true* ou *false*)
  - Avaliação para verdadeiro (**true**) faz com que o código entre a chave de abertura do corpo do **if** e a chave de fechamento correspondente seja executado.
  - Parte **else** é opcional. Deve ser utilizada quando se quer especificar o que deve ser feito quando a avaliação da condição resultar em falso (**false**).

Exemplo

- **If** sem a parte **else**:

```
var data = new Date();
var time = data.getHours();

if (time < 12) {
        document.write("<p><strong>Bom dia</strong></p>");
}
```

Exemplo – if-else aninhado

```
var d = new Date();
var time = d.getHours();

document.write(
    "<p> Hoje é dia: " +
    d.getDate()+ "/" + d.getMonth() + "/" + d.getYear() +
    "</p>");

if (time < 12) {
    document.write("<b>Bom dia!</b>");
} else if (time >= 12 && time < 18) {
    document.write("<b>Boa tarde!</b>");
} else {
    document.write("<b>Boa noite!</b>");
}
```

Estrutura Switch

- O switch é um caso especial de um IF aninhado.

Sintaxe:

```
switch(val) {
case 1:
  <instruções>
  break;
case 2:
  <instruções>
  break;
case n:
   <instruções>
default:
  <instruções executadas se val for diferente de todos  os casos anteriores>
}
```

Semântica:

1. **val** é uma expressão que é avaliada uma vez na entrada do *switch*;
2. O valor de **val** é comparado com cada caso (**case**) na sequência;
3. Se algum **case** casar exatamente com o valor de **val**, as instruções correspondentes são executadas
4. Caso não aja nenhum casamento, as instruções em **default** são executadas.

OBS-1.: *breaks* são colocados para evitar que casos posteriores ao caso selecionado sejam avaliados.

OBS-2: tipos de dados permitidos para **val** : números, caracteres ou strings.

Exemplo de script com Switch

```
var data = new Date();
var dia = data.getDay();

switch (dia)
{
case 0:
    document.write("Domingo: descansar,  pois ninguém é de ferro!");
    break;
case 1:
    document.write("Começando a semana de trabalho...");
    break;
case 5:
    document.write("Finalmente sexta, aula de Web e começo do fim de
    semana.");
    break;
case 6:
    document.write("Sábado, é hora de diversão!");
    break;
default:
    document.write("Só trabalho e estudo!");
}
```

Estruturas de Repetição

- Permitem repetir trechos de código em um programa
- Repetição depende da avaliação de uma certa condição (valor booleano)
- Estruturas de repetição em *JavaScript*
  - *while*
  - *do – while*
  - *for*

Estrutura de repetição "*while*"

```
while (<condição_de_teste>)
{
        <código a ser executado>
}
```

- Repete o trecho de código em seu bloco (entre **{** e **}**) *enquanto* a *condição de teste* for verdadeira (*true*)
- Condição de teste é qualquer expressão booleana

Exemplo

```
// Multiplica números de um vetor por 2
var numeros = [1, 6, 7, 9, 15];
var inc = 0;

while (inc < numeros.length) {
        numeros[inc] *= 2;
        document.writeln(
            "<p>numeros[" + inc +"] = " + numeros[inc] + "</p>"
        );
        inc++;
}
```

Estrutura de repetição "do-*while*"

```
do
{
        <código a ser executado>
} while (<condição_de_teste>)
```

- Semelhante à estrutura *while*, com a diferença que o teste é feito no final;
- Código é repetido ao menos 1 vez antes do teste ser feito

Exemplo

```
var numeros = [1, 2, 3, 4, 5];
var inc = 0;

do {
    numeros[inc] *= 2;
    document.writeln(
            "<p>numeros[" + inc +"] = " + numeros[inc] + "</p>“
     );
     inc++;
} while (inc < numeros.length);
```

Estrutura de repetição "*for*"

```
for (var = valorIni; teste; incremento)
{
    <código a ser executado>
}
```

- cabeçalho do *for* dividido em três partes:

1. **Inicialização** – variável de controle (*var*) recebe um valor inicial;
2. **Teste** – teste para a continuação da repetição, feito no início e após cada incremento. Continua enquanto verdadeiro;
3. **Incremento** – atualização do valor de *var* após cada repetição.

Exemplo

```
var sum = 0;

for ( var number = 2; number <= 100; number += 2) {
    sum += number;
}

document.writeln(
    "A soma dos inteiros pares de 2 a 100 é: " + sum);
```