

# COM222

## DESENVOLVIMENTO DE SISTEMAS WEB

Aula 02: JavaScript – Parte I

- Scripts vs. programas
- Introdução a JavaScript
  - Tipos de dados e expressões
  - Controle de execução
  - Funções e bibliotecas
  - Strings e Arrays
  - Objetos date, document e navigator
  - Classes de usuário

# Scripts vs Programas

3

- Uma linguagem de script é uma linguagem interpretada, em geral, mais simples e menos robusta que linguagens compiladas
- **Vantagens** das linguagens de scripts
  - ▣ Possuem um modelo de execução simples
    - Não necessitam de compilador ou IDE
  - ▣ Economizam banda, pois o código fonte é baixado e não um executável pré-compilado
  - ▣ São independentes de plataforma
    - Interpretados pelo browser

# Scripts vs Programas

4

- **Desvantagens** das linguagens de script
  - ▣ Scripts são mais lentos que código compilado
  - ▣ Não possuem tantos recursos
  
- JavaScript
  - ▣ Primeira linguagem de script da Web
  - ▣ Desenvolvida pela Netscape em 1995
  - ▣ Possui similaridades sintáticas com Java e C++
    - Mais simples
    - Mais limitada

# JavaScript é usado para:

5

- Adicionar comportamento dinâmico a páginas Web
  - ▣ Validar formulários
  - ▣ Gerenciar cookies
  - ▣ Atualizar elementos da página sem refresh (Ajax)
  
- Programar interfaces Web
  - ▣ Botões, imagens clicáveis, janelas popup, etc.

# Tag <script>

6

```
<html>
<!-- js01.html -->
<head>
  <meta charset="utf-8">
  <title>Página JavaScript</title>
</head>

<body>
  <script type="text/javascript">
    // código simples para demonstrar output

    document.write("<p>Hello world!</p>");

    document.write(" <p>Aprendendo<br/> " +
      " <i>JavaScript</i></p> ");
  </script>

  <p>Aqui vai um texto estático.</p>

</body>
</html>
```

- Javascript é inserido nas páginas usando a tag <script>
- document.write exibe texto na página
  - ▣ Pode incluir tags HTML
- Como ocorre em C++/Java, comandos terminam com “;”
- Comentários
  - ▣ //
  - ▣ /\* ... \*/

# Tipos de dados e variáveis

7

- JavaScript possui apenas 3 tipos de dados primitivos
  - ▣ String
    - “Laercio”    ‘como vai?’    “Eu disse ‘Olá’”
  - ▣ Numérico
    - 15    3.14159    1.5E6
  - ▣ Booleano
    - true    false    null

# Tipos de dados e variáveis

8

```
<html>
<!-- js02.html -->
<head>
  <meta charset="utf-8">
  <title>Tipos de dados e variáveis</title>
</head>

<body>
  <script type="text/javascript">
    var x, y;
    x= 1024;

    y=x; x = "foobar";
    document.write("<p>x = " + y + "</p>");
    document.write("<p>x = " + x + "</p>");
  </script>
</body>
</html>
```

- Nomeação de variáveis similar a Java/C++
- Não é necessário declarar variáveis, elas serão criadas na primeira vez que forem usadas
  - ▣ Entretanto é bom declarar com var para fins de controle de escopo
- Variáveis são fracamente tipadas
  - ▣ Uma mesma variável pode ser usada para conter um valor numérico e, posteriormente, uma string



# Operadores e comandos de controle

9

- Comandos de controle (if-else, switch, for, while, etc.) são similares a Java/C++
- Operadores são também similares
  - `+`, `-`, `*`, `/`, `%`, `++`, `--`, ...
  - `==`, `!=`, `<`, `>`, `<=`, `>=`
  - `&&`, `||`, `!`, `===`, `!==`

# Operadores e comandos de controle

10

```
<html>
<!-- js03.html -->
<head>
  <meta charset="utf-8">
  <title>Dobrando papel</title>
</head>

<body>
  <script type="text/javascript">
    var distanceToSun = 93.3e6*5280*12;
    var thickness = .002;

    var foldCount = 0;
    while (thickness < distanceToSun) {
      thickness *= 2;
      foldCount++;
    }
    document.write("<p>Nro de dobras = " +
                  foldCount+"</p>");

  </script>
</body>
</html>
```

## □ Exercício

- Suponha que você pegue um pedaço de papel e o dobre ao meio, depois dobre ao meio novamente, e assim sucessivamente.
- Quantas dobras você terá que fazer até que a grossura do papel dobrado chegue até o sol?

# Funções Matemáticas

11

- Objeto primitivo *Math* provê funções e constantes matemáticas
  - ▣ *Math.sqrt*, *Math.pow*, *Math.abs*, *Math.max*, *Math.min*, *Math.floor*, *Math.round*
- *Math.random* retorna um número real entre 0 e 1

```
<html>
<!-- js04.html -->
<head>
  <meta charset="utf-8">
  <title>Simulando jogo de dados</title>
</head>
<body>
  <div style="text-align:center">
    <script type="text/javascript">
      var jogada1 = Math.floor(Math.random()*6) + 1;
      var jogada2= Math.floor(Math.random()*6) + 1;
      document.write("<img src='http://www.baldochi.unifei.edu.br/'+"
        "COM222/Imagens/die" + jogada1 + ".gif' />");
      document.write("&nbsp;&nbsp;&nbsp;");
      document.write("<img src='http://www.baldochi.unifei.edu.br/'+"
        "COM222/Imagens/die" + jogada2 + ".gif' />");
    </script>
  </div>
</body>
</html>
```

# Interatividade com o comando prompt

12

- Modo rudimentar de implementar interatividade
- Sintaxe: `prompt(arg1, arg2)`
  - `arg1`: A mensagem de prompt que aparece na caixa de diálogo
  - `arg2`: Valor default que aparece na caixa
- Função retorna uma string contendo a digitação do usuário na caixa de diálogo
  - Se o valor for numérico, necessário usar `parseInt` ou `parseFloat`

# Interatividade com o comando prompt

13

```
<html>
<!-- js05.html -->
<head>
  <meta charset="utf-8">
  <title>Página interativa</title>
</head>

<body>  <p>
<script type="text/javascript">
  var userName = prompt("Informe seu nome: ", "");
  var userAge = prompt("Informe sua idade: ", "");
  var userAge = parseFloat(userAge);

  document.write("Olá " + userName + ".")
  if (userAge < 18) {
    document.write("Seus pais sabem que você está online?");
  }
  else {
    document.write("Bem-vindo!");
  }
</script>  </p>
</body>
</html>
```

# Funções definidas pelo usuário

14

- A definição de funções é similar a Java/C++
  - ▣ Nenhum tipo de retorno é especificado e os parâmetros também não são tipados
    - Variáveis são fracamente tipadas
  - ▣ Parâmetros são passados apenas por valor
    - Não é possível passar referências (ponteiros)
- É possível limitar o escopo das variáveis dentro de uma função usando a palavra reservada “var” na primeira vez que a variável for referenciada

# Funções definidas pelo usuário

## Exemplo

15

```
function isPrime(n)
// Assume: n > 0 e n é inteiro
// Retorna: true se n é primo
{
    if (n < 2) {
        return false;
    }
    else if (n == 2) {
        return true;
    }
    else {
        for (var i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

# Funções definidas pelo usuário

## Exemplo

16

```
<html>
<!-- js06.html -->
<head>
  <meta charset="utf-8">
  <title>Testa nro primo</title>
  <script type="text/javascript">
    function isPrime(n){
      // Código mostrado no slide anterior
    }
  </script>
</head>
<body>  <p>
  <script type="text/javascript">
    testNum = parseFloat(prompt("Digite um inteiro positivo", "7"));

    if (isPrime(testNum)) {
      document.write(testNum + " <b>é</b> primo.");
    }
    else {
      document.write(testNum + " <b>nao é</b> primo.");
    }
  </script>  </p>
</body>
</html>
```

Definição de funções deve ocorrer dentro de <head>



## Funções definidas pelo usuário

### Outro exemplo

17

```
<html> <!-- js07.html -->
<head>
  <meta charset="utf-8">
  <title>Simulando jogo de dados </title>
  <script type="text/javascript">
    function randomInt(low, high){
      return Math.floor(Math.random() * (high-low+1)) + low;
    }
  </script>
</head>

<body>
  <div style="text-align: center">
    <script type="text/javascript">
      var jogada1 = randomInt(1, 6);
      var jogada2 = randomInt(1, 6);

      document.write("<img src='http://www.baldochi.unifei.edu.br/" +
        "COM222/Imagens/die" + jogada1 + ".gif' />");
      document.write("&nbsp;&nbsp;&nbsp;");
      document.write("<img src='http://www.baldochi.unifei.edu.br/" +
        "COM222/Imagens/die" + jogada2 + ".gif' />");</script>
    </div>
</body>
</html>
```

Aqui estamos refazendo o exemplo do jogo de dados utilizando uma função de usuário para simular a jogada dos dados

# Bibliotecas JavaScript

18

- Se você definir funções que podem ser úteis em várias páginas, melhor armazenar essas funções num arquivo de biblioteca e carregar esse arquivo quando precisar dessas funções

Exemplo:

<http://www.baldochi.unifei.edu.br/COM222/random.js>

- ▣ Contém as seguintes funções

- `randomNum(low, high)`
- `randomInt(low, high)`
- `randomChar(String)`
- `randomOneOf(item1, ..., itemN)`

# Usando bibliotecas JavaScript

19

```
<html>
<!-- js08.html -->
<head>
  <meta charset="utf-8">
  <title>Jogando dados revisitado</title>

  <script type="text/javascript"
    src="http://www.baldochi.unifei.edu.br/COM222/random.js">
  </script>
</head>

<body>
  <div style="text-align: center">
    <script type="text/javascript">
      var jogada1 = randomInt(1, 6);
      var jogada2 = randomInt(1, 6);

      document.write("<img src='http://www.baldochi.unifei.edu.br/'+"
        "COM222/Imagens/die" + jogada1 + ".gif' />");
      document.write("&nbsp;&nbsp;&nbsp;");
      document.write("<img src='http://www.baldochi.unifei.edu.br/'+"
        "COM222/Imagens/die" + jogada2 + ".gif' />");</script>
    </div>
</body>
</html>
```

# Objetos JavaScript

20

- Um objeto define um **novo tipo de dado**
  - ▣ Formalmente um tipo abstrato de dados (TAD)
- Exemplo: String
  - ▣ Encapsula uma sequência de caracteres entre aspas
  - ▣ Propriedades
    - length
  - ▣ Métodos
    - charAt(index)
    - substring(start, end)
    - toUpperCase()
    - toLowerCase()

# Objetos JavaScript

21

## □ Exemplo: String (cont)

- Para criar uma String, pode-se usar o operador **new** ou fazer uma atribuição direta (new implícito)
  - `word = new String("foo");` OU
  - `word = "foo";`
- Propriedades e métodos são chamados exatamente como em Java/C++
  - `word.length`
  - `word.charAt(0)`

# Exemplo com String

## Testar se String é palíndrome

22

```
// Assume: str eh uma string
// Retorna: str contendo apenas letras e sem espaços
function strip(str) {
  var copy = "";
  for (var i = 0; i < str.length; i++) {
    if ((str.charAt(i) >= "A" && str.charAt(i) <= "Z") ||
        (str.charAt(i) >= "a" && str.charAt(i) <= "z")) {
      copy += str.charAt(i);
    }
  }
  return copy;
}

// Assume: str eh uma string
// Retorna: true se str for palíndrome
function isPalindrome(str) {
  str = strip(str.toUpperCase());

  for (var i = 0; i < Math.floor(str.length/2); i++) {
    if (str.charAt(i) != str.charAt(str.length-i-1)) {
      return false;
    }
  }
  return true;
}
```

isPalindrome

deve retornar true:

-radar

-arara

-Madam, I'm Adam

# Exemplo com String

## Testar se String é palíndrome

23

```
<html>
<!-- js09.html -->
<head>
  <title>Testa palindrome</title>
  <script type="text/javascript">
    function strip(str){
      // Código mostrado no slide anterior
    }
    function isPalindrome(str)
    {
      // Código mostrado no slide anterior
    }
  </script>
</head>
<body>  <p>
  <script type="text/javascript">
    text = prompt("Digite uma palavra ou frase", "Madam, I'm Adam");
    if (isPalindrome(text)) {
      document.write("'" + text + "' <b>eh</b> um palindrome.");
    }
    else {
      document.write("'" + text + "' <b>nao eh</b> um palindrome.");
    }
  </script>  </p>
</body>
</html>
```

# Arrays

24

- Arrays armazenam coleções de itens, que podem ser acessados por meio de índices
  - ▣ Por ser **fracamente tipada**, JavaScript permite armazenar em arrays itens de tipos diferentes
  - ▣ Para criar um array, pode-se alocar espaço usando o operador `new`

```
itens = new Array(10); // aloca espaço para 10, mas  
                        pode crescer  
itens = new Array();  // sem parâmetros, cresce  
                        dinamicamente  
Itens = [0, 0, 0, 0, 0, 0, 0] // criar e inicializar
```



# Arrays

25

- Para acessar, basta usar um índice (igual Java/C++)

```
for (i = 0; i < 10; i++) {  
    items[i] = i;    // inicializa com valores  
                     0, ..., 9  
}
```

- A propriedade length informa o nro de itens no array

```
for (i = 0; i < items.length; i++) {  
    document.write(items[i] + "<br/>");  
    // mostra elementos do vetor  
}
```

# Exemplo de Array

26

```
<html> <!-- js10.html -->
<head>
  <meta charset="utf-8">
  <title>Estatística do dado</title>
  <script type="text/javascript"
    src="http://www.baldochi.unifei.edu.br/COM222/random.js">
  </script>
</head>
<body>  <p>
  <script type="text/javascript">
    numRolls = 60000;
    dieSides = 6;
    rolls = new Array(dieSides+1);
    for (i = 1; i < rolls.length; i++) {
      rolls[i] = 0;
    }
    for(i = 1; i <= numRolls; i++) {
      rolls[randomInt(1, dieSides)]++;
    }
    for (i = 1; i < rolls.length; i++) {
      document.write("Number of " + i + "'s = " + rolls[i] + "<br />");
    }
  </script>  </p>
</body> </html>
```

# Arrays (cont.)

27

Arrays possuem métodos predefinidos que permitem que eles sejam usados como pilhas, filas e outras estruturas de dados

```
var stack = new Array();
stack.push("azul");           // colocar o valor "azul" no vetor
stack.push(12);               // pilha é agora o array ["azul", 12]
stack.push("verde");          // stack = ["azul", 12, "verde"]
var item = stack.pop();        // item é igual a "verde"
                               // e pilha = ["azul", 12]

var q = [1,2,3,4,5,6,7,8,9,10];
item = q.shift();             // item é agora igual a 1, o restante dos
                               // elementos de q movem 1 posição para baixo
                               // dentro do array, e.g. q[0] igual a 2
q.unshift(125);               // q é agora o array [125,2,3,4,5,6,7,8,9,10]
q.push(244);                   // q = [125,2,3,4,5,6,7,8,9,10,244]
```

# Objeto Date

28

- Objeto Date pode ser usado para obter data/hora e para fazer aritmética de datas
- Construtor

```
hoje = new Date()    // Similar a Java  
anoNovo = new Date(2016, 0, 1);
```

- Métodos relevantes

```
newYear.getYear()  
newYear.getMonth()  
newYear.getDay()  
newYear.getHours()  
newYear.getMinutes()  
newYear.getSeconds()  
newYear.getMilliseconds()
```

# Exemplo de Date

29

```
<html> <!-- js11.html -->
<head>
  <meta charset="utf-8">
  <title>Time page</title>
</head>
<body>
  Hora quando a página foi carregada:
  <script type="text/javascript">
    now = new Date();
    document.write("<p>" + now + "</p>");
    time = "AM";
    hours = now.getHours();
    if (hours > 12) {
      hours -= 12;
      time = "PM"
    }
    else if (hours == 0) {
      hours = 12;
    }
    document.write("<p>" + hours + ":" + now.getMinutes() +
      ":" + now.getSeconds() + " " + time + "</p>");
  </script>
</body>
</html>
```

Por default, a hora é impressa no formato longo.

É possível pegar partes do objeto date para imprimir a hora no formato curto.

# Outro exemplo de Date

30

```
<html>
<!-- js12.html -->
<head>
  <meta charset="utf-8">
  <title>Time page</title>
</head>
<body>
  <p>Tempo decorrido este ano:
  <script type="text/javascript">
    now = new Date();
    newYear = new Date(2017,0,1);
    secs = Math.round((now-newYear)/1000);
    days = Math.floor(secs / 86400);
    secs -= days*86400;
    hours = Math.floor(secs / 3600);
    secs -= hours*3600;
    minutes = Math.floor(secs / 60);
    secs -= minutes*60;
    document.write(days + " days, " + hours + " hours, " +
      minutes + " minutes, and " + secs + " seconds.");
  </script>
  </p>
</body>
</html>
```

É possível adicionar e subtrair datas: o resultado é uma quantidade de milisegundos

## Solução:

Obter o número de segundos desde 01 de janeiro e dividir esse número em dias, horas, minutos e segundos

# Objeto document

31

- Browsers tais como IE, Firefox, Chrome, Opera, etc. permitem o acesso a informações de um documento HTML através do objeto *document*

*document.write(...)*

método para exibir texto na página

*document.URL*

propriedade que informa a localização de um documento HTML

*document.lastModified*

propriedade que informa a última modificação de um documento HTML

# Objeto navigator

32

`navigator.appName`

propriedade que informa o nome do browser

`navigator.appVersion`

propriedade que informa a versão browser

```
<!-- MSIE.css -->
```

```
a {text-decoration:none;
    font-size:larger;
    color:red;
    font-family:Arial}
a:hover {color:blue}
```

```
<!-- Netscape.css -->
```

```
a {font-family:Arial;
    color:white;
    background-color:red}
```

```
<html>
<!-- js14.html -->

<head>
  <title>Dynamic Style Page</title>

  <script type="text/javascript">
    if (navigator.appName == "Netscape") {
      document.write('<link rel=stylesheet ' +
        'type="text/css" href="Netscape.css">');
    }
    else {
      document.write('<link rel=stylesheet ' +
        'type="text/css" href="MSIE.css">');
    }
  </script>
</head>

<body>
Here is some text with a
<a href="javascript:alert('GO AWAY')">link</a>.
</body>
</html>
```



# Exercícios

Entrega: 10/08

33

- 1) Escreva um script que leia uma temperatura em Celsius, converta a mesma para Farenheit e exiba o resultado
- 2) Escreva um script que conte as ocorrências de uma substring dentro de uma string. A string e a substring devem ser obtidas do usuário
- 3) Escreva um script que leia uma string e informe a maior palavra contida na string.
- 4) Escreva um script que informe a soma e o produto de um array de inteiros informado pelo usuário.
- 5) Escreva um script que leia a data e a hora de seu nascimento e informe a sua idade em anos, dias, horas, minutos e segundos

# Exercícios

Entrega: 10/08

34

- 6) Elabore um script que leia dois números e imprima em uma tabela (gerada dinamicamente) o resultado das operações aritméticas soma, produto, divisão e resto da divisão (%) entre esses números.

Exemplo:

Operação	Valor
$2 + 1$	3
$2 * 1$	2
$2 / 1$	2
$2 \% 1$	0

# Exercícios

Entrega: 10/08

35

- 7) As Organizações Tabajara resolveram dar um aumento de salário aos seus colaboradores e lhe contrataram para desenvolver o programa que calculará os reajustes. Faça um programa que recebe o salário de um colaborador e calcule reajuste segundo o seguinte critério, baseado no salário atual:
- Salários até R\$ 280,00 (incluindo): aumento de 20%
  - Salários entre R\$ 280,00 e R\$ 700,00: aumento de 15%
  - Salários entre R\$ 700,00 e R\$ 1500,00: aumento de 10%
  - Salários de R\$ 1500,00 em diante: aumento de 5%

Após o aumento ser calculado, deverá ser informado: o salário antes do reajuste; o percentual de aumento aplicado; o valor do aumento; o novo salário, após o aumento.