Sistemas Web I

Aula 2

Tecnologias cliente e tec. servidor

JavaScript

Tiago Cruz de França

tcruz.franca@gmail.com

Agenda

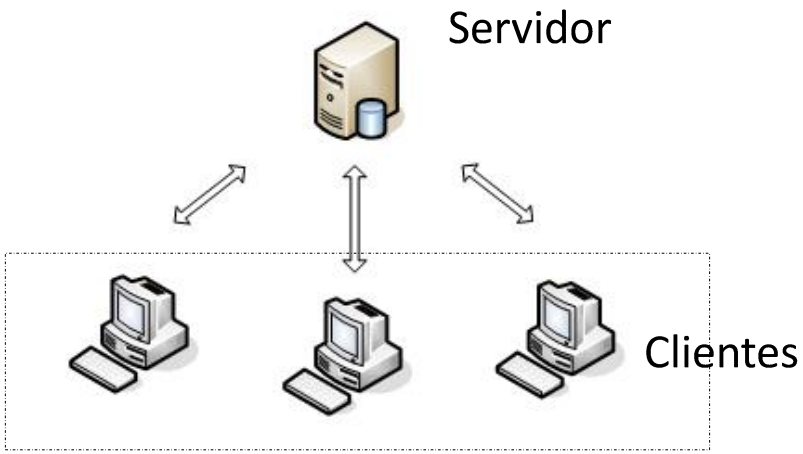
- **Visão geral da web**
  - Conceitos básicos
  - Páginas estáticas e páginas dinâmicas
    - **Dinâmica cliente e servidor**
  - Entender aplicações das principais linguagens de programação no contexto da web
    - **Linguagens Cliente e Linguagens Servidor**
- **JavaScript**

Web

- **Web é um sistema Hipertexto/Hipermídia**
  - A estrutura de uma página web é definida por meio do uso da linguagem HTML
  - Páginas web podem ser
    - **Estáticas**
      - Apenas código HTML e CSS, não existe código na página ou no cliente, todas as funcionalidades são fornecidas apenas por conteúdo HTML e páginas de estilo
    - **Dinâmicas**
      - Cliente realiza tarefas (funções) que exigem processamento para realizar uma lógica programada
      - Servidor possui código especializado para tratar as requisições obtidas das páginas executando tarefas previamente programadas

Web e Internet

- **A web existe sobre a internet**
  - Internet – serviços de rede e transporte
    - **Além da web, existem serviços como FTP, SSH, SMTP, etc.**
  - Web – camada de aplicação que funciona sobre os serviços de rede e transporte
    - **A web é distribuída: modelo cliente-servidor**



A Web é um sistema Hipertexto/Hipermídia

- **O que isso significa?**
  - Existe inúmeras páginas ligadas por hiperlinks
  - As páginas são compostas por vários arquivos (objetos)
    - **Texto**
    - **Imagens**
    - **Vídeo**
    - **Código fonte**
      - HTML, por exemplo

Páginas Estáticas

- **Sem lógicas de programação**
  - Sem funcionalidades mais além dos *links*
    - **Apenas HTML e CSS**
      - Com imagens, textos, etc.
  - Mais fáceis de criar
    - **Não existe lógica, apenas a parte visual**

      - Alunos de SI da UFRRJ aprendem a fazer tudo isso em introdução à web. Lá o aluno também tem o primeiro contato com servidores HTTP

Páginas Dinâmicas

- **Páginas com efeitos e funcionalidades (no cliente e/ou servidor) programadas**
  - Ou seja, são páginas que realizam efeitos especiais ou implementa alguma funcionalidade ou interatividade
  - Uso de linguagem de programação
    - **Lembre-se HTML e CSS não são linguagens de programação, mas de marcação e estilo, respectivamente**
    - **O dinamismo (as funcionalidades programadas) são adicionados ao conteúdo estático (HTML e CSS) das páginas**

Paginas Dinâmicas

- **O que é preciso?**
  - Certamente só HTML e CSS não serão suficientes
  - Linguagens de programação apropriadas para
    - **Rodar no cliente**
    - **Tratar requisições no servidor**

Paginas Dinâmicas no Cliente

- **Processamento é realizado no navegador do usuário**
  - Usada para:
    - **Dar efeitos especiais visuais aos usuários, tratar formulários, manter a página atualizada, realizar cálculos, manipulação da página como se fosse um aplicativo desktop**
  - Código é inserido na página HTML como um script
    - **Principal linguagem: JavaScript**
    - **Vimos na aula passada como inserir JavaScript**
  - Desvantagens
    - **Depende do navegador do cliente**
      - Navegadores podem operar de maneiras diferentes, levando o usuário a experiências distintas

Páginas Dinâmicas no Servidor

- **Processamento realizado no servidor**
  - Páginas atuais combinam funcionalidades programadas no servidor e funcionalidades programadas no cliente
  - Uso: manipulação de banco de dados, centralizar processamento, processar conteúdo sigiloso, etc.
  - Existe a possibilidade de inserir código dentro da página ou em arquivos separados
  - Linguagens
    - **Perl, ASP, Java (Servlet e JSP), PHP, Django, etc.**
  - Vantagens e desvantagens?

JavaScript

- **Linguagem de programação que permite:**
  - Programar funcionalidades dentro das páginas web
    - **Linguagem *client-side*. Navegador é responsável por realizar as tarefas programadas**
- **É simples**
  - Pensada para se criar soluções rápidas para os problemas
- **O que dá pra fazer**
  - Programar efeitos especiais para página
  - Criar páginas interativas

Incluindo código JavaScript na Página

```
<script type="text/javascript">
//direto na página
   alert('Olá mundo!');
   </script>
```

```
<script type="text/javascript" src="path-do-meu-arquivo.js">
//chamando arquivo com codigo (pode ser uma URL)
</script>
```

```
<button type="button" onclick="alert('ao clicar do mouse')">
<!– Direto no evento de um componente HTML -->
```

JavaScript - Exemplos

```
//Abrir uma segunda janela

<script>
window.open("http://www.google.com","","width=550,height=420,menubar=no")
</script>
```

```
//Botão voltar

<input type=button value=Atrás onclick="history.go(-1)">
```

JavaScript - Sintaxe

- **Semelhante ao C**
  - Como várias linguagens
- **Variáveis**
  - Tipagem dinâmica
    - **Não precisa declarar o tipo ao criar a variável**

```
var a;
a =  “Web 1”;
var i = 10;
i = 10 – 5 * 2 / 2;
```

JavaScript – Peradores Aritméticos

Operador	Descrição
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo
++	Incrementar
--	Decrementar

JavaScript – Operadores de Atribuição

Operador	Exemplo	Equivalente a
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

JavaScript – Operadores de String

- **“+”, usado com string, leva ao efeito conhecido como contração**

```
txt1 = "What a very";
txt2 = "nice day";
txt3 = txt1 + txt2; //resultado: What a verynice day
```

- **String e Números**

```
x = 5 + 5;
y = "5" + 5;
z= "Hello" + 5;
```

```
10
55
Hello5
```

Exercícios

- **Crie um código javascript para somar dois números e imprimir o resultado na página**
  - Faça o mesmo para apresentar:
    - **O resultado da subtração desses dois números**
    - **Da multiplicação**
    - **Da divisão**
    - **O módulo da divisão**
- **Use o operador "+=" para adicionar 2 a uma variável x**

JavaScript – Tipos de Dados

```
var length = 16;                       // Number
var lastName = "Johnson";              // String
var cars = ["Saab", "Volvo", "BMW"];   // Array
var x = {firstName:"John", lastName:"Doe"};   // Object
```

```
//Desafio – sem tipo, o que isto gera:
var x = 16 + "Volvo";
var x = 16 + 4 + "Volvo";
var x = "Volvo" + 16 + 4;
```

Adaptado da W3C

JavaScript – Tipos Dinâmicos e String

```
var x;               // Agora x esta indefinido
var x = 5;           // Agora e um numero
var x = "John";      // Agora uma string
```

```
var carName = "Volvo XC60";   // aspas duplas
var carName = 'Volvo XC60';   // aspas simples
```

JavaScript - Booleanos, Arrays e Mapas (objetos)

- **Booleans**

```
var x = true;
var y = false;
```

- **Array**

```
var cars = ["Saab", "Volvo", "BMW"];
```

- **Objetos**

```
var person = {firstName:"John", lastName:"Doe",
age:50, eyeColor:"blue"};
```

Verificando o Tipo do Operador

```
typeof "John"                 // Returns string
typeof 3.14                   // Returns number
typeof false                  // Returns boolean
typeof [1,2,3,4]              // Returns object
typeof {name:'John', age:34}  // Returns object
```

```
var person;                   // The value is undefined, the typeof is undefined
var car = "";                 // The value is "", the typeof is string
```

Funções JavaScript

- **Simples, basta usar a palavra “function” e dá uma nome a função**
  - Podem receber parâmetros e retornar valores
    - **Tipos não precisam ser definidos previamente**

```
//Sintaxe
function mensagem(){
            alert(“Acorda!!!”);
}

function dividir(var a, var b){ //sem tipo para variavel
             return a / b; //sem tipo para o retorno
}
```

Função - Exemplo

```
var x = myFunction(4, 3);       // Function is called, return value will end up in x

function myFunction(a, b) {
   return a * b;                // Function returns the product of a and b
}
```

```
function toCelsius(fahrenheit) {
   return (5/9) * (fahrenheit-32);
}
document.getElementById("demo").innerHTML = toCelsius(32);
```

Estrutura de Controle de Fluxo

- **if ... else**

```
//Sintaxe
if (condicao_1){

        //codigo

}
else if(condicao_2){

      //codigo

}else {

      //codigo

}
```

Exemplo if-else

```
if(idade > 18)
{
alert(“É maior de idade”);
}
else
{
alert(“É menor de idade”);
}
```

Usando swtich

```
//Sintaxe
switch(variável)
{
case valor1:
//ações caso valor1
break;
case valor2:
//ações caso valor2
break;
case valor3:
//ações caso valor3
break;
default:
//ações caso nenhum dos valores
break
}
```

Exemplo do uso de switch

```
switch(dia)
{
case 1:
alert(“Hoje é domingo”);
break;
case 2:
alert(“Hoje é segunda”);
break;
case 3:
alert(“Hoje é terça”);
break;
default:
alert(“Hoje não é nem domingo, nem segunda, nem terça”);
break
}
```

Laço – while e do-while (sintaxe e exemplo)

```
//sintaxe
while(condição)
{
//ações
}
```

Exemplo

```
var contador = 0;
while(contador % 2 == 0)
{
alert("Olá");
contador = contador + 1;
}
```

```
do
{
   //ações
}
while(condição)
```

Exemplo

```
var contador = 0;
do
{
alert("Oi");
contador = contador + 1;
}
while(contador <= 4)
```

Laço - For

```
//sintaxe
for(inicializa; cond; complem)
{
   //ações
}
```

Exemplo

```
var contador;
for(contador = 0; contador < 10;
contador++)
{
   alert(contador);
}
```

Exercícios[1]

- **Crie funções para:**
  - Calcular a sequencia de Fibonacci dado um valor máximo
  - Calcular todos os números perfeitos em um intervalo
    - **Valor de início e fim**
  - Identificar se um número é primo
  - Calcular o fatorial de um número inteiro
  - Dados três valores de comprimentos de retas, verificar se as mesmas podem formar um triângulo
    - **Se sim, qual tipo de triângulo**
  - Multiplicar duas matrizes 3x3
  - Verificar se um número de CPF é válido