



Universidade de Ribeirão Preto
Campus Ribeirão Preto - Campus Guarujá

Programação para Web II

JavaScript

Prof. Eliézer Zarpelão

Schedule

- History
- Concepts
 - server-side/client-side
 - DOM
- Conventions JS
- Challenge

"Part of being a good steward to a successful project is realizing that writing code for **yourself** is a **bad Idea**TM. If thousands of people are using your code, then write your code for maximum **clarity**, not your **personal preference** of how to get clever within the spec."

Idan Gazit

Client-side vs Server-side

- Web pages more dynamic
- Customized content based on user input
- Where application code runs

Server-Side script

```
<?php
    include('functions.php');
    showTop('Home');
?>

<!-- The main body of your web page, in HTML, will
go here -->

<?php
    showBottom();
?>
```

Client-side Scripts

- JavaScript / VBScript
- Older browsers don't support them
- Newer browsers allow users to disable scripts
- AJAX - Asynchronous JavaScript and XML

JavaScript History

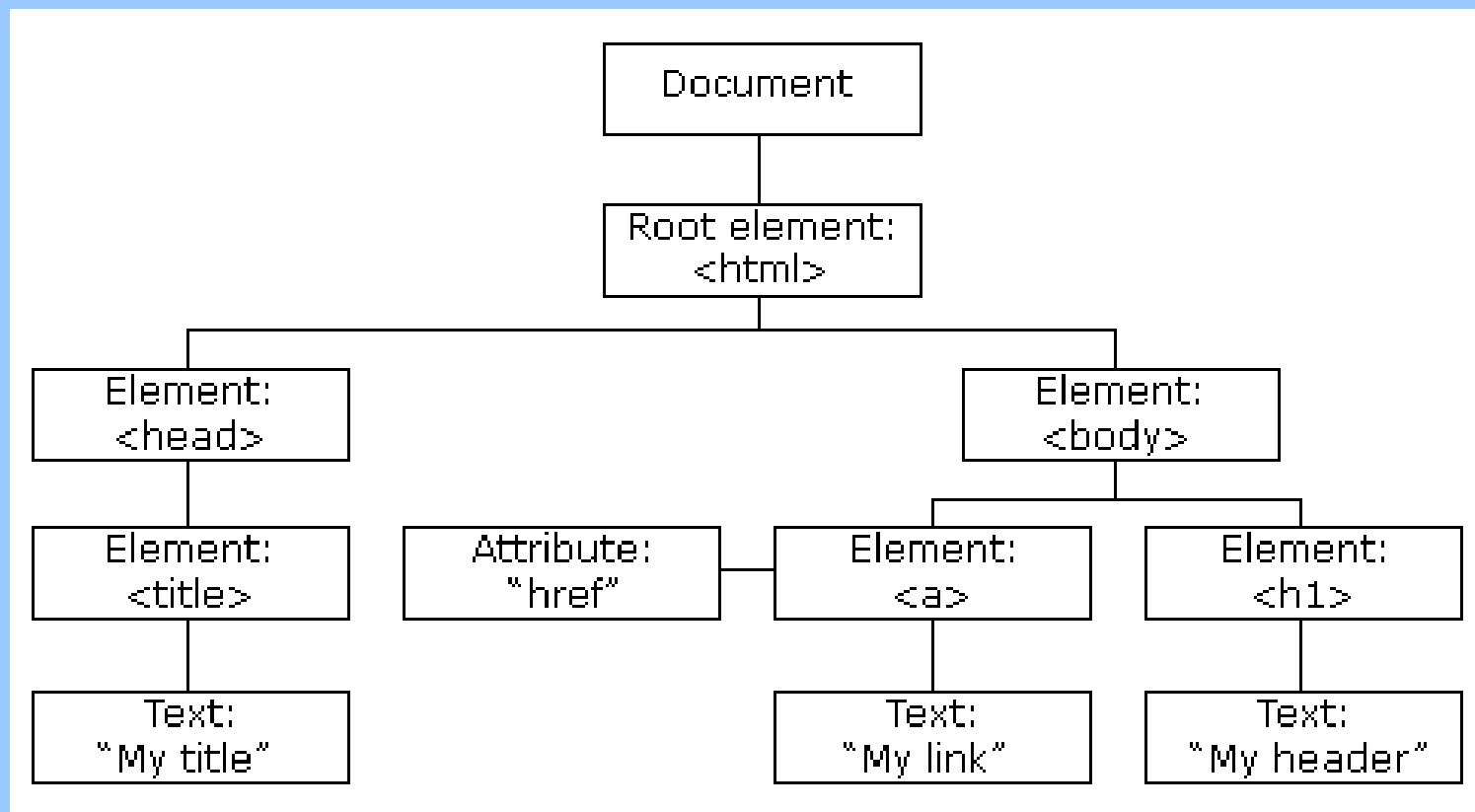
- Netscape 1995 – HTML extension
- Manipulation of HTML documents
- Form validation
- Mocha > LiveScript
- = Java?
- Interpreted language

Conventions

- Use sempre var
- NAMES_LIKE_THIS for constant values
- Always use semicolons
- Function Declarations Within Blocks: NO!
- With() {}: NO!
- Strings: Prefer ' over "
- <https://google.github.io/styleguide/javascriptguide.xml>
- https://github.com/rwaldron/idiomatic.js/tree/master/translations/pt_BR

DOM

- Document Object Model



JavaScript – Dynamic HTML

- Change all the HTML elements in the page
- Change all the HTML attributes in the page
- Change all the CSS styles in the page
- Remove existing HTML elements and attributes
- Add new HTML elements and attributes
- React to all existing HTML events in the page
- Create new HTML events in the page

JS - Strings & Arrays

- Split
- Join
- Reverse

Challenge 01

Local / jsbin.com / TutorialsPoint – Coding
Ground

```
function FirstFactorial(num) {
```

```
    // code goes here
```

```
    return num;
```

```
}
```

Challenge 01

- function FirstFactorial(num) {
 - var result = 1;
 - for (var i=1;i<=num;i++){
 - result *= i;
 - }
 - return result;
 - }
- console.log(FirstFactorial(4));
- Case 1
 - Input 4
 - Output 24
- Case 2
 - Input 10
 - Output 3628800

Challenge 02

- Reverse a string
 - function FirstReverse(str) {
 // code goes here
 return str;
}
- Case 1
 - Input:"palmeiras"
 - Output:sariemlap
- Case 2
 - Input:"nao tem mundial"
 - Output:laidnum met oan

Challenge 02

- Reverse a string
 - ```
function FirstReverse(str) {
 // code goes here
 return str;
}
```
- Case 1
  - Input: "palmeiras"
  - Output: "sariemlap"
- Case 2
  - Input: "nao tem mundial"
  - Output: "laidnum met oan"



# Challenge 03

- `function TimeConvert(num) {  
 // code goes here  
 return num;  
}`
- Case 1
  - Input: 129
  - Output: 2:9
- Case 2
  - Input: 33
  - Output: 0:33
- `parseInt(2.4) // 2`

# Challenge 03

```
function TimeConvert(num) {

 // code goes here
 return parseInt(num/60).toString() + ':' + (num%60).toString();

}
```

# JS Challenge 01

- <https://github.com/elizarp/unaerp>
- Only display the email paragraph when the checkbox is checked
  - Hiding the email fields by default
  - Adding a function toggleEmail
  - Display the email fields
  - Making the email field toggle
  - When Javascript is disabled

# JS Challenge 02

```
<fieldset>
 <legend>Billing Information</legend>
 <p>
 <label>
 Postal Address:

 <input name="postaladdress" id="postaladdress"></input>
 </label>
 </p>
 <p>
 Home Address:

 <label>
 <input type="checkbox" name="homepostalcheck" id="homepostalcheck">
 Same as above
 </label>

 <input name="homeaddress" id="homeaddress"></input>
 </p>
</fieldset>
```

# JS Challenge 02

- Add js\_challenge02.txt
- Automatically copy the postal address to the home address field if a user checks the "Same as above" checkbox
- Disable the home address field when the checkbox is checked
- Steps
  - Add a function setHomeAddress
  - Get the Postal Address
  - Set the value for the Home Address
  - Disable the Home Address field
  - Toggle the Home Address

# JS Challenge 03

- Radio Buttons
- Add js\_challenge03.txt
- Steps
  - Set up your input validation
  - Find out if a radio button has been checked
  - Getting it to work properly
  - Halting progress

# JS Challenge 04

- Download js\_challenge04.html
- Create in-line error messages that only appear when they need to
- Steps
  - Add CSS - error messages use red text
  - Set up your checkForm() function
  - Display your error messages

# JS Challenge 05

- Send code to [eliezer.zarpelao@gmail.com](mailto:eliezer.zarpelao@gmail.com)
  - Until next class
  - Subject: “JS Challenge 05 - “+ RA
- E-mail validation
- Steps
  - Add an email field
  - Make sure that the '@' symbol is present
  - Ensure there's something before the @
  - Ensure there's a full stop (.) after the @
  - Ensure there's at least two characters after the dot
  - Advanced: doing it all with a Regular Expression (regex)



# Questões Concursos

(UFRR 2019) JavaScript é uma linguagem de script orientada a objetos, multiplataforma. É uma linguagem pequena e leve que pode rodar num navegador web. Analise o trecho de código abaixo, escrito em JavaScript.

Sobre esse trecho, assinale a afirmação INCORRETA.

```
var signo = prompt("Qual é o seu signo?");

if (signo.toLowerCase() == "escorpião") {
 alert("Legal! Eu sou de Escorpião também!");
}
else {
 alert ("Hoje seu dia será maravilhoso");
}
```

- A - Uma caixa de diálogo prompt contém uma caixa de texto de linha única, um botão Cancelar, e um botão OK, e retorna o texto que o usuário digitou naquela caixa de texto
- B - A caixa de diálogo de aviso (alert) deve ser utilizada para mensagens que não requeiram nenhuma resposta da parte do usuário, a não ser o reconhecimento da mensagem
- C - A função toLowerCase() irá converter o texto da variável signo em letra minúscula
- D - Na primeira linha do código, ao declarar a variável 'signo' o tipo de dado string deve ser especificado para o funcionamento do script ocorrer sem erros;
- E - Para esse código JavaScript ser executado não há necessidade de instalação de complemento no navegador Mozilla Firefox

# Questões Concursos

(UFRR 2019) JavaScript é uma linguagem de script orientada a objetos, multiplataforma. É uma linguagem pequena e leve que pode rodar num navegador web. Analise o trecho de código abaixo, escrito em JavaScript.

Sobre esse trecho, assinale a afirmação INCORRETA.

```
var signo = prompt("Qual é o seu signo?");

if (signo.toLowerCase() == "escorpião") {
 alert("Legal! Eu sou de Escorpião também!");
}
else {
 alert ("Hoje seu dia será maravilhoso");
}
```

- A - Uma caixa de diálogo prompt contém uma caixa de texto de linha única, um botão Cancelar, e um botão OK, e retorna o texto que o usuário digitou naquela caixa de texto
- B - A caixa de diálogo de aviso (alert) deve ser utilizada para mensagens que não requeiram nenhuma resposta da parte do usuário, a não ser o reconhecimento da mensagem
- C - A função toLowerCase() irá converter o texto da variável signo em letra minúscula
- D - Na primeira linha do código, ao declarar a variável 'signo' o tipo de dado string deve ser especificado para o funcionamento do script ocorrer sem erros;
- E - Para esse código JavaScript ser executado não há necessidade de instalação de complemento no navegador Mozilla Firefox

# Dúvidas

Maior dúvida – enviar via e-mail para  
[ezarpelao@unaerp.br](mailto:ezarpelao@unaerp.br) com assunto  
“Maior dúvida – 09/08/2019 - “+RA

