

JavaScript.js

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

Tecnologias Web

AJAX

Asynchronous JavaScript And XML

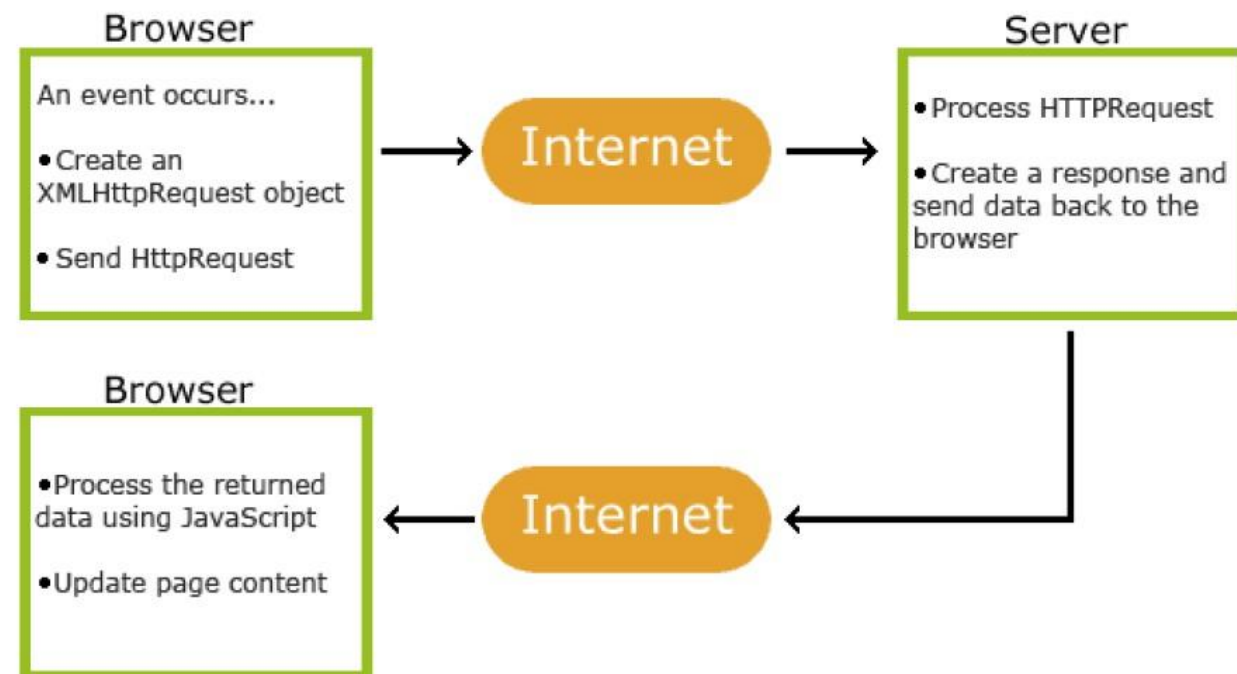
HTML – CSS – JavaScript – PHP - AJAX





Como funciona?

1. Um evento ocorre em uma página da web (a página é carregada, um botão é clicado)
2. Um objeto XMLHttpRequest é criado por JavaScript
3. O objeto XMLHttpRequest envia um pedido para um servidor web
4. O servidor processa a solicitação
5. O servidor envia uma resposta de volta à página da web
6. A resposta é lida por JavaScript
7. A ação adequada (como atualização de página) é realizada por JavaScript



JavaScript.js



XMLHttpRequest

```
1
2
3
4
5
6
7
8
9
10 function loadDoc() {
11     var xhttp = new XMLHttpRequest();
12     xhttp.onreadystatechange = function() {
13         if (this.readyState == 4 && this.status == 200) {
14             document.getElementById("demo").innerHTML = this.responseText;
15         }
16     };
17     xhttp.open("GET", "demo_get.asp", true);
18     xhttp.send();
19 }
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
```

Source

https://www.w3schools.com/xml/tryit.asp?filename=tryajax_get

<https://developer.mozilla.org/pt-BR/docs/Web/API/XMLHttpRequest>



JQUERY - AJAX

```
1  
2  
3  
4  
5  
6  
7  
8 $.ajax({  
9     url: 'superman',  
10    type: 'POST',  
11    data: jQuery.param({ field1: "hello", field2 : "hello2"}) ,  
12    contentType: 'application/x-www-form-urlencoded; charset=UTF-8',  
13    success: function (response) {  
14        alert(response.status);  
15    },  
16    error: function () {  
17        alert("error");  
18    }  
19 });
```

Source:
<https://api.jquery.com/jquery.ajax/>

JavaScript.js



JQUERY - AJAX

```

1
2
3
4
5 $.ajax({
6     type: 'POST',
7     url: url,
8     data: data,
9     beforeSend: function() {
10         // setting a timeout
11         $(placeholder).addClass('loading');
12         i++;
13     },
14     success: function(data) {
15         if (append) {
16             $(placeholder).append(data);
17         } else {
18             $(placeholder).html(data);
19         }
20     },
21     error: function(xhr) { // if error occurred
22         alert("Error occurred.please try again");
23         $(placeholder).append(xhr.statusText + xhr.responseText);
24         $(placeholder).removeClass('loading');
25     },
26     complete: function() {
27         i--;
28         if (i <= 0) {
29             $(placeholder).removeClass('loading');
30         }
31     },
32     dataType: 'html'
33 });
34
35
36
37

```


JavaScript.js



FETCH - AJAX

```
1
2
3
4
5
6 fetch('https://jsonplaceholder.typicode.com/posts', {
7     method: 'POST',
8     data: {
9         title: 'foo',
10        body: 'bar',
11        userId: 1
12    },
13    credentials: 'include',
14    headers: {}
15  })
16  .then((response) => {
17    return response.json();
18  })
19  .then((jsonObject) => {
20    console.log(jsonObject)
21    document.write(`ID ${jsonObject.id} was created!`);
22  })
23  .catch((error) => {
24    document.write(error);
25  });
26
27
28
29
30
31
32
33
34
35
36
37
```

Source

https://developer.mozilla.org/pt-BR/docs/Web/API/Fetch_API/Using_Fetch

JavaScript.js

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

Tecnologias Web

Forms

HTML – CSS – JavaScript – PHP - AJAX



<!-- João Vieira 2020 - 2021 -->



Forms - Elementos

- INPUT (text, number, password, radio, check, ...)
- TEXTAREA
- BUTTON
- SELECT
 - OPTIONS
 - OPTIONGROUP
- FIELDSET
- LABEL



Forms - Atributos

accept-charset: Especifica as codificações de caracteres usadas para o envio do formulário

action: Especifica para onde enviar os dados do formulário quando um formulário é enviado

autocomplete: Especifica se um formulário deve ter o autocomplete ativado ou desativado

enctype: Especifica como os dados do formulário devem ser codificados ao enviá-los ao servidor (somente para method="post")

method: Especifica o método HTTP a ser usado ao enviar dados de formulário

name: Especifica o nome do formulário

novalidate: Especifica que o formulário não deve ser validado quando submetido

rel: Especifica a relação entre um recurso vinculado e o documento atual

target: Especifica onde exibir a resposta recebida após o envio do formulário

JavaScript.js



Forms

```

1
2
3
4
5
6
7
8
9
10
11 <form action="/action_page.php" method="GET">
12   First name:<br>
13   <input type="text" name="firstname" value="Mickey">
14   <br>
15   Last name:<br>
16   <input type="text" name="lastname" value="Mouse">
17   <br><br>
18   <input type="submit" value="Submit">
19   <input type="reset" value="Reset">
20
21 </form>
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

```

First name:

Mickey

Last name:

Mouse

Submit



Forms - Radio

```

1
2
3
4
5
6
7
8
9 <form action="/action_page.php">
10
11   <label for="male">Male</label>
12
13   <input type="radio" name="gender" id="male" value="male"><br>
14
15   <label for="female">Female</label>
16
17   <input type="radio" name="gender" id="female" value="female"><br>
18
19   <label for="other">Other</label>
20
21   <input type="radio" name="gender" id="other" value="other"><br><br>
22   <input type="submit" value="Submit">
23
24 </form>
25
26
27
28
29
30
31
32
33
34
35
36
37

```



Forms -

```
<input id="id1" type="number" min="100" max="300" required>  
<button onclick="myFunction()">OK</button>
```

```
<p>If the number is less than 100 or greater than 300, an error message will  
be displayed.</p>
```

```
<p id="demo"></p>
```

```
<script>  
function myFunction() {  
    var inpObj = document.getElementById("id1");  
    if (!inpObj.checkValidity()) {  
        document.getElementById("demo").innerHTML = inpObj.validationMessage;  
    } else {  
        document.getElementById("demo").innerHTML = "Input OK";  
    }  
}  
</script>
```




Forms – JQuery Validações

Precisa de ser utilizado um plugin de jQuery:
jQuery Validation Plugin

Incluir o plugin como script

<https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.19.3/jquery.validate.js>

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.19.3/jquery.validate.js"></script>
```

Source:

<https://jqueryvalidation.org/files/demo/>
<https://jqueryvalidation.org/>

```
$("#form").validate({
  rules: {
    firstname: "required",
    lastname: "required",
    email: {
      required: true,
      email: true
    },
    password: {
      required: true,
      minlength: 5
    }
  },
  messages: {
    firstname: "Please enter your firstname",
    lastname: "Please enter your lastname",
    password: {
      required: "Please provide a password",
      minlength: "Your password must be at least 5 characters long"
    },
    email: "Please enter a valid email address"
  },
  submitHandler: function(form) {
    form.submit();
  }
});
```