

Aula 08 - Atividades

- Criar uma pasta `Aula08` no projeto `AtividadesPHP`
- Criar `index.html` na pasta `Aula08`
- Fazer as atividades abaixo

Atividade 1

1. Criar o arquivo `atividade1.html`
2. Faça um *link* na `index.html` para acessar a página `atividade1.html`
3. Criar uma página `request_method.php`
4. Criar um *link* na página `atividade1.html` para acessar a página `request_method.php`
5. Criar um `<form>` na página `atividade1.html` apenas com um botão de `submit`, para enviar uma requisição para a página `request_method.php` usando o método `POST`
6. Na página `request_method.php` escreva um programa para seguir a seguinte lógica:
 1. Se o método utilizado para acessar a página `request_method.php` for o `GET`, envie uma página de resposta com a mensagem "Bem-vindo"
 2. Se o método for o `POST`, envie uma página de resposta com a mensagem "Request feito com sucesso"

Atividade 2

1. Criar o arquivo `atividade2.html`
 2. Faça um *link* na `index.html` para acessar a página `atividade2.html`
 3. Criar uma página `tipo_do_valor.php`
 4. Criar um *link* na página `atividade2.html` para acessar a página `tipo_do_valor.php`
 5. Na página `tipo_do_valor.php` escreva um programa para seguir a seguinte lógica:
 1. Se o método utilizado para acessar a página `tipo_do_valor.php` for o `GET`, envie uma página HTML de resposta que contém um `<form>` com um `<input>` e que utiliza o método `POST` para enviar os dados. Esse `<input>` vai receber um valor do usuário, que enviará esse valor ao clicar no botão de `submit`
 2. Se o método utilizado para acessar a página for o `POST`, escreva um código para enviar uma página de resposta informando o **tipo** do valor que foi enviado pelo usuário (`int`, `string`, `float`, etc.)
-

Atividade 3

1. Criar o arquivo `atividade3.html`
2. Faça um *link* na `index.html` para acessar a página `atividade3.html`
3. Criar uma página `lancador_de_dados.php`
4. Criar um *link* na página `atividade3.html` para acessar a página `lancador_de_dados.php`
5. Na página `lancador_de_dados.php` escreva um programa para seguir a seguinte lógica:
 1. Se o método utilizado para acessar a página for o `GET`, envie uma página HTML de resposta que contém um `<form>` com dois `<input>` e que utiliza o método `POST` para enviar os dados. Esses dois `<input>` vão receber valores numéricos, que são os valores mínimo e máximo que o dado pode lançar
 2. Se o método utilizado para acessar a página for o `POST`, escreva um código para sortear um valor **entre** os valores mínimo e máximo informados pelo usuário e envie uma página de resposta informando o valor sorteado.

Estruturas para fazer os códigos acima

```
1  is_numeric( $var ) // se o valor da variável $var for numérico, retorna
   true. Caso contrário, retorna false
2
3  is_int( $var ) // se o valor da variável $var for int, retorna true. Caso
   contrário, retorna false
4
5  is_string( $var ) // se o valor da variável $var for string, retorna true.
   Caso contrário, retorna false
6
7  is_float( $var ) // se o valor da variável $var for float, retorna true.
   Caso contrário, retorna false
8
9  rand( $minimo, $maximo ) // retorna um valor entre os valores das variáveis
   $minimo e $maximo
10
11 $_SERVER["REQUEST_METHOD"] // retorna o método do request ("GET" ou "POST")
12
13 $_GET["chave"] // retorna o valor que o usuário inseriu no input com
   name="chave"
14
15 $_POST["chave"] // retorna o valor que o usuário inseriu no input com
   name="chave" (quando o form é configurado para usar o método POST)
```