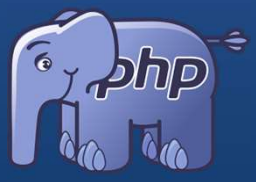
Treinamento Fundamentos de PHP

Turma 2023/01 – Aula 4/8



1

Semana 1

Aula 4 – Tratamento de erros

- Pilha de execução
- Erros e exceções
- Tratamento de exceções
- Lançamento de exceções
- KAHOOT

2

Recapitulando...

Na última aula vimos sobre:

- Funções definidas pelo usuário
  - Argumentos de funções
- Escopo
  - Global, local e superglobais
- Funções anônimas (closure)
- Arrow function
- Inclusão de arquivos php

- PHP na web
  - Subindo o servidor de desenvolvimento PHP
  - PHP com HTML
- Atividade 2: Alterar jogo TicTacToe
- Vamos complementar hoje
  - Mais sobre HTML
  - Como pegar informações do navegador para o servidor

3

PHP na web

Formulários HTML

- Usado para pegar entrada de dados do usuário
- O submit envia os dados do formulário para a url informada no "action" pelo método especificado (GET ou POST)

```
<h2>Formulário</h2>
<form action="pagina.php" method="post">
    <label for="nomeFuncionario"><b> Nome do funcionário</b>:</label>
    <input type="text" name="nomeFuncionario" id="nomeFuncionario"><br>

    <span><b> Alocação</b>:</span>
    <label for="alocadoSim">Sim</label>
    <input type="radio" name="alocado" id="alocadoSim">
    <label for="alocadoNao">Não</label>
    <input type="radio" name="alocado" id="alocadoNao"><br>

    <span><b> Habilidades: </b></span>
    <label for="habilidadePHP">PHP</label>
    <input type="checkbox" name="habilidades" id="habilidadePHP">
    <label for="habilidadeJava">Java</label>
    <input type="checkbox" name="habilidades" id="habilidadeJava"><br>

    <label for="local"><b> Local</b></label>
    <select name="local" id="local">
        <option value="0" selected>Selecione</option>
        <option value="Recife">Recife</option>
        <option value="São Paulo">São Paulo</option>
        <option value="Belo Horizonte">Belo Horizonte</option>
        <option value="Campinas">Campinas</option>
    </select>

    <br><br>
    <input type="submit" value="Enviar">
    <input type="reset" value="Limpar formulário">
</form>
```


4

PHP na web

Formulários HTML

- Método GET
  - Os dados são passados pela URL
  - O comprimento da URL é limitado (cerca de 3.000 caracteres)
  - NÃO pode ser usado para dados confidenciais
- Método POST
  - Os dados são passados no corpo da solicitação HTTP
  - É mais seguro
  - Tamanho ilimitado

```
<h2>Formulário</h2>
<form action="pagina.php" method="post">
    <label for="nomeFuncionario"><b> Nome do funcionário</b>:</label>
    <input type="text" name="nomeFuncionario" id="nomeFuncionario"><br>

    <span><b> Alocação</b>:</span>
    <label for="alocadoSim">Sim</label>
    <input type="radio" name="alocado" id="alocadoSim">
    <label for="alocadoNao">Não</label>
    <input type="radio" name="alocado" id="alocadoNao"><br>

    <span><b> Habilidades: </b></span>
    <label for="habilidadePHP">PHP</label>
    <input type="checkbox" name="habilidades" id="habilidadePHP">
    <label for="habilidadeJava">Java</label>
    <input type="checkbox" name="habilidades" id="habilidadeJava"><br>

    <label for="local"><b> Local</b></label>
    <select name="Local" id="local">
        <option value="0" selected>Selecione</option>
        <option value="Recife">Recife</option>
        <option value="São Paulo">São Paulo</option>
        <option value="Belo Horizonte">Belo Horizonte</option>
        <option value="Campinas">Campinas</option>
    </select>

    <br><br>
    <input type="submit" value="Enviar">
    <input type="reset" value="Limpar formulário">
</form>
```


5

PHP na web

Formulários HTML

- Para pegar os dados vindos da requisição HTTP, no PHP, utilizamos as variáveis superglobais **$_GET** e **$_POST**

```
<h2>Formulário</h2>
<form action="pagina.php" method="post">
    <label for="nomeFuncionario"><b> Nome do funcionário</b>:</label>
    <input type="text" name="nomeFuncionario" id="nomeFuncionario"><br>

    <span><b> Alocação</b>:</span>
    <label for="alocadoSim">Sim</label>
    <input type="radio" name="alocado" id="alocadoSim">
    <label for="alocadoNao">Não</label>
    <input type="radio" name="alocado" id="alocadoNao"><br>

    <span><b> Habilidades: </b></span>
    <label for="habilidadePHP">PHP</label>
    <input type="checkbox" name="habilidades" id="habilidadePHP">
    <label for="habilidadeJava">Java</label>
    <input type="checkbox" name="habilidades" id="habilidadeJava"><br>

    <label for="local"><b> Local</b></label>
    <select name="Local" id="local">
        <option value="0" selected>Selecione</option>
        <option value="Recife">Recife</option>
        <option value="São Paulo">São Paulo</option>
        <option value="Belo Horizonte">Belo Horizonte</option>
        <option value="Campinas">Campinas</option>
    </select>

    <br><br>
    <input type="submit" value="Enviar">
    <input type="reset" value="Limpar formulário">
</form>
```


6

Pilha de execução (*Call Stack*)

- Como em qualquer outra linguagem de programação, o *call stack* (ou "pilha de chamadas") é uma estrutura de dados que registra as chamadas de função ou método em execução na aplicação. Quando uma função ou método é chamado, um novo frame é adicionado à pilha, contendo informações sobre a função ou método chamado, bem como sobre o estado atual da execução da aplicação.

- O *call stack* permite que a aplicação saiba em qual ponto da execução se encontra, o que é importante para garantir que as chamadas de função sejam executadas e encerradas na ordem correta, sem conflitos ou erros.

7

Pilha de execução (*Call Stack*)

- É importante entender a pilha de execução para conseguir identificar erros ocorridos, principalmente quando se trata de projetos grandes

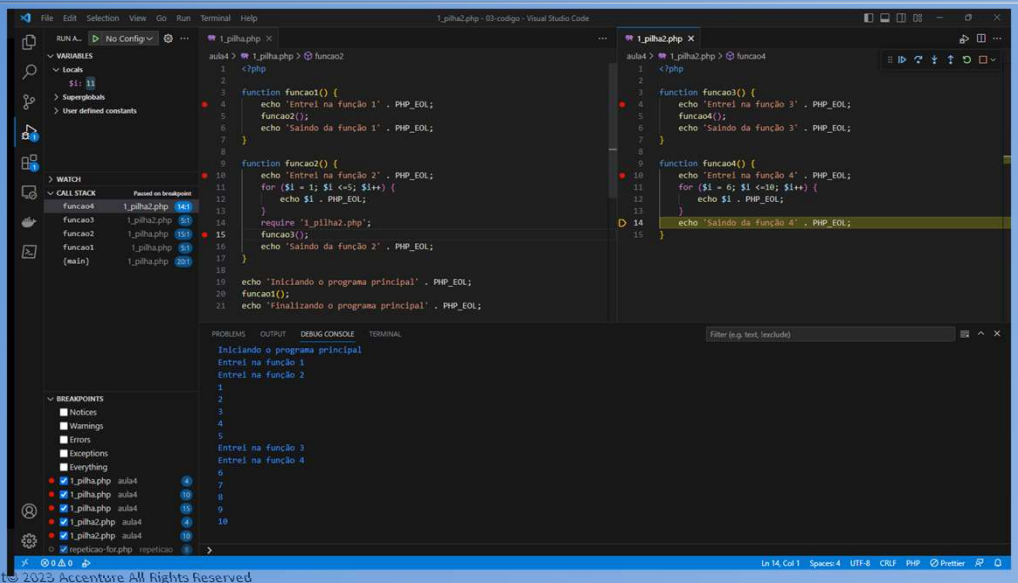
- Existem 3 tipos de erros que podem ocorrer em um programa: de sintaxe, de execução e de semântica

- Para identificar e corrigir erro, devemos depurar o programa

8

Pilha de execução (*Call Stack*)

Debug



Erros e Exceções

- No PHP, **erros** e **exceções** são duas formas de lidar com situações indesejadas ou excepcionais que ocorrem durante a execução de um programa.
- Embora ambas se refiram a problemas no código, elas são tratadas de maneiras diferentes.

Erros e Exceções

Erros:

- são problemas que geralmente indicam falhas no funcionamento do PHP ou violações nas regras da linguagem
- interrompem a execução normal do programa
- Exemplos comuns de erros são "*Parse error: syntax error*", "*Fatal error: Call to undefined function*"

Exceções:

- são eventos excepcionais ou situações anormais que ocorrem durante a execução de um programa, mas que podem ser previstas e tratadas
- Podem ser capturadas e tratadas por meio de blocos ***try-catch***, permitindo que o programa lide com o problema de forma adequada e continue sua execução normalmente

11

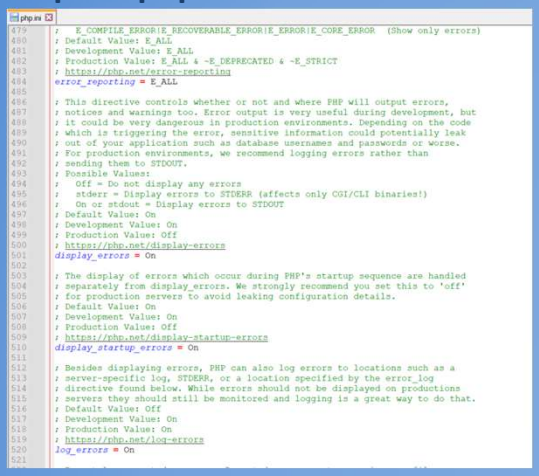
Erros e Exceções

- O PHP retornará erros em resposta a inúmeras condições de erros internos. Pode ser utilizado para sinalizar inúmeras condições diferentes, e podem ser exibidos e/ou registrados em log se necessário.
- Cada erro que o PHP gera inclui um tipo. A lista destes tipos de erros está disponível, junto a uma breve descrição de seu comportamento e como podem ser causados: https://www.php.net/manual/pt_BR/errorfunc.constants.php
- Se nenhuma manipulação de erro for definida, o PHP tratará todos os que ocorrerem de acordo com sua configuração

12

Erros e Exceções

Arquivo php.ini



Dica para localizar o php.ini

```
$ php --ini
Configuration File (php.ini) Path:
Loaded Configuration File:         C:\php\php.ini
Scan for additional .ini files in: (none)
Additional .ini files parsed:      (none)
```


Erros e Exceções

Diretivas no php.ini para manipulação de erros no PHP

- **erro_reporting** – quais erros são reportados e quais são ignorados
- **display_errors** – controla se o erro será mostrado como parte da saída do script
- **log_errors** – logará qualquer erro para um arquivo ou para o syslog definido na diretiva error_log.
- **error_log** – Nome do arquivo onde os erros de script devem ser registrados

Tratamento de exceções

- try-catch | try-catch-finally

```php
<?php

error_reporting(E_ALL);

$dividendo = 10;
$divisor = 0;

try {

    $divisao = $dividendo / $divisor;
    echo "$dividendo / $divisor = $divisao<br>" ;

} catch (\Throwable $t) {

    echo 'Erro capturado: ' . $t->getMessage();
}

echo 'Fim do programa.';

```

```php
<?php

error_reporting(E_ALL);

$dividendo = 10;
$divisor = 0;

try {
    // Trecho com possível erro
    $divisao = $dividendo / $divisor;
    echo "$dividendo / $divisor = $divisao<br>" ;
} catch (\Throwable $t) {
    // Captura do erro
    echo 'Erro capturado: ' . $t->getMessage();
} finally {
    // Trecho sempre executado (opcional)
    // Geralmente usado para fechar conexão com banco,
    // ou fechar arquivo
    echo 'mensagem qualquer<br>';
}

echo 'Fim do programa.';
```


Lançamento de exceções

- Throw (Throwable)
  - Exception
  - Error
  - LogicException
  - DomainException

- Exemplo:

```
if (!validaCPF($cpf) {
    throw new DomainException('CPF inválido');
}
```

Atividade

Calculadora WEB

- Transforme a calculadora da primeira atividade em uma Calculadora para web. Você deve ter as funções da calculadora separado do arquivo principal.