



INSTITUTO FEDERAL

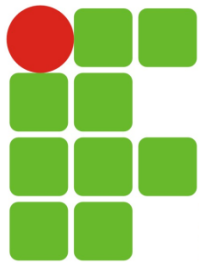
Farroupilha

Campus Avançado Uruguaiana

PHP – Revisão

Prof. Leandro Dallanora

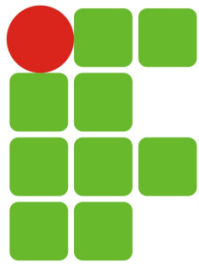
leandro.dallanora@iffarroupilha.edu.br



PHP - Revisão

- Roteiro
 - Sintaxe
 - Variáveis
 - Strings
 - Condição
 - Array
 - Repetição
 - Formulários
 - GET e POST





PHP - Sintaxe

- Sintaxe
- o código php é delimitado por

`<?php`

`//código`

`?>`

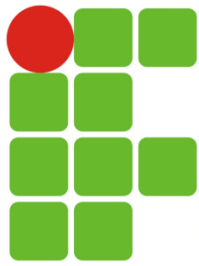
Ou na forma reduzida:

`<?`

`//código`

`?>`





PHP - Sintaxe

- Comentários:

`<?php`

`// de uma linha`

`/* Com mais`

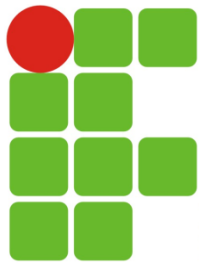
`de`

`uma`

`linha */`

`?>`





PHP - Variáveis

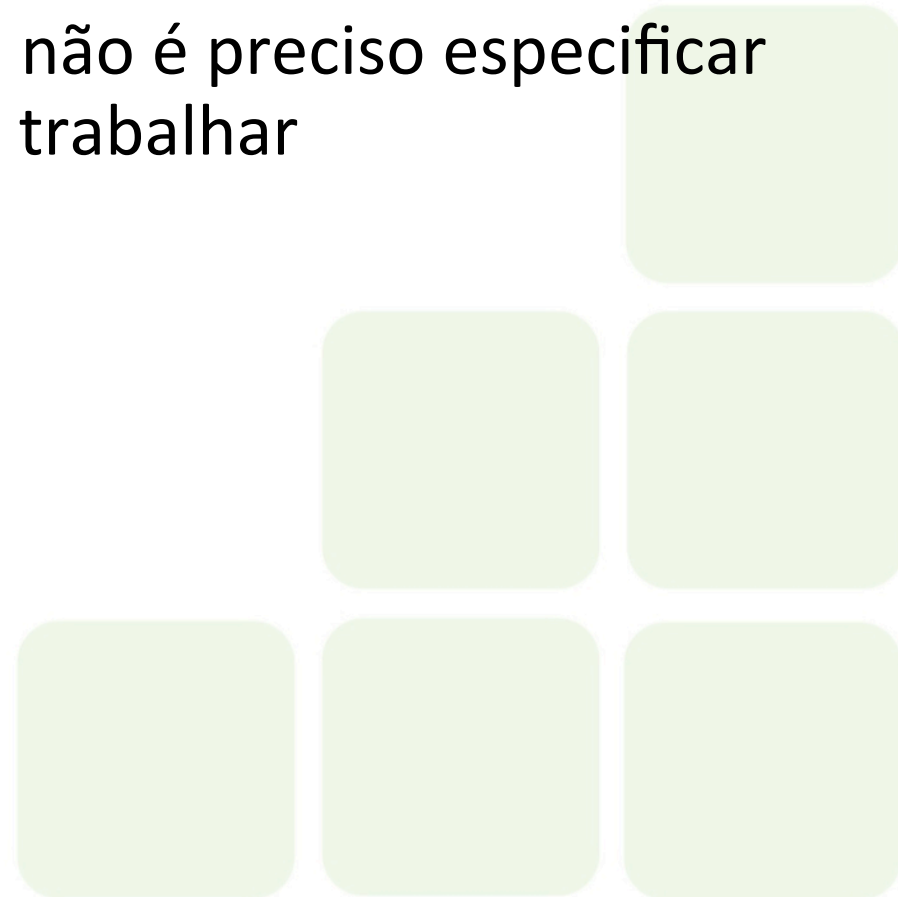
- as variáveis do php são especificadas por \$ antes do nome
- dispensam tipagem, ou seja, não é preciso especificar com qual tipo de dado irá se trabalhar
- devem iniciar por **letra** ou _

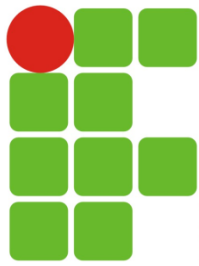
<?

```
$ze = "nome";
```

```
$nu = 6;
```

?>





PHP - Strings

- Há diferenças em se usar aspas simples e aspas duplas em Strings:

por exemplo

```
<? $x = "abc";  
    $y = "$x def";  
?>
```

o resultado será : abc def

já em

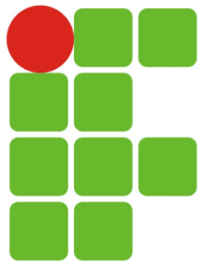
```
<? $x = "abc";  
    $y = '$x def';  
?>
```

o resultado será apenas : \$x def

- para concatenar uma string em php usa-se o (.) ponto

```
<? $z = $x . $y; ?>
```

o resultado será : abcdef

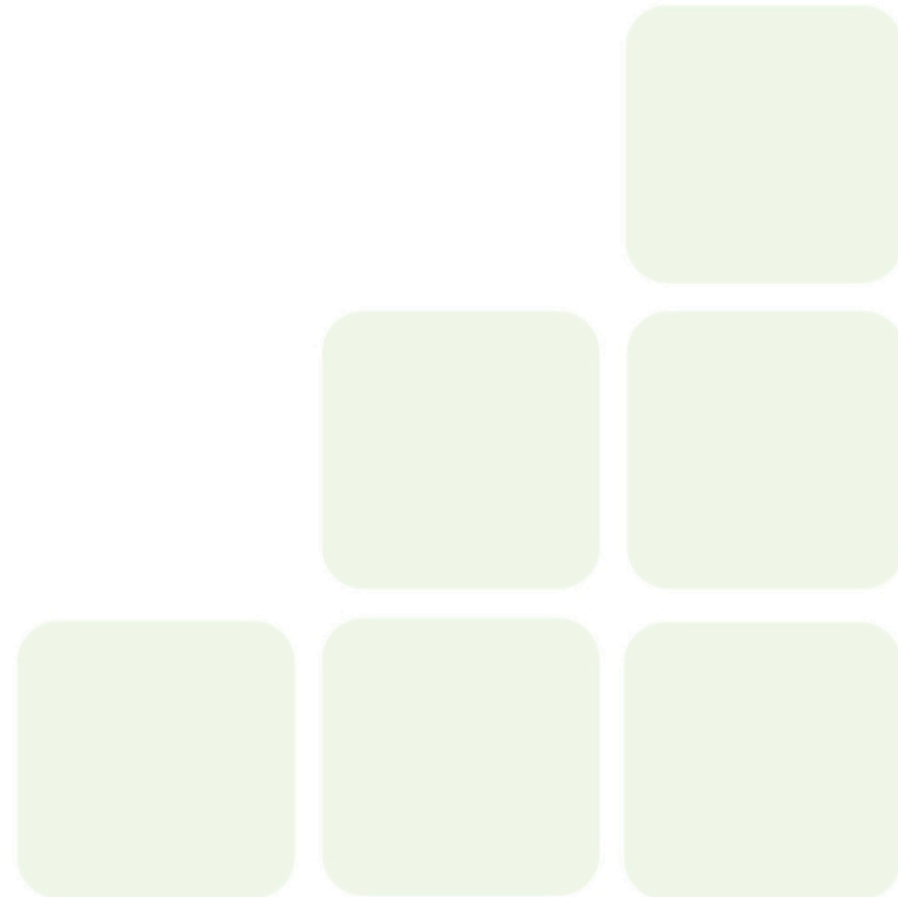


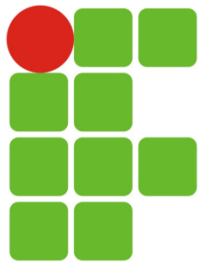
PHP - Condição

- if - sua estrutura em php é

```
if (condicao){  
    // codigo  
} else {  
    // codigo  
}
```

```
if ($a > $b) {  
    $maior = $a;  
} else {  
    $maior = $b;  
}
```

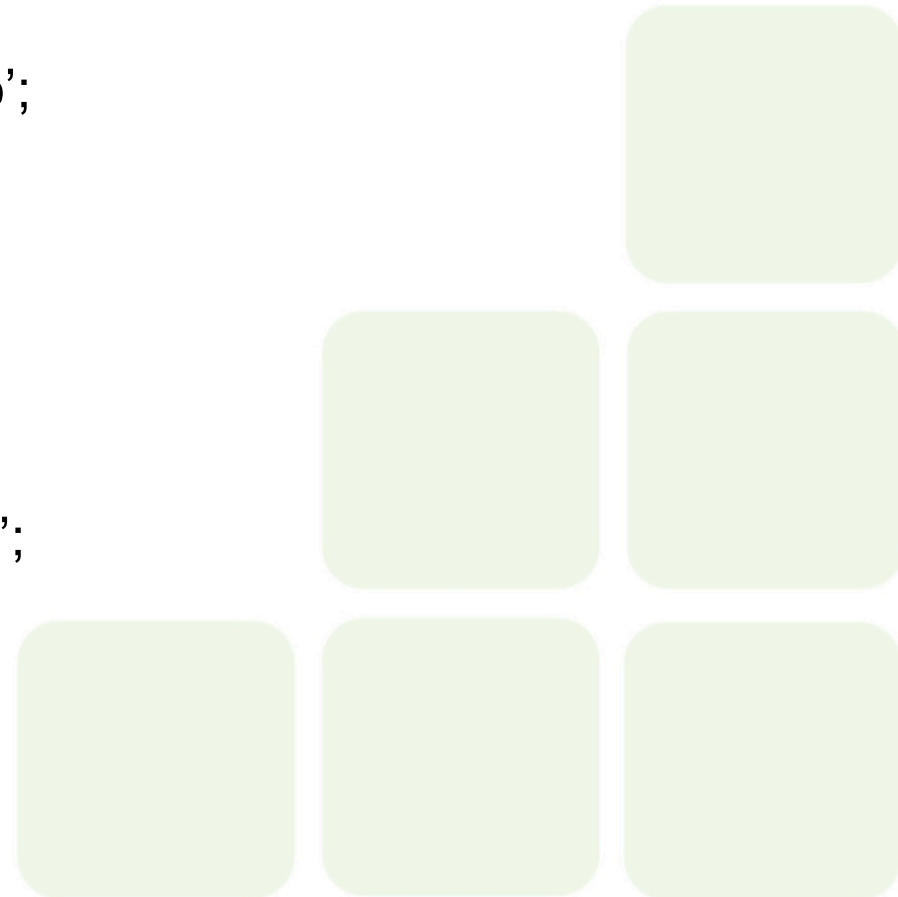


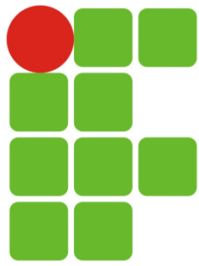


PHP - Condição

- switch - sua estrutura em php é

```
switch ($i) {  
    case 0:  
        echo 'i é igual a zero';  
        break;  
    case 1:  
        echo 'i é igual a um';  
        break;  
    case 2:  
        echo 'i é igual a dois';  
        break;  
    default:  
        echo 'Inválido!';  
}
```





PHP - Arrays

- permitem guardar os dados em vetores
- **Numéricos**
 - possuem seu índice como um valor numérico
 - podem ser criados de dois modos
- automaticamente

```
$nomes = array("Joao","Maria","Jose");
```

ou

```
$nomes[] = "Joao";
```

```
$nomes[] = "Maria";
```

```
$nomes[] = "Jose";
```

- Ou podem ser definidos manualmente

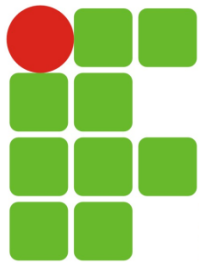
```
$nomes = array(5 => "Joao", 32 => "Maria", 26 => "Jose");
```

ou

```
$nomes[5] = "Joao";
```

```
$nomes[32] = "Maria";
```

```
$nomes[26] = "Jose";
```



PHP - Arrays

- **Associativos**
- possuem seu índice como um texto
- na criação precisam ter seu índice informado explicitamente

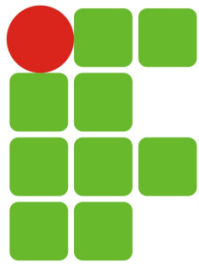
```
$idades = array('Joao' => 32, 'Maria' => 30, 'Jose' => 34);
```

Ou assim:

```
$idades["Joao"] = "32";
```

```
$idades["Maria"] = "30";
```

```
$idades['Jose'] = "34";
```

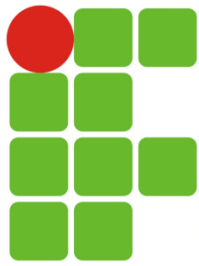


PHP - Repetição

- **while**
- verifica a condição antes de executar o bloco de códigos.

```
while (condicao){  
    // bloco de codigos  
}  
<?  
$i=1;  
while($i<=5) {  
    echo "O Numero é " . $i;  
    $i++;  
}  
?>
```

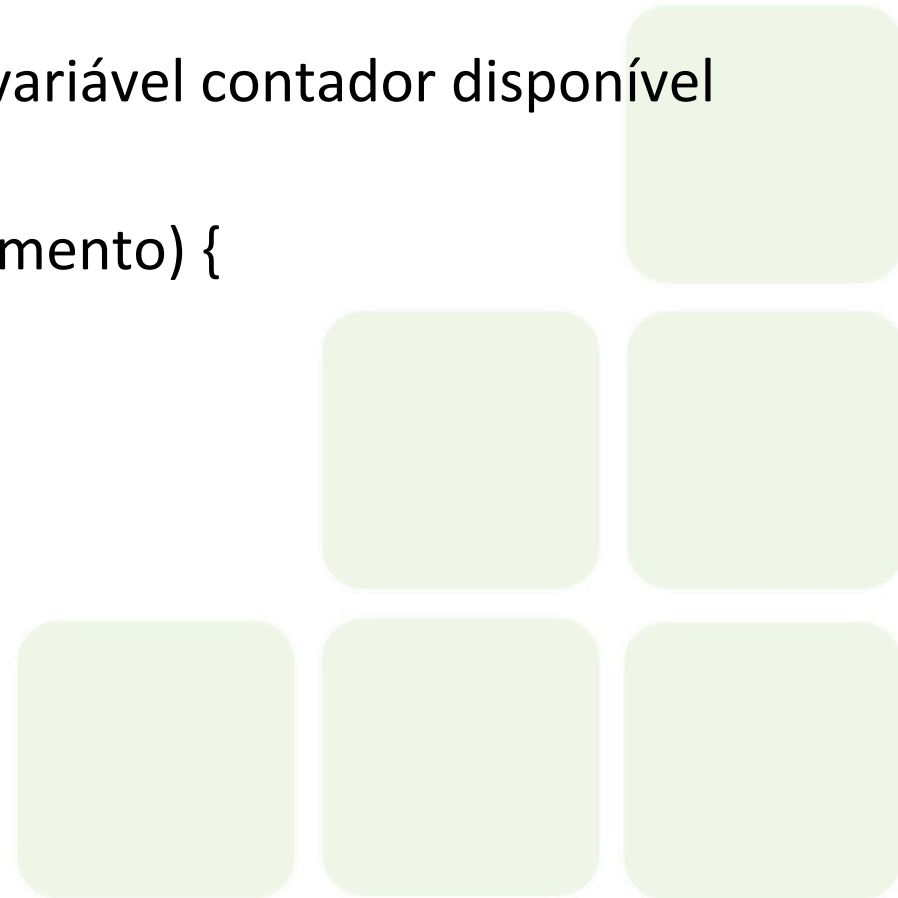


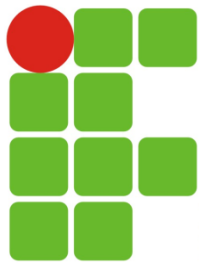


PHP - Repetição

- **for**
- é usado para situações quando se conhece a quantidade de vezes a ser repetido
- a cada interação haverá uma variável contador disponível

```
for (inicialização; condição; incremento) {  
    // bloco de comandos  
}  
<?  
for ($i=1; $i<=5; $i++) {  
    echo "o numero é " . $i;  
}  
?>
```





PHP - Repetição

- **foreach**
- usado em Arrays, percorre o mesmo desde sua primeira posição até a última, em cada interação haverá uma variável com o índice e outra com o valor atual

```
foreach (array as chave => valor) {
```

```
    //Bloco de comandos
```

```
}
```

```
<?
```

```
$arr = array("um", "dois", "três");
```

```
foreach ($arr as $k => $v) {
```

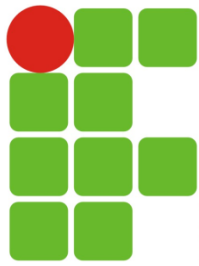
```
    echo "Posição: ". $k;
```

```
    echo " Valor: " . $v;
```

```
}
```

```
?>
```





PHP - Formulários

- executam a interação entre um usuário e o sistema, são por eles que os dados serão cadastrados

```
<form action="entra.php" method="post">
```

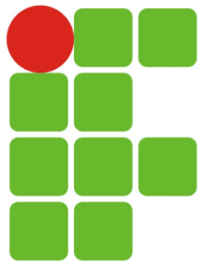
```
Nome: <input type="text" name="nome" />
```

```
senha: <input type="password" name="senha" />
```

```
<input type="submit" />
```

```
</form>
```

- o tipo de método de envio (method) irá definir como esses dados serão tratados no PHP
- se esse dado for enviado pelo endereço da página, irá ser tratado como method="get"



PHP - GET e POST

- os Arrays `$_GET` e `$_POST`, irão permitir que os dados enviados por um formulário sejam acessados por uma página em php
- nesses vetores cada um dos elementos serão um campo do formulário

Olá <?php **`$_POST["nome"]`**; ?>.

Sua senha é <?php echo **`$_POST["senha"]`**; ?>.