

# Desenvolvimento Web II

PHP Básico

# Funções pré-definidas

- Dentre as principais funções pré-definidas mais utilizadas, as que são utilizadas para manipulação de String são:
- **str-replace**

```
str_replace(  
    array|string $search,  
    array|string $replace,  
    string|array $subject,  
    int &$count = null  
): string|array
```

# Funções pré-definidas

- Dentre as principais funções pré-definidas mais utilizadas, as que são utilizadas para manipulação de String são:
- str-replace
- **substr**

```
<?php
$rest = substr("abcdef", -1);    // returns "f"
$rest = substr("abcdef", -2);    // returns "ef"
$rest = substr("abcdef", -3, 1); // returns "d"
?>
```

# Funções pré-definidas

- Dentre as principais funções pré-definidas mais utilizadas, as que são utilizadas para manipulação de String são:
- str-replace
- substr
- **strpos**

`strpos(string $myString, string $findMe,  
int $offset = 0): int | false`

# Funções pré-definidas

- Dentre as principais funções pré-definidas mais utilizadas, as que são utilizadas para manipulação de String são:
- str-replace
- substr
- strpos
- **strlen**

`strlen(string $string): int`

# Funções pré-definidas

- Dentre as principais funções pré-definidas mais utilizadas, as que são utilizadas para manipulação de String são:

- str-replace
- substr
- strpos
- strlen
- **trim**

```
trim(string $string, string $characters = " \n\r\t\v\x00"): string
```

- " " (ASCII 32 (0x20)), an ordinary space.
- "\t" (ASCII 9 (0x09)), a tab.
- "\n" (ASCII 10 (0x0A)), a new line (line feed).
- "\r" (ASCII 13 (0x0D)), a carriage return.
- "\0" (ASCII 0 (0x00)), the NUL-byte.
- "\v" (ASCII 11 (0x0B)), a vertical tab.

```
$hello  = "Hello World";  
$trimmed = trim($hello, 'HdWr');  
string(9) "ello Worl"
```

# Funções pré-definidas

- Dentre as principais funções pré-definidas mais utilizadas, as que são utilizadas para manipulação de String são:

- str-replace
- substr
- strpos
- strlen
- **trim**

```
trim(string $string, string $characters = " \n\r\t\v\x00"): string
```

- " " (ASCII 32 (0x20)), an ordinary space.
- "\t" (ASCII 9 (0x09)), a tab.
- "\n" (ASCII 10 (0x0A)), a new line (line feed).
- "\r" (ASCII 13 (0x0D)), a carriage return.
- "\0" (ASCII 0 (0x00)), the NUL-byte.
- "\v" (ASCII 11 (0x0B)), a vertical tab.

```
$text    = "\t\tThese are a few words :) ... ";  
$trimmed = trim($text, " \t.");  
string(24) "These are a few words :)"
```

# Funções pré-definidas

- Dentre as principais funções pré-definidas mais utilizadas, as que são utilizadas para manipulação de String são:
- str-replace
- substr
- strpos
- strlen
- **trim**
  - ltrim e rtrim
  - Mesmo funcionamento, porém atuando apenas do lado esquerdo da string (ltrim) e atuando apenas do lado direito da string (rtrim).



# Funções pré-definidas

- Dentre as principais funções pré-definidas mais utilizadas, as que são utilizadas para manipulação de String são:
- str-replace
- substr
- strpos
- strlen
- trim
- **str\_repeat**

```
str_repeat(string $string, int $times): string  
<?php  
echo str_repeat("-", 10);  
?>  
-----
```

# Funções pré-definidas

- Dentre as principais funções pré-definidas mais utilizadas, as que são utilizadas para manipulação de String são:
- str-replace
- substr
- strpos
- strlen
- trim
- str\_repeat
- **strtoupper e strtolower**

```
<?php
$str = "Mary Had A Little Lamb and She LOVED It So";
$str = strtoupper($str);
echo $str; // Prints MARY HAD A LITTLE LAMB AND SHE LOVED IT SO
?>

<?php
$str = "Mary Had A Little Lamb and She LOVED It So";
$str = strtolower($str);
echo $str; // Prints mary had a little lamb and she loved it so
?>
```

# Funções pré-definidas

- Dentre as principais funções pré-definidas mais utilizadas, as que são utilizadas para manipulação de String são:
- str-replace
- substr
- strpos
- strlen
- trim
- str\_repeat
- strtoupper e strtolower
- date

```
<?php
// Assuming today is March 10th, 2001, 5:16:18 pm, and that we are in the
// Mountain Standard Time (MST) Time Zone

$today = date("F j, Y, g:i a");           // March 10, 2001, 5:16 pm
$today = date("m.d.y");                   // 03.10.01
$today = date("j, n, Y");                 // 10, 3, 2001
$today = date("Ymd");                     // 20010310
$today = date('h-i-s, j-m-y, it is w Day'); // 05-16-18, 10-03-01, 1631 1618 6 Satpm01
$today = date('\i\t \i\s \t\h\e jS \d\a\y.'); // it is the 10th day.
$today = date("D M j G:i:s T Y");         // Sat Mar 10 17:16:18 MST 2001
$today = date('H:m:s \m \i\s\ \m\o\n\t\h'); // 17:03:18 m is month
$today = date("H:i:s");                   // 17:16:18
$today = date("Y-m-d H:i:s");             // 2001-03-10 17:16:18 (the MySQL DATETIME
format)
?>
```

# Funções pré-definidas

- Dentre as principais funções pré-definidas mais utilizadas, as que são utilizadas para manipulação de datas são:
- **mktime**

```
mktime(  
    int $hour,  
    ?int $minute = null,  
    ?int $second = null,  
    ?int $month = null,  
    ?int $day = null,  
    ?int $year = null  
): int | false
```

# Funções pré-definidas

- Dentre as principais funções pré-definidas mais utilizadas, as que são utilizadas para manipulação de datas são:
- mktime
- **date\_default\_timezone\_set** e **date\_default\_timezone\_get**

America/Rankin_Inlet	America/Recife	America/Regina	America/Resolute
America/Rio_Branco	America/Santarem	America/Santiago	America/Santo_Domingo
America/Sao_Paulo	America/Scoresbysund	America/Sitka	America/St_Barthelemy
America/St_Johns	America/St_Kitts	America/St_Lucia	America/St_Thomas
America/St_Vincent	America/Swift_Current	America/Tegucigalpa	America/Thule
America/Tijuana	America/Toronto	America/Tortola	America/Vancouver
America/Whitehorse	America/Winnipeg	America/Yakutat	

# Funções pré-definidas

- Dentre as principais funções pré-definidas mais utilizadas, as que são utilizadas para manipulação de datas são:
- mktime
- date\_default\_timezone\_set e date\_default\_timezone\_get
- **time**

```
<?php
echo 'Now: ' . time();
?>
Now: 1660338149
```

# Funções pré-definidas

- `isset($var)`: verifica se a variável `$var` já foi inicializada dentro do programa.
- `unset($var)`: remove a variável do programa.
- `getType($var)`: captura o tipo da variável `$var`. Existem outros métodos semelhantes que checam se uma variável é de um **determinado tipo**

## » `is_array`

`is_bool`

`is_callable`

`is_countable`

`is_double`

`is_float`

`is_int`

`is_integer`

`is_iterable`

`is_long`

`is_null`

`is_numeric`

`is_object`

`is_real`

`is_resource`

`is_scalar`

`is_string`

# Funções pré-definidas

- Para manipulação de arquivos:
- **file-exists**

```
<?php
$filename = '/path/to/foo.txt';

if (file_exists($filename)) {
    echo "The file $filename exists";
} else {
    echo "The file $filename does not exist";
}
?>
```



# Funções pré-definidas

- Para manipulação de arquivos:
- file-exists
- **fopen e fclose**

mode	Descrição
'r'	Abre somente para leitura; coloca o ponteiro do arquivo no começo do arquivo.
'r+'	Abre para leitura e escrita; coloca o ponteiro do arquivo no começo do arquivo.
'w'	Abre somente para escrita; coloca o ponteiro do arquivo no começo do arquivo e reduz o comprimento do arquivo para zero. Se o arquivo não existir, tenta criá-lo.
'w+'	Abre para leitura e escrita; fora isso, tem o mesmo comportamento de 'w'.
'a'	Abre somente para escrita; coloca o ponteiro do arquivo no final do arquivo. Se o arquivo não existir, tenta criá-lo. Neste modo, <a href="#">fseek()</a> não tem efeito, a escrita é sempre adicionada.
'a+'	Abre para leitura e escrita; coloca o ponteiro do arquivo no final do arquivo. Se o arquivo não existir, tenta criá-lo. Neste modo, <a href="#">fseek()</a> somente afeta a posição de leitura, escrita é sempre adicionada.

```
Cria uma variável do tipo stream
<?php
$handle = fopen("c:\\data\\info.txt",
"r");
?>
```

```
// Fecha o arquivo
$statusFechamento = fclose($fp);
if ($statusFechamento)
    echo "Arquivo fechado com sucesso";
```

# Funções pré-definidas

---

- Para manipulação de arquivos:
- file-exists
- fopen e fclose
- **fwrite e fread**

```
// Abre o arquivo para escrita e posiciona  
o cursor no final  
$fp = fopen($filename, 'a');  
// Escrever alguma coisa no arquivo.  
fwrite($fp, $somecontent);  
// Fecha o arquivo  
$statusFechamento = fclose($fp);  
if ($statusFechamento)  
    echo "Arquivo fechado com sucesso";
```

```
<?php  
// lê o conteúdo do arquivo para uma string  
$filename = "/usr/local/qualquer.txt";  
$handle = fopen ($filename, "r");  
$conteudo = fread ($handle, filesize ($filename));  
fclose ($handle);  
?>
```

Quantidade de caracteres que se deseja ler

# Arrays - Sintaxe

Um array pode ser criado com o construtor de linguagem `array()`. Ele leva qualquer quantidade de pares separados por vírgula `chave => valor` como argumentos.

```
array(  
    chave => valor,  
    chave2 => valor2,  
    chave3 => valor3,  
    ...  
)
```

```
<?php  
$array = array(  
    "foo" => "bar",  
    "bar" => "foo",  
);
```

```
$array = array("a", "b", "c");
```

Automaticamente o valor "a" é associado a chave 0, o valor "b" é associado a chave 1 e assim por diante.