

**PHP**

***php***

***php***

***php***

**JOSÉ FABRÍCIO FIGUEIREDO**

# BÁSICO

The PHP logo, consisting of the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

**Condições: if, else if, else, switch..case**

**Repetição: while, do while, for, foreach**

**Comparadores: ==, ===, <, >, <=, >=, !=, !==**

**Lógicos: &&, ||, AND, OR, XOR**

**Comando de saída output: echo, print, var\_dump, print\_r**

**Dados de formulários: \$\_GET, \$\_POST**

**Concatenar: "\$var e \$var"**

**function myfunction (string \$var):string {}**

**Tipos: (int, integer), (real, float, double), (string)**

**\$GLOBALS[""] -> Variável global.**

**Terminal:**

**php -S localhost:8000**

The PHP logo, consisting of the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

**JOSÉ FABRÍCIO FIGUEIREDO**

# ARRAYS



```
$n = array(1, 2, 3, 4, 5)
```

```
$n[] = 7
```

```
$v = array(1=>"A", 3=>"B", 6=>"C")
```

```
$n = array(array(2, 3), array(3, 4), array(9, 5))
```

## Comandos:

- **count()**
- **array\_push(\$n, 7)**
- **array\_pop()**
- **array\_unshift()**
- **array\_shift()**
- **sort()**
- **rsort()**
- **asort()**
- **arsort()**
- **krsort()**
- **ksort()**



**JOSÉ FABRÍCIO FIGUEIREDO**

# STRING

The PHP logo, featuring the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

- **strlen()**
- **printf("O %s custa R\$ %.2f", \$p, \$pr)**
- **print\_r()** -> Array ([0] => valor [1] => valor)
- **wordwrap("texto", cadaCaracter, "<br>")**
- **trim()** -> Exclui espaços
- **ltrim()**
- **rtrim()**
- **str\_word\_count(\$frase, inicio)**
- **explode()**
- **str\_split()**
- **implode()**
- **chr()**
- **ord(\$letra)**
- **strtolower()**
- **strtoupper()**
- **ucfirst()**
- **ucwords()**
- **strrev()**
- **strpos()**
- **stripos()**
- **substr\_count()**
- **substr()**
- **str\_pad()**
- **str\_repeat()**
- **str\_replace()**

The PHP logo, featuring the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

**JOSÉ FABRÍCIO FIGUEIREDO**

# DATA E HORA / EXCEÇÕES

The PHP logo, consisting of the lowercase letters 'php' in a bold, black, sans-serif font, with a white outline, set against a dark blue oval background.

## Data e Hora:

**date("d")** -> [d, m, Y, H, h, i, s]

## Exceções:

**throw new Exception("Essa é uma exceção");**

```
try {  
} catch (Exception $erro) {  
    echo $erro->getMessage();  
} finally {}
```

The PHP logo, consisting of the lowercase letters 'php' in a bold, black, sans-serif font, with a white outline, set against a dark blue oval background.

**JOSÉ FABRÍCIO FIGUEIREDO**



# MANIPULAÇÃO DE ARQUIVOS

The PHP logo, consisting of the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

**r -> Leitura, ponteiro no início.**

**w -> Escrita, ponteiro no início e zera, se arquivo não existir ele o cria.**

**a -> Escrita, ponteiro no final, se arquivo não existir ele o cria.**

**x -> Escrita, ponteiro no início, se arquivo já existir ele gera um erro do tipo E\_WARNING.**

**[r+ | w+ | a+ | x+] -> Leitura e escrita.**

**\$arquivo = fopen("aula.txt", "w");**

**\$conteudo = fread(\$arquivo, 100);**

**fwrite(\$arquivo, "\r\n");**

**feof(\$arquivo); -> Chegou no final?**

**fgets(\$arquivo); -> Retorna a linha.**

**fclose(\$arquivo);**

The PHP logo, consisting of the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

**JOSÉ FABRÍCIO FIGUEIREDO**

# FUNÇÕES



**php**

## Seção:

**session\_start()**

**\$\_SESSION['id']**

**session\_destroy()**

## Variável:

**unset()** -> Destrói a variável.

**isset()** -> A variável existe?

**empty()** -> A variável não está vazia?

## Importar arquivo:

**include()** -> Importar arquivo.

**require()** -> Importa o arquivo, caso não exista o código para aqui.

**[include\_once() | require\_once()]** -> Se o arquivo existe, não adicionará novamente.



**php**

**JOSÉ FABRÍCIO FIGUEIREDO**

# FUNÇÕES

The PHP logo, consisting of the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

**header()**

**rand(int \$min, int \$max): int ->** Gera um inteiro aleatório.

**Cookie:**

**setcookie(name, value, expire)**

expire -> time()+1000

**\$\_COOKIE['name']**

**PHP no <style>:**

**<style>**

p {font-size:<?php echo \$tam; ?>}

**</style>**

The PHP logo, consisting of the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

**JOSÉ FABRÍCIO FIGUEIREDO**



# CLASS

The PHP logo, featuring the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

```
class MyClass {  
(const, public, private, protected, + (static))  
    $res;  
  
    public function __construct(){  
    public function __clone(){  
  
(public, private, protected, + (static))  
    function myFuncao() {  
        echo $this->res;  
        echo self::$res; //[static | const]  
    }  
}  
  
$resClass = new MyClass();  
$resClass->res = "admin";  
$resClass->myFuncao();
```

The PHP logo, featuring the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

JOSÉ FABRÍCIO FIGUEIREDO

# CLASS

The PHP logo, consisting of the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

**// Get & Set**

```
class MyClass {  
    public function getName() {  
        return $this->res;  
    }  
  
    public function setName($res) {  
        $this->res = $res;  
    }  
}
```

**// Clonar class**

```
$resClass2 = clone $resClass;
```

**// Extends**

```
class MyClass2 extends MyClass {}
```

The PHP logo, consisting of the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

**JOSÉ FABRÍCIO FIGUEIREDO**

# CLASS

The PHP logo, consisting of the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

**// Método static -> Não precisa instanciar a class.**

```
MyClass::$res = "admin";
```

```
MyClass::myFuncao();
```

**abstract class MyClass{}; -> Não pode instanciar objetos de classes abstract. Função pode ser abstract mas a class deve ser implementado.**

**// Interface**

```
interface MyClass3 {
```

```
    public function myFunction();
```

```
}
```

```
class MyClass4 implements MyClass3{
```

```
    public function myFunction(){// lógica}
```

```
}
```

The PHP logo, consisting of the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

**JOSÉ FABRÍCIO FIGUEIREDO**

# CLASS

The PHP logo, consisting of the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

**// Namespace**

**namespace nome;** -> Apelidar o arquivo.

**require 'class.php';**

**\$res = new \nome\MyClass();**

**use nome\MyClass as myclass;**

**\$res = new myclass();**

The PHP logo, consisting of the letters 'php' in a stylized, lowercase font with a white outline, set against a dark blue oval background.

**JOSÉ FABRÍCIO FIGUEIREDO**

# VALIDAÇÃO *php*

**FILTER\_VALIDATE\_INT**

**FILTER\_VALIDATE\_EMAIL**

**FILTER\_VALIDATE\_FLOAT**

**FILTER\_VALIDATE\_IP**

**FILTER\_VALIDATE\_URL**

**filter\_input(INPUT\_POST, 'name\input',  
FILTER\_VALIDATE\_-)**



**JOSÉ FABRÍCIO FIGUEIREDO**



# UPLOAD DE ARQUIVO



```
<?php
    if (isset($_POST["enviar-formulario"])){
        $extensao = pathinfo(
            $_FILES['file']['name'],
            PATHINFO_EXTENSION);
        $nome = uniqid().".$extensao";
        move_uploaded_file(
            $_FILES['file']['tmp_name'],
            'arquivos/'.$nome);
    }
?>

<form action="" method="post"
    enctype="multipart/form-data">
    <input type="file" name="file">
    <input type="submit" name="enviar-
formulario">
</form>
```



JOSÉ FABRÍCIO FIGUEIREDO

## Conceção:

```
$con = mysqli_connect("localhost", "root",  
"senha");
```

```
mysqli_select_db($con, "banco");
```

```
$res = mysqli_query($con, sql);
```

```
mysqli_num_rows($res); -> Quantidade de  
linhas.
```

```
mysqli_affected_rows($con); -> Linhas  
afetadas.
```

```
mysqli_fetch_row($res); -> Coletar dados da  
linha.
```

```
mysqli_close($con); -> Fechar banco de  
dados.
```