

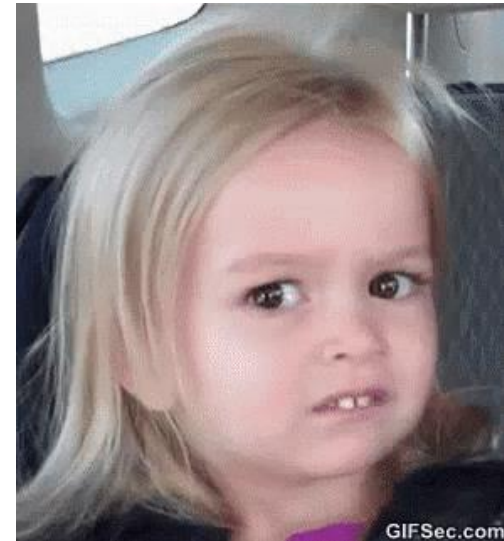


Arquivos e Hashing

FULLSTACK - CAMPUS VILA OLÍMPIA
TURMA 03 - JULHO/2018

Aula passada...

- JSON
- json_encode
- json_decode
- Manipulação de Arquivos
- fopen, fwrite, fread, fclose
- file_get_contents
- file_put_contents



JSON (JavaScript Object Notation)

é um formato de texto leve para a troca de dados,
baseado em objeto de JavaScript.



```
{
  "alunos" : [
    { "nome" : "Flavio", "sobrenome" : "Cordeiro" },
    { "nome" : "Larissa", "sobrenome" : "Novaes"},
    { "nome" : "Ricardo", "sobrenome" : "Zini" }
  ]
}
```

json_encode

Retorna a representação JSON do valor dado

```
$carro = [  
    "Marca" => "Ford",  
    "Cor" => "Preto"  
];
```

```
json_encode($carro);
```

Retorno do json_encode

```
{  
    "Marca": "Ford",  
    "Cor": "Preto"  
}
```

json_decode

Decodifica uma string de JSON

```
$carro = '{  
    "Marca": "Ford",  
    "Cor": "Preto"  
';
```

```
json_decode($json, true);
```

Retorna um array associativo

```
array(2) {  
    [Marca] => "Ford"  
    [Cor] => "Preto"  
}
```

Beleza, já sei pegar informações inseridas pelo usuário.

Mas como eu faço pra guardar essas informações?



Gerenciamento de arquivos

O que é podemos fazer?

- ✓ Criar arquivo
- ✓ Editar arquivo
- ✓ Ler arquivo

- Caminho do arquivo
- Recurso:
variável especial, que mantém uma referência a um recurso externo.
- Conteúdo do arquivo
- E outras informações.

Criar/Editar arquivo

```
fopen($filename, $mode);
```

```
fwrite($recurso, $texto);
```

```
$caminhoDoArquivo = 'hobbies.txt';  
$recurso = fopen($caminhoDoArquivo, 'a');  
fwrite($recurso, "Leitura \n");  
fwrite($recurso, "Corrida \n");  
fclose($recurso);
```

O conteúdo de 'hobbies.txt' agora é:

Leitura

Corrida

Ler conteúdo do arquivo

```
fread(resource $recurso, int $length)
```

```
$caminhoDoArquivo = "hobbies.txt";  
$recurso = fopen($caminhoDoArquivo, "r");  
$tamanhoDoArquivo = filesize($caminhoDoArquivo);  
$conteudo = fread($recurso, $tamanhoDoArquivo);  
fclose($recurso);
```

```
echo $conteudo; //retorna "Leitura Corrida";
```

fopen

Abre um arquivo ou uma URL e **retorna um recurso** (resource).

```
fopen(string $filename, string $mode);
```

Alguns modos:

- **r**: Abre **somente para leitura**; 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.
- **a**: Abre **somente para escrita**; coloca o ponteiro do arquivo no final do arquivo. Se o arquivo não existir, tenta criá-lo.
- **x**: Cria e abre o arquivo **somente para escrita**; coloca o ponteiro no começo do arquivo. Se o arquivo já existir, a chamada a fopen() falhará, retornando FALSE e gerando um erro de nível E_WARNING. Se o arquivo não existir, tenta criá-lo.

fopen

Abre um arquivo ou uma URL e **retorna um recurso** (resource).

```
fopen(string $filename, string $mode);
```

Alguns modos:

- **r+:** Abre para **leitura e escrita**; coloca o ponteiro do arquivo no começo do arquivo.
- **w+:** Abre para **leitura e 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.
- **a+:** Abre para **leitura e escrita**; coloca o ponteiro do arquivo no final do arquivo. Se o arquivo não existir, tenta criá-lo.
- **x+:** Cria e abre o arquivo para **leitura e escrita**; coloca o ponteiro no começo do arquivo. Se o arquivo já existir, a chamada a fopen() falhará, retornando FALSE e gerando um erro de nível E_WARNING. Se o arquivo não existir, tenta criá-lo.

file_get_contents

É uma espécie de atalho para usar
fopen -> fread -> fclose

```
$caminhoDoArquivo = "hobbies.txt";  
$hobbies = file_get_contents($caminhoDoArquivo);
```

```
echo $hobbies; //retorna "Leitura Corrida";
```

file_put_contents

É uma espécie de atalho para usar
fopen -> fwrite -> fclose

```
$caminhoDoArquivo = "hobbies.txt";
```

```
$hobbies = "Leitura \n Corrida \n";
```

```
file_put_contents($caminhoDoArquivo, $hobbies);
```

fgets

Retorna uma linha de um arquivo aberto

```
$recurso = fopen("hobbies.txt", "r");  
if ($recurso) {  
    while (($linha = fgets($recurso)) !== false) {  
        echo "$linha <br/>";  
    }  
}  
fclose($recurso);
```

file_exists

Verifica se um arquivo existe

```
$arquivo = hobbies.txt';  
if (file_exists($arquivo)) {  
    echo "O arquivo $arquivo existe";  
} else {  
    echo "O arquivo $arquivo NÃO existe";  
}
```

Vamos praticar!



Arquivos e JSON

DAILY SCRUM

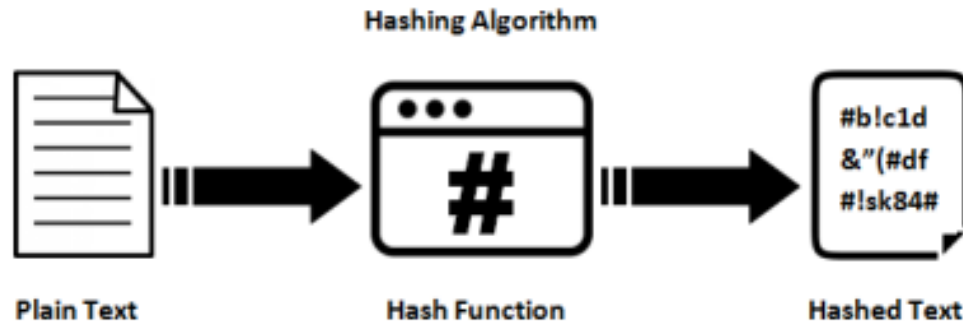
maximo 15min

- o que fiz ontem?
- o que estou fazendo?
- tem algum impedimento?

dica: use um cronometro!

HASHING

Uma função de hash é uma função **unilateral** que pega qualquer texto e transforma em um código criptografado que não pode voltar atrás



HASHING

Existem diferentes algoritmos de hashing, e o PHP permite utilizá-los tanto para criptografar quanto para verificar as informações

Ele é muito usado para a implementação de senhas para poder armazenar de forma segura.

password_hash

Cria um hash de um texto
(normalmente uma senha)

```
password_hash(string $password, int $algoritmo);
```

password_hash

```
<?php
    $hash = password_hash("Olá", PASSWORD_DEFAULT);
?>
```

password_verify

```
<?php
```

```
$hash1 = password_hash("Olá", PASSWORD_DEFAULT);  
$hash2 = password_hash("Tchau", PASSWORD_DEFAULT);
```

```
password_verify("Olá", $hash1); //true  
password_verify("Oi", $hash2); //false
```

```
?>
```

Vamos praticar!



Exercícios

OBRIGADO!

