

## Soubory - úvod

Se soubory se setkal asi každý, kdo si kdy sedl k počítači. Zabudované funkce v PHP je možné rozdělit do dvou částí: Na první část, které pracuje se soubory pomocí tzv. **handlů** a na druhou část, která **handly** nepotřebuje. V tomto díle se budu zabývat pouze přístupem pomocí **handlů**.

### Přístup pomocí **handlů** - otevření a zavření souboru

Základní přístup při práci se soubory je přístup pomocí **handlů**. Pokud potřebujeme pracovat s nějakým souborem, můžeme si to představit jako práci se šanonem plným papírů. Nejdříve vezmu šanon, otevřu ho, listuji v něm, čtu si, případně přidám další dokument a nakonec šanon zavřu a uložím do poličky.

Představa práce se šanonem je analogická k tomu, jak se pracuje se souborem pomocí **handlů**.

Nejdříve se soubor otevře, pracuje se s ním, čtou se, případně zapisují data do souboru a nakonec se zavře. Protože PHP dovoluje mít otevřeno naráz více souborů, má každý otevřený soubor tzv. **handle**, aby se vědělo, se kterým souborem se právě bude pracovat.

A teď konkrétně, pro otevření souboru existuje funkce **fopen**, které můžeme předat 2, nebo 3 parametry. Pro zavření souboru, pak mám funkci **fclose**, která má 1 parametr. Nejjednodušší příklad, který mohu udělat je soubor otevřít, a hned zavřít. I takto jednoduché použití může mít svůj význam, jak je ukázáno v následujícím příkladu:

```
<html>
<head>
<title>Soubory</title>
</head>
<body>
<?php
    $handle = fopen("a.txt", "w+");
    fclose($handle);
?>
</body>
</html>
```

Pokud si příklad spustíte, zdánlivě to vůbec nic nedělá. Nic nevypíše, ale jednu věc přesto udělá. Založí nový prázdný soubor s názvem a.txt, který se Vám objeví ve stejném adresáři, jako máte skript. Tedy výsledkem je založení nového souboru. Funkce **fopen** totiž při otevírání může volitelně založit i nový soubor, pokud ještě neexistuje.

Zkusme si vzít na mušku nejdříve funkci **fopen**, která má tento tvar při použití dvou parametrů:

```
fopen (jméno_souboru, způsob_otevření);
```

Jako první parametr se zadává jméno souboru se kterým chceme pracovat. V našem příkladě to bylo a.txt, ale použít můžeme samozřejmě cokoli jiného. Pokud to způsob otevření podporuje, soubor s takovým jménem nemusí existovat, a funkce **fopen** ho pak založí.

Jako druhý parametr se pak zadává způsob otevření souboru. V zásadě jsou tři základní způsoby otevření:

<b>r</b>	ze souboru budeme chtít číst (soubor musí existovat)
<b>w</b>	do souboru budeme chtít zapisovat (pokud existuje soubor, smaže ho, vždy vytvoří nový soubor)
<b>a</b>	budeme chtít připisovat na konec souboru nová data (pokud soubor neexistuje, vytvoří ho)

Kromě toho můžeme za písmenko přidat ještě znak +, aby se umožnilo to druhé, tedy:

<b>r+</b>	budeme chtít číst i zapisovat (soubor musí existovat)
<b>w+</b>	budeme chtít zapisovat i číst (pokud existuje soubor, smaže ho, vždy vytvoří nový soubor)
<b>a+</b>	budeme chtít připisovat na konec souboru nová data i číst (pokud soubor neexistuje, vytvoří ho)

To jsou tedy parametry funkce **fopen**. Funkce **fopen** vrací tzv. **handle**, což je hodnota, kterou uložíme do proměnné a používáme jí při dalších akcích se souborem. Stejně jako v příkladu. Výsledný handle jde přímo použít i jako podmínka, abychom otestovali, zda všechno dobře dopadlo:

```
<html>
<head>
<title>Soubory</title>
</head>
<body>
<?php
    $handle = fopen("a.txt", "w+");
    if ($handle)
        echo "Otevření souboru se povedlo.";
    else
        echo "Otevření souboru selhalo.";
?>
</body>
</html>
```

Výše uvedený příklad obsahuje otevření souboru i s kontrolou, zda všechno proběhlo v pořádku. Protože otevření souboru se nemusí podařit, měli byste vždy udělat tuto kontrolu. Soubor, který požadujete nemusí existovat, nebo k němu nemáte práva a spousta jiných dalších příčin může způsobit selhání.

Pokud někdy soubor otevřete, mělo by někdy dojít k zavření souboru. Pokud to neuděláte v PHP skriptu, tak na konci skriptu PHP samo všechny soubory zavře. A nebo to můžete udělat vy sami funkcí **fclose**, která má takový tvar:

```
fclose (handle);
```

Jako parametr předáte handle získaný pomocí funkce **fopen**. Po zavolání funkce **fclose** je handle dále už neplatný a nedá se použít pro práci se souborem. Tady je tedy celý příklad s otevřením i zavřením souboru a kontrolu selhání:

```
<html>
<head>
<title>Soubory</title>
</head>
<body>
<?php
    $handle = fopen("a.txt", "w+");
    if ($handle)
        echo "Otevření souboru se povedlo.";
    else
        die("Otevření souboru selhalo.");

    fclose($handle);
?>
</body>
</html>
```

## Zápis do souboru

Pokud jste to vydrželi až sem, vážení čtenáři, ukážu vám, jak zapisovat do souboru. Nejlépe přímo na příkladu. Následující příklad připiše do souboru slovo "Haló":

```
<html>
<head>
<title>Soubory</title>
</head>
<body>
<?php
    $handle = fopen("a.txt", "a");
    if ($handle)
        echo "Otevření souboru se povedlo.";
    else
        die("Otevření souboru selhalo.");

    fwrite($handle, "Haló");
    fclose($handle);
?>
</body>
</html>
```

Příklad obsahuje už známé části, a kromě toho obsahuje funkci **fwrite**, která slouží pro zápis do souboru. Má dva parametry, jako první se jí musí předat handle a jako druhý se jí předá, co se má zapsat do souboru.

Protože soubor je otevřen tak, že druhý parametr funkce fopen je "a", tak se veškeré zápisy do souboru připisují na konec. Pokud tedy spustíte tento příklad vícekrát, slovo "Haló" se napíše na konec několikrát, což si můžete vyzkoušet.

Pro vyzkoušení příkladu se pak musíte podívat do souboru s názvem a.txt, jestli obsahuje slovo "Haló".

Funkce **fwrite** má následující tvar:

```
fwrite (handle, co_se_má_zapsat, maximální_délka_zápisu);
```

Funkce fwrite může mít dva, nebo tři parametry. Pokud použijeme poslední třetí parametr, je to číslo, které říká, kolik znaků se má maximálně zapsat. Funkce fwrite si pak dá pozor na to, aby nezapsala více. Následující příklad je stejný jako předchozí, ale říká, že se mají zapsat maximálně dva znaky z "Haló", zapíše se tedy jenom "Ha":

```
<html>
<head>
<title>Soubory</title>
</head>
<body>
<?php
    $handle = fopen("a.txt", "a");
    if ($handle)
        echo "Otevření souboru se povedlo.";
    else
        die("Otevření souboru selhalo.");

    fwrite($handle, "Haló", 2);

    fclose($handle);
?>
</body>
</html>
```

## Čtení ze souboru

Čtení ze souboru se provádí pomocí funkce **fread**. Zkusme hned přejít na příklad:

```
<html>
<head>
<title>Soubory</title>
</head>
<body>
<?php
    $handle = fopen("a.txt", "r");
    if ($handle)
        echo "Otevření souboru se povedlo.<br>";
    else
        die("Otevření souboru selhalo.");

    $načteno = fread($handle, 1000);

    echo "Přečetl jsem : ", $načteno;

    fclose($handle);
?>
</body>
</html>
```

Příklad přečte, co jsme až dosud do souboru a.txt v předchozích příkladech zapsali. Samotné čtení se provádí funkcí fread, která má dva parametry. Prvním parametrem je už obligátní handle a druhým parametrem je kolik znaků máme maximálně načíst. V našem příkladě načteme maximálně 1000 znaků.

## Pozice v souboru

Každý otevřený soubor má přiřazeno myšlené ukazovátko, které ukazuje z jakého místa v souboru se provede příští zápis, nebo čtení. Po otevření souboru se bude číst ze začátku souboru, nebo zapisovat na začátek souboru (pokud není použit způsob otevření "a"), a ukazovátko pozice je tedy na začátku souboru.

Pozice ukazovátko se vyjadřuje ve znacích, které jakoby musíme přeskočit, abychom se k současné pozici dostali. Je-li tedy ukazovátko pozice na začátku souboru, pozice je nula. Pokud přečtu 4 znaky, posune se ukazovátko na pozici 4, a další čtení, nebo zápis bude pokračovat z této pozice.

Jednodušší je asi pochopení na příkladu. Použiji funkci **ftell**, která mi vrátí pozici ukazovátko:

```
<html>
<head>
<title>Soubory</title>
</head>
<body>
<?php
    $handle = fopen("a.txt", "r");
    if ($handle)
        echo "Otevření souboru se povedlo.<br>";
    else
        die("Otevření souboru selhalo.");

    $pozice = ftell($handle);
    echo "Pozice ukazovátko je ", $pozice, "<br>";

    $načteno = fread($handle, 4);
    echo "Přečetl jsem : ", $načteno, "<br>";

    $pozice = ftell($handle);
    echo "Pozice ukazovátko je ", $pozice, "<br>";

    fclose($handle);
?>
</body>
</html>
```

V příkladu je po otevření souboru použita funkce **ftell** ke zjištění, na jaké pozici je ukazovátko v souboru. Mělo by se vypsát, že je na nulté pozici, protože je na začátku souboru. Poté přečtu ze souboru 4 znaky a znovu se zeptám na pozici. Měl bych dostat číslo 4.

Samo o sobě je pozice v souboru zajímavá i proto, že je možné pozici ovlivňovat. Například funkce **rewind** vrátí pozici znovu na začátek souboru, takže je možné číst, nebo zapisovat soubor znovu od začátku. Tuto funkci použiji ve svém počítačle stránek.

## Počítadlo stránek

Počítadlo stránek bude využívat soubor s názvem kolik.bin, ve kterém si bude pamatovat, kolikrát byla naše stránka už návštěvníky prohlédnuta. Po každém prohlédnutí stránky se zvětší číslo v souboru kolik.bin o jedničku:

```
<html>
<head>
<title>Soubory</title>
</head>
<body>
<?php
$handle = fopen("kolik.bin","a");
fclose($handle);

$handle = fopen("kolik.bin","r+");
$kolik = fread($handle,10);
++$kolik;
rewind($handle);
fwrite($handle,$kolik);
fclose($handle);

echo "Počet zobrazení stránky: ", $kolik;
?>
</body>
</html>
```

Pokud si toto počítadlo vyzkoušíte, zjistíte, že pokaždé, když znovu spustíte počítadlo, přiskakují vám tam počty zobrazení stránek. A jak to funguje?

První 2 řádky v PHP mají za úkol jenom vytvořit soubor kolik.php, pokud by čistě náhodou neexistoval:

```
$handle = fopen("kolik.bin","a");
fclose($handle);
```

Způsob otevření "a" je vhodný pro takovou akci. Pokud soubor s názvem kolik.php ještě neexistuje, pak jej způsob otevření "a" nechá beze změny. Pokud ještě neexistuje, pak jej způsob otevření "a" vytvoří. Protože ale použitý způsob otevření nám nevyhovuje pro další práci, soubor hned poté zavřeme a otevřeme ho jinak. Důležité pro nás je, že po provedení prvních dvou řádků PHP kódu zaručeně existuje soubor s názvem kolik.bin.

Na dalším řádku otevíráme soubor pro čtení i zápis. Použijeme proto způsob otevření "r+":

```
$handle = fopen("kolik.bin","r+");
```

O řádek dál přečtu prvních 10 znaků ze souboru. To pro mě bude znamenat uložené číslo, kolikrát se moje stránka už zobrazila:

```
$kolik = fread($handle,10);
```

Toto číslo musím zvětšit o jedničku, protože teď se zobrazuje právě tato stránka znovu:

```
++$kolik;
```

Další řádky pak slouží k tomu, aby tohle nové číslo bylo zapsáno zpět do souboru kolik.bin. Funkcí rewind si posunu ukazovátka pozice zpátky na začátek, abych číslo zapsal zpět na začátek souboru:

```
rewind($handle);
fwrite($handle,$kolik);
```

Pak následuje už jen zavření souboru a pak vypsání počtu zobrazení.