

PHP

Soubory, pole, funkce

WEB 2013-2014

Otevření souboru (1)

- Otevření souboru:
 - `$fw = fopen("cesta/soubor", mód);`
- Root webserveru:
 - `$_SERVER['DOCUMENT_ROOT'];`
- Módy otevření souboru:
 - `r` – read
 - `r+` – read + write
 - `w` – write
 - `w+` – write +read
 - `a` – append
 - `a+` – append and read
 - `b` – binární mód
- Pozor na `safemod` a přístupová práva
- Při neúspěchu vráceno `false`

Otevření souboru (2)

```
@ $fw = fopen("cesta/soubor", r);  
if (!$fw) {  
    echo "Nelze otevřít soubor...";  
    exit;  
}
```

Zápis do souboru, zavření souboru

□ Otevření souboru pro zápis

```
@ $fw = fopen("cesta/soubor", w);
```

□ Zápis do souboru

```
■ fwrite($fw, $retezec);
```

```
■ fputs($fw, $retezec);
```

□ Zavření souboru

```
■ fclose($fw);
```

Čtení ze souboru

❑ Otevření souboru pro čtení

```
@ $fw = fopen("cesta/soubor", r);
```

❑ Test konce souboru

- `feof($fr);`

❑ Čtení řádky:

- `fgets($fw, 999)` – max 998 bytů

- `fgetss` – podobné, vyhazuje HTML/PHP tagy

- `fgetcsv` – třetí parametr jako oddělovač

❑ Čtení celého souboru

- `readfile(soubor), $filearray = file($fr)`

❑ Čtení znaků

- `fgetc($fr)`

❑ Čtení bytů

- `fread($fr, délka);`

Další užitečné funkce

- Test existence souboru
 - `file_exists(soubor)`
- Zjištění velikosti souboru
 - `filesize(soubor)`
- Vymazání souboru
 - `unlink(soubor)`
- Navigace v souboru
 - `rewind($fw)` – na začátek souboru
 - `ftell($fw)` – zjištění pozice
 - `fseek($fw, offset, start)` – nastavení pozice v souboru

Klasická pole

- ❑ `$produkty = array('pneu', 'výfuk', 'sklo');`
- ❑ `$produkty[1]; // výfuk`
- ❑ `$cisla = range(1,10); // pole čísel 1-10`
- ❑ **Změna prvku pole:** `$produkty[1] = 'karosérie'`
- ❑ **Lze přidat nový prvek:** `$produkty[3] = 'karosérie'`
- ❑ **Procházení pole:** `for, foreach`

Asociativní pole

- ❑ Klíče místo indexů
- ❑ `$ceny = array('pneu'=>1000, 'výfuk'=>3000, 'sklo'=>2500);`
- ❑ Přístup k prvku: `$ceny['výfuk']`
- ❑ Přidání nového prvku:
`$ceny['karosérie'] = 10000;`
- ❑ Procházení pole – `foreach`, `while`, `list`, `each`, `reset`

Vícerozměrná pole (1)

Kód	Popis	Cena
PN	pneumatika	1000
VY	Výfuk	3000
CS	čelní sklo	5000

```
$produkty = array( array( 'PN', 'pneumatika', 1000),  
                   array( 'VY', 'výfuk', 3000),  
                   array( 'CS', 'čelní sklo', 5000));  
  
echo $produkty[0][1];
```

Vícerozměrná pole (2)

□ Vícerozměrné asociativní pole

```
$produkty = array( array( kod=>'PN',  
                        popis=>'pneumatika',  
                        cena=>1000),  
                  array( kod=>'VY',  
                        popis=>'výfuk',  
                        cena=> 3000),  
                  array( kod=>'CS',  
                        popis=>'čelní sklo',  
                        cena=> 5000));  
  
echo $produkty[0]['popis'];
```

□ Procházení - for + while/list/each

Řazení polí

- ❑ `$produkty = array('pneu', 'výfuk', 'sklo');`
- ❑ `sort($produkty);`
- ❑ Case sensitive, velká písmena před malými
- ❑ Asociativní pole
 - Podle hodnoty: `asort($ceny);`
 - Podle klíče: `ksort($ceny);`
- ❑ Reverzní řazení: `rsort()`, `arsort()`, `krsort()`
- ❑ Řazení definované uživatelem
 - `usort(pole, funkce);`

Pole - další funkce

- ❑ Náhodné zamíchání pole: `shuffle(pole);`
- ❑ Zásobníkové vkládání a vybírání: `array_push,`
`array_pop`
- ❑ Načtení pole ze souboru:
`$objednavky = file(soubor); // + explode`
- ❑ Zjištění velikosti pole
 - `count()`
 - `sizeof()`
 - `array_count_values()` – vrací pole, klíče jsou původní hodnoty, hodnoty jsou počty původních hodnot
- ❑ Extrakce asoc. pole do proměnných: `extract(pole)`

Řetězce (1)

- ❑ Odříznutí bílých znaků: trim(), ltrim(), chop()
- ❑ Tisk řetězců
 - echo – rozdíl mezi "..." a `...`
 - printf – navíc formátování
 - sprintf – vrací zformátovaný řetězec
- ❑ Převody znaků: strtoupper(), strtolower()
- ❑ Přidání escape znaků při vkládání do db: addslashes(), stripslashes()
- ❑ Rozdělování/slučování řetězců: explode(), implode()/join(), strtok()

Řetězce (2)

- Podřetězce
 - substr(řetězec, start, délka)
 - Záporná čísla znamenají od konce
- Porovnávání
 - ==
 - strcmp(\$ret1, \$ret2) – vrací <0, 0, >0
- Délka řetězce: strlen(\$ret)
- Hledání podřetězce v řetězci:
strstr(\$retezec, \$hledany);
- Pozice v řetězci: strpos(\$retezec, \$hledany)
- Nahrazení: str_replace

Require a include (1)

- ❑ Oba umožňují vložení kódu z jiného souboru
- ❑ Kód se parsuje – `<?php ... ?>`
- ❑ Rozdíly
 - Starší verze – include proveden tolikrát kolikrát je vložen a require jednou (nově `require_once`, `include_once`)
 - Při neexistenci způsobuje include warning, ale require fatal error => require použít u životně důležitých funkcí
 - Soubory vložené pomocí include mohou vracet hodnotu
 - Soubor vložený require je načten ikdyž je v podmínce a ta není splněná
- ❑ Proměnné dostupné v hlavním i ve vloženém souboru

Require a include (2)

□ Hlavní soubor:

```
<?php require('header.php'); ?>
```

Obsah

```
<?php require('header.php'); ?>
```

□ header.php

- Začíná <html>
- logo, menu atd.

□ footer.php

- Patička – např. copyright
- Končí </html>

Require a include (3)

openfile.php:

```
<?php
    @ $fp = fopen($name,
    $mode);
    if (!$fp) {
        echo "Nelze...";
        return 0;
    }
    else {
        return 1;
    }
?>
```

main.php:

```
<?php
$name = 'file.txt';
$mode = 'r';
$result =
    include('openfile.php')
    ;
if ($result == 1) {
    ... lze použít $fp...
}
?>
```

Funkce

- ❑ Jména funkce nejsou case sensitive
- ❑ Konvence – lower case
- ❑ Volání: `fname(parametry);`
- ❑ Deklarace: `function fname() {...}`
- ❑ Funkce lze deklarovat odděleně + `require`
- ❑ Uvnitř funkce lze ukončit php kód a psát html
- ❑ Jména funkcí
 - Lze: `jmeno()`, `jmeno2()`, `jmeno_tri()`, `_jmenoctyri()`
 - Nelze: `5jmeno()`, `jmeno-sest()`, `fopen()`

Parametry

- Povinné:
function create_table(\$ data) {...}
- Volitelné:
function create_table(\$table, \$border=1) {...}
- Rozsah platnosti lokálních proměnných je pouze uvnitř funkce
- Globální proměnné nejsou viditelné uvnitř funkcí (Ize pomocí global \$var;)
- Předávání parametrů
 - Hodnotou: function fname(\$par) {}
 - Odkazem: function fname(&\$par) {}
- Vracení hodnot:
 - return;
 - return 5;