

Програмски Практикум

Втора лабораториска вежба

Полиња (Arrays):

Што треба да знаете за полињата:

- ✓ Се дефинираат со функцијата: `array()`. Како параметри се ставаат парови *клуч=>вредност*, одделени со запирка. Може и да додавате само вредности одделени со запирка. Во тој случај клучевите ќе бидат integer вредности.
Пример: `$arr = array('kluc' => 'vrednost', 'kluc1' => 'vrednost1', ...);`
- ✓ Во основа, во php може да дефинирате **нумерички** и **асоцијативни** полиња. Нумеричките полиња за клучеви имаат integer вредности по редослед. Асоцијативните полиња може да содржат било кои integer или string вредности за клучеви.
- ✓ До вредностите на полето може да пристапите со:
`$value = $arr[$key]`, или пак може да доделите некоја вредност со: `$arr[$key] = $value`.
- ✓ Исто така може да дефинирате и **повеќе-димензионали** полиња. Тоа се полиња кои што содржат едно или повеќе полиња.
- ✓ Најчесто употребувани функции за работа со php полиња:
 - `count($arr)` – го дава бројот на елементи во полето. За оваа намена може да се користи и функцијата `sizeof($arr)`.
 - `array_push($arr, $value)` – се користи за додавање на елементи во полето.
 - `array_pop($arr)` – се користи за исфрлање на последниот елемент од полето
 - `array_key_exists($key, $arr)` – се користи за проверка дали одреден клуч постои во полето.
 - Постојат уште голем број на функции за работа со полиња во php, комплетната листа може да ја најдете на:
<http://www.php.net/manual/en/ref.array.php>

Вежба 1.1 – Дефинирање на полиња

Дефинирајте 3 полиња и тоа:

1. Нумеричко поле кое ќе ги содржи вредностите: 2, 5, 6, 10, 41, 24, 32, 9, 16, 19.
2. Асоцијативно поле кое ќе содржи 3 вредности и тоа: *клучот ќе биде: вашето име, вредноста ќе биде: вашето име*. Истото и за презиме и градот од кој доаѓате.
3. Повеќе-димензионално поле кое ќе содржи integer вредности по ваш избор.

Вежба 1.2 – Изминување на полиња

Со користење на `foreach()` циклусот, изминете го нумеричкото поле од претходната вежба. Притоа на екран испишете ги вредностите на полето за секој клуч:

```
foreach (array as key=>value)
{
    code to be executed;
}
Пример:
foreach($arr as $key=>$value)
{
    ...
}
```

Вежба 1.3 – Изминување на полиња и додавање во ново поле

Со помош на некое од горните изминувања, изминете го нумеричкото поле од првата вежба и во ново поле додајте ги сите вредности поголеми од 20. Додавањето извршете го со функцијата **`array_push()`**.

Напомена: Пред почетокот на циклусот, дефинирајте го полето како празно поле.

Вежба 1.4 - Должина на стринг

Во дадената реченица, пресметајте ја должината на секој од зборовите и вредностите ставете ги во асоцијативно поле, така што клуч ќе биде самиот збор, а вредноста ќе биде бројот на карактери во тој збор.

PHP is a widely-used general-purpose scripting language that is especially suited for Web development

*Напомена: Може да ја користите функцијата `explode()`, која што **враќа** поле за даден стринг:*

`array explode (string $delimiter , string $string)`

\$delimiter – по што да делите, во случајов тоа е празно место (' ').

Работа со html форми

Во php, вредностите од html формите може да ги собирате со помош на два методи:

✓ GET

Доколку го користите овој метод, вредностите што ги испраќате до php скриптата се видливи за сите (се прикажуваат во барот за адреса во веб прелистувачот). До вредностите може да пристапите преку предефинираната променлива: `$_GET`.

✓ POST

Во овој метод, вредностите што ги испраќате не се видливи за другите. И големината на вредностите што ги испраќате не е ограничена, за разлика од претходниот метод. До вредностите може да пристапите преку предефинираната променлива: `$_POST`.

Со користење на предефинираната променлива `$_REQUEST`, може да пристапите до вредностите испратени преку двете методи (POST и GET).

`$_GET`, `$_POST` и `$_REQUEST` се глобални предефинирани полиња во php и до нивните вредности може да пристапите со:

`$_GET['ime_na_input_pole']` или `$_POST['ime_na_input_pole']`.

Вежба 2.1 – Работа со форми

Со помош на post и get методите и со html формата прикажана подолу, соберете ги податоците од формата и прикажете ги на екран.

```
<html>
  <head>
    <title>Vezba so formi</title>
  </head>
  <body>
    <form method="GET" action="ime na skripta">
      <h3>Registracija:</h3>
      Ime: <input type="text" name="ime" /> <br />
      Prezime: <input type="text" name="prezime" /> <br />
      Email: <input type="text" name="email" /> <br />
      Masko: <input type="radio" name="pol" value="1" />
      Zensko: <input type="radio" name="pol" value="0" /> <br />
      <input type="submit" name="submit" value="Vnesi" />
    </form>
  </body>
</html>
```

Напомена: Формата поставете ја во една скрипта: form.php, а собирањето на податоците во друга: collect.php (треба да ја дефинирате во атрибутот **action** на формата). Со помош на функцијата print и користење на конкатенација на стрингови, испишете ги податоците на екран.

Вежба 2.2– Валидација на форма

Со користење на функцијата isset() и проверка дали е внесен текст, направете едноставна валидација на формата од горната вежба. Проверката дали е внесен текстот може да ја направите со помош на споредба со празен стринг. На пример: (\$var != "").

Функцијата isset() проверува дали некоја променлива е сетирана и враќа соодветна Boolean вредност.

Вежба 2.3– Форма за регистрација и користење на JQuery валидација

Во делот code/jquery_form.php имате дадено форма со валидација преку употреба на JQuery. Ваша задача е да го разгледате кодот, и да го испробате.

Вежба 2.4– Форма за регистрација и користење на колачиња

Изменете ја претходната форма на следниов начин. Додадете checkbox “запамти ме”. Доколку корисникот го штиклира, тогаш потребно е при пополнување на претходната форма, да поставите колаче во кое ќе го запишете корисничкото име на корисникот, како и точното време на неговото логирање. Доколку времето е помалку од 2 часа, потребно е да се испечати порака со содржина „Hello \${ime_na_korisnikot}“.

Вежба 2.4– Форма за регистрација и користење на колачиња и сесија

При submit на претходната форма, почнете сесија и во колачето зачувајте го сесиското ID. Доколку времето на последен пристап е помалку од 1 час, и сесиското ID е поставено во колачето, потребно е да се испечати порака со содржина „Hello \${ime_na_korisnikot}, you are still logged in!!!!“.