**Uključenje fajlova**

Jedinica: 9 od 19

**Rezime**

* U PHP-u možete uključiti kod iz eksternih fajlova u aktuelni kod;
* Naredbe i funkcije za importovanje eksternih fajlova su include, require, include\_once i require\_once;
* Ako izvršenje programa zavisi od eksternog koda, koristite require ili require\_once;
* Ako izvršenje koda nije zavisno od eksternog koda, koristite include ili include\_once;
* require i require\_once bezuslovno prekidaju izvršenje programa ako fajl ne postoji;
* include i include\_once nastavljaju izvršenje programa ako fajl ne postoji.

U ovoj lekciji radićemo sa fajlovima i videćemo kako jedan fajl možemo uključiti u drugi. Korišćenje već napisanog koda je veoma korisna i česta praksa u programiranju. Na primer, ukoliko web sajt sadrži meni koji se ponavlja na svakoj strani, tada je mnogo jednostavnije napisati kod jedanput, a zatim ga dinamički uključiti na strane na kojima treba da se pojavi, nego pisati kod u okviru svake strane. Ovo je moguće uključivanjem fajlova od strane servera.

Iako ćemo u nastavku lekcije govoriti o ukupno 4 funkcije čija je svrha uključivanje fajlova, možemo reći da su dve osnovne:

* + include()
  + require()

Ove funkcije imaju mogućnost da sadržaj eksternog fajla uključe u kod sa kog su pozvane pre nego što se pristupi izvršavanju tog koda. Veoma su značajne jer se koriste za inkludovanje istog sadržaja na veći broj strana (headeri, footeri, segmenti strane i sl.).

Fajlovi koji se uključuju obično sadrže HTML ili PHP kod i snimaju se često sa ekstenzijom *.inc*, iako mogu biti snimljeni i sa drugim ekstenzijama. Sadržaj ovakvih fajlova se kodira jedanput, a zatim poziva od strane strana na kojima je potreban. Ukoliko dođe do izmene na originalnom fajlu, one su vidljive na svim stranama na kojima se koristi.

Na primer, dešava se da sve strane jednog web sajta, na svom vrhu, treba da imaju isto zaglavlje. Neka je HTML kod zaglavlja:

|  |  |
| --- | --- |
| 1  2  3 | <div style = "border: ridge 1px; width: 95%; background-color: #F0F0F0; padding: 5px" >  <h3> Welcome to our site! </h3>  </div> |

Ovaj kod je moguće snimiti unutar fajla nazvanog header.inc. Nakon kreiranja fajla zaglavlja, neophodno je da se on uključi na svaku stranu na kojoj treba da se prikaže. To se postiže pomoću naredbi za uključivanje eksternih fajlova.

|  |  |
| --- | --- |
| 1  2  3  4 | <?php  require('header.inc');  echo "<p> Some text..... </p>";  ?> |

Ukoliko fajlovi koji se uključuju sadrže neke poverljive informacije, tada bi trebalo da budu snimljeni sa ekstenzijom .php, kako korisnici ne bi mogli da vide njihov originalan sadržaj, već samo HTML kod, koji je rezultat izvršavanja njihovog koda.

Kao što vidite, fajl koji se importuje nije jezički ograničen (u primeru, to je html fajl), a može biti i bilo koji drugi jezik (sve dok sintaksa ostaje ispoštovana).

Funkcija ***require*** nije jedina funkcija koja može da uključi eksterni fajl u aktuelni [PHP skript](https://www.link-elearning.com/linkdl/opisPojma.php?id=139122). Zapravo, postoje četiri funkcije koje ovo mogu da urade i svaka ima svoje osobenosti.

**Require**

Funkcija **require()** poziva fajl header.inc i čita njegov sadržaj. Ovaj sadržaj se onda prikazuje kao deo strane sa koje je pozvan. Na primer, ako imamo jedan PHP fajl sa sledećim sadržajem i taj fajl nazovemo *myFile.php*:

|  |  |
| --- | --- |
| 1  2  3 | <?php    echo "Hi!";  ?> |

A zatim, napravimo još jedan fajl sledećeg sadržaja:

|  |  |
| --- | --- |
| 1  2  3 | <?php  require "myFile.php";  ?> |

Ukoliko startujemo ovaj fajl, on će prikazati na strani poruku: *Hi!*, jer je u sebe implementirao sadržaj drugog fajla.

Prilikom implementacije fajla na ovaj način morate biti pažljivi jer se ne implementira izlazni rezultat fajla koji se uključuje, već čitav njegov izvorni sadržaj.

To znači sledeće:

Fajl myFile.php iz primera, ako se startuje zasebno, emituje poruku **Hi!**. To je sve što će biti izlaz tog fajla, jer je to ono što PHP interpreter emituje. Ali, ukoliko taj fajl implementiramo u drugi PHP fajl kroz funkcije za implementaciju, u njega ulazi njegov izvorni PHP kod, a to je echo „hi”;.

Kada bismo, na primer, stavili u sadržaj fajla myFile.php sledeći kod:

|  |  |
| --- | --- |
| 1  2  3 | <?php  $a = "my variable";  ?> |

i emitovali taj fajl direktno u pretraživač, strana bi bila prazna.

Ali, ako isti fajl uvrstimo u postojeći PHP kod i zatražimo promenljivu $a, ona će dati regularan rezultat.

|  |  |
| --- | --- |
| 1  2  3  4 | <?php  require "myFile.php";  echo $a;  ?> |

**Primer:**

Kao što smo već napomenuli, suština svakog programiranja je da se jedan kod piše samo jednom. S obzirom na kreiranje web stranica, koliko god da ih je potrebno, uvek se javlja potreba za header i footer delovima za koje je preporučljivo da budu jednaki na svim stranicama. Ovakavo rešenje formatiranja stranica dobijamo pomoću sledećeg koda i korišćenjem funkcije require:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <?php  require ('header.php'); ?>    ... content of page ....    <?php  require ('footer.php'); ?> |

Ovakvim načinom rada smo u mogućnosti da, iako smo napisali veliki broj stranica, njihove headere i footere menjamo samo na jednom mestu i sadržaj će biti reprodukovan na svim stranicama.

U okviru ovakvog primera uglavnom se koristi kombinacija HTML-a i PHP naredbi koje generišu dinamičke delove stranica. Ukoliko vam se javi potreba da sadržaj datoteka obavezno obrađuje kao običan tekst i da se ne izvršava php kod, alternativu predstavlja funkcija koja samo učitava sadržaj datoteke bez sintaksne analize, funkcija je *readfile(),* koja iščitava sadržaje fajlova.

**Include**

Ukoliko sadžaj strane zavisi od sadržaja fajla koji importujete (na primer, u njemu su neke ključne promenljive), onda možete koristiti funkciju *require*, jer će ona, ukoliko fajl iz bilo kog razloga ne bude uvršćen u postojeći kod, prekinuti izvršavanje programa.

Ali, ako vam sadržaj nije od velike važnosti (na primer, fajl koji uvršćujete je neki čitač vesti, koji prikazuje neke vesti na nekom manje bitnom delu strane), možete iskoristiti i funkciju include. Njena sintaksa je ista kao i require, samo što će se, ukoliko učitavanje ne uspe, nastaviti izvršavanje skripte.

Najbolje je da funkciju include „zamaskirate” error suppressor operatorom, tako da problem pri učitavanju ne poremeti definiciju izgleda strane:

|  |  |
| --- | --- |
| 1 | @include "myFile.php"; |

**Include\_once i require\_once()**

Zamislite sledeću situaciju:

|  |  |
| --- | --- |
| 1  2  3  4  5 | require( "myFile.php" );  echo $a;  $a = "some other value: ";  require( "myFile.php" );  echo $a; |

Ako koristimo malopređašnji sadržaj fajla myFile.php, onda će ovaj program dva puta emitovati tekst moja promenljiva, što najverovatnije nije efekat koji bismo želeli, jer smo u toku koda izmenili vrednost promenljive.

Da se ovo ne bi događalo, koristimo naredbu **include\_once** i funkciju **require\_once**.

Kada PHP dođe do require\_once ili include\_once funkcije, proveriće da li već postoji implementacija istoimenog fajla i, ukoliko postoji, neće izvršiti ponovno uključivanje.

Imajte na umu da, za razliku od include naredbi, require ne vraćaju nikakav rezultat, s obzirom na to da momentalno prekidaju izvršavanje programa. To znači da sledeća izjava neće funkcionisati:

|  |  |
| --- | --- |
| 1 | require( "nonexistentFile.php" ) or die( "no file" ); |

Ako želite da postignete ovako nešto, onda je možda najbolje da upotrebite drugačiji mehanizam:

|  |  |
| --- | --- |
| 1  2 | if( !@include\_once "nonexistentFile.php" )      echo "no file"; |

Ili možete proveriti postojanje fajla, a zatim u odnosu na tu proveru, izvršiti narednu akciju:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | $file = "nonexistentFile.php";  if( file\_exists($file)){      require\_once ( $file );  }else{      echo "file does not exist";  } |

**Opcije *auto\_prepend\_file* i *auto\_append\_file***

Postoji još jedan način da na svakoj stranici dodajete zaglavlje i podnožje (header i footer) bez upotrebe require() i include() naredbi. Datoteka php.ini, između ostalog, ima i dve opcije za konfiguraciju, *auto\_prepend\_file* i *auto\_append\_file*. Ukoliko podesite ove opcije tako da upućuju na vaše datoteke zaglavlja i podnožja, one će se učitavati na početku i na kraju svake stranice. Vaše datoteke koje dodajete na ovaj način ponašaju se kao da ste ih dodali pomoću naredbi require() i include() (što znači da će se, ukoliko datoteka nije dostupna, pojaviti greška).

U okviru Windowsa, ove opcije se zadaju na sledeći način:

auto\_prepend\_file = "c:/path/to/files/header.php"  
auto\_append\_file = "c:/path/to/files/footer.php"

Ovim postupkom uključivanja fajlova bez upotrebe include() naredbe omogućavate uključivanje fajlova na svaku vašu stranicu, ali istovremeno se i obavezujete da će vam svaka stranica imati zaglavlje i podnožje. Šta to znači? Opcije koje smo objasnili vam nameću uključene fajlove na svim stranicama, što ukazuje na problem kada želite da imate samostalnu stranicu bez zaglavlja i podnožja.

Korisnici Apache Web servera imaju mogućnosti izmene raznih konfiguracionih opcija, na isti način kao i pojedinačne direktorijume. Server mora biti podešen tako da omogućava redefinisanje konfiguracionih detoteka. Da bi se za određeni direktorijum definisali automatsko zaglavlje i podnožje, potrebno je da u tom direktorijumu kreirate datoteku ekstenzije *.htaccess*, i u tako kreiranu datoteku napišete sledeći sadržaj:

php\_value auto\_prepend\_file "path/to/files/header.php"  
php\_value auto\_append\_file "path/to/files/ footer.php"

Definisanje raznih opcija u datoteci *.htaccess*umesto u datoteci [php.ini](https://www.link-elearning.com/linkdl/opisPojma.php?id=139123) ili u konfiguracionoj datoteci Web servera pruža vam veliku fleksibilnost pri radu. Konfiguracione opcije, koje se podešavaju u višekorisničkom okruženju, mogu se podesiti tako da važe samo u direktorijumu svakog korisnika posebno. Ipak, kao i kod svakog metoda, i u ovom postoje mane. Konkretan problem se javlja kada korisnik pokuša da učita bilo koju datoteku iz tog direktorijuma, a ne samo jednom na početku, što nam ukazuje na usporavanje izvršenja programa.

**Vežba 1**

***Problem:***

U aplikaciju ulaze sledeće dve promenljive:

$file = "test";  
$validExtension = "php";

Potrebno je napisati takav kod koji na osnovu naziva fajla i dozvoljene ekstenzije učitava određeni fajl naredbom *include*.

***Rešenje:***

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | <?php  $file = "test";  $validExtension = "php";  switch( $validExtension )      {          case "php":              include $file . ".php";              break;          case "html":              include $file . ".html";              break;          case "js":              include $file . ".js";              break;      }  ?> |

Promenljive $file i $validExtension su stringovi. Prva promenljiva predstavlja naziv fajla, dok je druga string, koji predstavlja ekstenziju. Upotrebom switcha proveravamo da li je navedena ekstenzija validna, odnosno da li odgovara dozvoljenim. Onog trenutka kada se ekstenzija poklopi sa vrednošću za case, izvršavaju se sve linije koda do break i upravo zato ovde vršimo inkludovanje fajla. Da bismo uključili eksterni fajl, moramo definisati putanju do tog fajla. Kada se fajl koji želimo da uključimo nalazi u istom folderu u kom se nalazi i fajl za koji pišemo kod, putanja je praktično sam naziv fajla sa ekstenzijom.

**Razlika između require i include naredbi/funkcija je:**

 U require; ukoliko dođe do greške i fajl iz bilo kog razloga ne bude učitan, program prekida izvršavanje i dolazi do greške, dok u include to nije slučaj.

 U include; ukoliko dođe do greške i fajl iz bilo kog razloga ne bude učitan, program prekida izvršavanje i dolazi do greške, dok u require to nije slučaj, već skripta jednostavno nastavlja izvršavanje i pored greške.