

Kushtëzimet dhe Funksionet në PHP

Deklarimet e kushtëzuara përdoren për të kryer veprime të ndryshme bazuar në kushte të ndryshme. Mund të përdoren deklarimet e kushtëzuara për ta bërë këtë. Në PHP ekzistojnë deklarimet e mëposhtme të kushtëzuara:

`if` deklarata – ekzekuton një pjesë kodi nëse një kusht është i saktë

`if...else` deklarata – ekzekuton një pjesë kodi nëse një kusht është i saktë dhe një tjetër pjesë kodi nëse kushti është i pasaktë

`if...elseif...else` deklarata – ekzekuton një pjesë të ndryshme kodi për më shumë se dy kushte

`switch` deklarata- përzgjedh një ose më shumë blloqe kodi për tu ekzekutuar

1. Shkruani një program në PHP i cili printon një string "Mirëdita!" nëse koha aktuale është (ORA) është më e vogël se 20.
2. Shkruani një program në PHP që printon një string "Mirëdita!" nëse koha aktuale është më e vogël se 20, dhe printon një string "Mirëmbrema!" përndryshe.
3. Shkruani një program në PHP që printon një string "Mirëmëngjes!" nëse koha aktuale është më e vogël se 10, dhe printon një string "Mirëdita!" nëse koha aktuale është më e vogël se 20, përndryshe do të printojë një string "Mirëmbrema!".
4. Shkruani një program në PHP që në varësi të komandës që zgjidhet nga përdoruesi, kryhet veprimi përkatës (në rastin tonë për të qenë programi më i thjeshtë do të japim vetëm një mesazh për veprimin që do të kryhet). Cila strukturë e deklarimeve të kushtëzuara është më e përshtatshme për tu përdorur në këtë rast. Sqaroni përgjigjen tuaj!
5. Shkruani një program në PHP, që mundëson regjistrimin dhe kycjen e përdoruesit të regjistruar në faqen. Programi do të ketë këto kërkesa:
 - a. Krijoni një HTML formë që do të mundësoj regjistrimin e përdoruesit në 'sistem'. Forma le të ketë një fushë teksti për emrin dhe mbiemrin, një fushë teksti për username, një fushë teksti për fjalëkalimin dhe një tjetër fushë teksti për konfirmimin e fjalëkalimit. Poashtu forma duhet të përmbajë edhe butonin *Submit*.
 - b. Pasi të klikohet butoni *Submit* të dhënat duhet të ruhen. Ruajtjen në këtë shembull do ta bëjmë në një fajll të cilit pastaj do ti qasemi për të kontrolluar përdoruesin e regjistruar paraprakisht. Një fajll do të hapet për shkrim ku do të ruhen të dhënat e përdoruesit (emri dhe mbiemri, username, password) në një rresht të vetëm. Fajlli mbyllet. Përdoruesi do të drejtohet në faqen që përmban formën e kycjes.
 - c. Pastaj krijoni një tjetër HTML formë që do të mundësoj kycjen e përdoruesit të sapo regjistruar në 'sistem'. Forma do të ketë dy fusha teksti, një për username-in dhe një për fjalëkalimin. Poashtu edhe një buton *Submit*.

- d. Pasi të klikohet butoni *Submit* të dhënat dërgohen në server për validim. Mënyra se si bëhet validimi në rastin tonë shpjegohet në vijim. Fajlli në të cilin më herët i ruajtëm të dhënat për përdoruesin e regjistruar do ta hapim por tani për lexim të të dhënave. Shikojmë nëse kemi përputhje të të dhënës së ruajtur në fajllin me të dhënë që përdoruesi ka shënuar në formën e kyçjes. Nëse *username* nuk është i saktë, paraqitet një mesazh që thotë se përdoruesi i tillë nuk ekziston, përndryshe kontrollohet në vazhdim për *fjalëkalimin*. Nëse fjalëkalimi për atë përdorues nuk është i saktë, paraqitet një mesazh informues për gabimin e bërë, përndryshe përdoruesi drejtohet në një faqe të re me një mesazh falenderimi “Faleminderit për regjistrimin në sistemin tonë *Emri Mbiemri*”.
- e. Në fajll do të ruajmë vetëm një përdorues. Pasi të mësojmë ciklet do të kemi mundësi të ruajmë më shumë se një përdorues, ku ruajmë në rreshta të ndryshëm përdorues me të dhënat e ndryshme dhe gjithashtu të kontrollojmë secilin prej tyre për të parë se nëse përdoruesi përkatës ekziston në fajll.

PHP Date dhe Time

Funksioni `date()` i PHP përdoret për të formatuar datën dhe/ose kohën.

`date(format, timestamp)`

`format` – e detyrueshme, specifikon formatin e kohës

`timestamp` – opsionale, specifikon kohën, vlera default është data dhe koha aktuale

Timestamp është një sekuencë e karaktereve, që tregojnë datën dhe/ose kohën në të cilën ka ndodhur një ngjarje e caktuar.

Format specifikon se si të formatojmë datën dhe/ose kohën.

- `d` – përfaqëson ditën e muajit (01 deri në 31)
- `m` – përfaqëson muajin (01 deri në 12)
- `Y` – përfaqëson vitin (në katër shifra)
- `l` (shkronja e vogël e ‘L’) – përfaqëson ditën e javës

Karakteret të tjera si `“/”`, `“.”` ose `“-”` mund të përfshihen ndërmjet karaktereve për të shtuar formatim shtesë.

Karakteret tjera që përdoren për kohën:

- `H` – formati 24-orësh i orës (00 deri në 23)
- `h` – formati 12-orësh i orës me zerot paraprijëse (01 deri në 12)
- `i` – minutat me zerot paraprijëse (00 deri në 59)

- s – sekondat me zerot paraprijëse (00 deri në 59)
- a – (am ose pm)

File Handling

Është një pjesë e rëndësishme e një web aplikacioni. PHP ka funksione të ndryshme për krijim, lexim, ngarkim dhe editim të fajllave.

Funksioni `readfile(filename)` lexon një fajll dhe shkruan atë në buferin dalës. Funksioni kthen numrin e bajtëve që janë lexuar me sukses. Funksioni `readfile()` është i dobishëm nëse e gjitha që ju dëshironi është të hapni një fajll dhe të lexoni përmbajtjen e tij.

PHP File Open/Read/Close

Si bëhet hapja, leximi dhe mbyllja e një fajlli në server.

PHP Open File – `fopen()`

Një metodë më e mirë për të hapur fajllat është duke përdorur funksionin `fopen(filename, mode)`, i cili ofron më shumë opsione se vetëm funksioni `readfile()`. Parametri i parë i funksionit përmban emrin e fajllit që do të hapet dhe parametri i dytë specifikon në cilën mënyrë do të hapet fajlli.

Mënyrat në të cilën fajlli mund të hapet:

- **r** – hap një fajll vetëm për lexim; pointeri i fajllit fillon në fillim të fajllit.
- **w** – hap një fajll vetëm për shkrim; fshin përmbajtjen e fajllit ose krijon një fajll të ri nëse fajlli nuk ekziston; pointeri i fajllit fillon në fillim të fajllit.
- **a** – hap një fajll vetëm për shkrim; përmbajtja ekzistuese në fajll ruhet; pointeri i fajllit fillon në fund të fajllit; krijon një fajll të ri nëse fajlli nuk ekziston.
- **x** – krijon një fajll të ri vetëm për shkrim; kthen FALSE dhe një gabim nëse fajlli aktualisht ekziston.
- **r+** – hap një fajll për lexim/shkrim; pointeri i fajllit fillon në fillim të fajllit.
- **w+** – hap një fajll për lexim/shkrim; fshin përmbajtjen e fajllit ose krijon një fajll të ri nëse fajlli nuk ekziston; pointeri i fajllit fillon në fillim të fajllit.
- **a+** – hap një fajll lexim/shkrim; përmbajtja ekzistuese në fajll ruhet; pointeri i fajllit fillon në fund të fajllit; krijon një fajll të ri nëse fajlli nuk ekziston.
- **x+** – krijon një fajll të ri për lexim/shkrim; kthen FALSE dhe një gabim nëse fajlli aktualisht ekziston.

PHP Read File – `fread()`

Funksioni `fread()` lexon nga një fajll i hapur. Parametri i parë i funksionit `fread()` përmban emrin e fajllit nga i cili lexohet dhe parametri i dytë specifikon numrin maksimal të bajtëve që do të lexohen. Funksioni `filesize(filename)` jep madhësinë e fajllit përkatës.

PHP Close File – `fclose()`

Funksioni `fclose()` përdoret për të mbyllur një fajll. Është praktikë e mirë programimi që të mbyllet të gjithë fajllat pasi të kryhet puna me ta. Funksioni `fclose(filename)` kërkon si parametër hyrës emrin e fajllit që do të mbyllet.

PHP Read Single Line – `fgets(filename)`

PHP Check End-Of-File – `feof(filename)`

PHP Read Single Character – `fgetc(filename)`

PHP File Create/Write

Si të krijojmë dhe të shkruajmë në një fajll në server.

PHP Create File – `fopen()`

Funksioni `fopen()` gjithashtu përdoret për të krijuar një fajll. Nëse funksioni përdoret në një fajll që nuk ekziston, një fajll i ri me emrin përkatës të fajllit, do të krijohet.

PHP Write to File – `fwrite()`

Funksioni `fwrite(filename, string)` përdoret për të shkruar në një fajll. Parametri i parë i funksionit përmban emrin e fajllit në të cilin do të shkruajmë dhe parametri i dytë është stringu që do të shkruhet.

PHP Overwriting

Çfarë ndodh nëse hapim një fajll ekzistues për shkrim. Të gjithë të dhënat ekzistuese do të fshihen dhe fillohen me një fajll të zbrazët.