



2. ÓRA

A PHP telepítése

Mielőtt megkezdénénk az ismerkedést a PHP nyelvvel, be kell szerezniünk, telepíteniünk és beállítanunk a PHP-értelmezőt. A feldolgozóprogram számos operációs rendszerre elérhető és több kiszolgálóval is képes együttműködni.

Ebben az órában a következőket tanuljuk meg:

- Milyen platformokat, kiszolgálókat és adatbázisokat támogat a PHP?
- Honnan szerezhetjük be a PHP-t és más nyílt forráskódú programokat?
- Hogyan telepíthető a PHP Linux rendszerre?
- Hogyan állíthatók be a fontosabb paraméterek?
- Hol található segítség, ha nem sikerül a telepítés?

Operációs rendszerek, kiszolgálók, adatbázisok

A PHP teljesen platformfüggetlen, ami azt jelenti, hogy fut Windows operációs rendszeren, a legtöbb UNIX rendszeren – beleértve a Linuxot –, sőt még Macintosh gépeken is. A támogatott kiszolgálók köre igen széles. A legnépszerűbbek: Apache (szintén nyílt forráskódú és platformfüggetlen), Microsoft Internet Information Server, WebSite Pro, iPlanet Web Server, OmniHTTPD és Microsoft Personal Web Server. Az utóbbi kettő akkor tehet nagy szolgálatot, ha internetkapcsolat nélkül szeretnénk fejleszteni, bár az Apache is alkalmas erre Windows környezetben.

A PHP fordítható önálló alkalmazássá is, így az értelmező parancssorból is hívható. Ebben a könyvben webalkalmazások fejlesztéséhez fogjuk használni a PHP-t, de nem szabad alábecsülni a szerepét általános programozói eszközként sem.

A PHP-t alapvetően úgy tervezték, hogy könnyen összhangba hozható legyen a különböző adatbázisokkal. Ez az egyik oka a PHP népszerűségének a webalkalmazások készítése terén. Számos adatbázis – Adabas D, InterBase, Solid, dBASE, mSQL, Sybase, Empress, Microsoft SQL, MySQL, Velocis, FilePro, Oracle, UNIX dbm, Informix és PostgreSQL – közvetlenül csatlakoztatható a PHP-hez. A közvetlenül nem támogatott adatbázisok mellett a PHP-ben kapcsolatot létesíthetünk az ODBC szabványt használó programokkal is.

A könyv példáiban Linux operációs rendszer alatt Apache és MySQL programokat használunk. Ezek ingyenesen letölthetők az Internetről, valamint könnyen telepíthetők és használhatók majdnem minden PC-n. A Linux rendszer telepítéséről további információ a <http://www.linux.org/help/beginner/distributions.html> címen található. A Linux PowerPC gépen is használható, a LinuxPPC rendszerrel: <http://www.linuxppc.org/>. Magyarországon a Linux közösség honlapja a <http://www.linux.hu/>.

A MySQL adatbázisrendszer, amit ebben a könyvben használni fogunk, a <http://www.mysql.com/> címről tölthető le. Számos operációs rendszerre elérhető, beleértve a UNIX, Windows és OS/2 rendszereket. A MySQL magyar tükörszolgálója a <http://mysql.sote.hu/>.

Természetesen nyugodtan dolgozhatnánk Windows NT vagy MacOS rendszer alatt is, mivel a PHP platformfüggetlen nyelv.

A PHP beszerzése

A PHP 4-es változata a <http://www.php.net/> címről tölthető le. Mivel a PHP nyílt forráskódú, nem kell a bankkártyánkat kéznél tartanunk, amikor letöltjük az értelmezőt. Magyarországon a <http://hu.php.net/> tükörkiszolgálót érdemes meglátogatni.

A PHP webhelye kiváló információforrás PHP programozóknak. A <http://www.php.net/manual/> címen a teljes kézikönyv elolvasható, kiegészítve más programozók által írt hasznos megjegyzésekkel. Ezen a címen olvasható a magyar PHP kézikönyv is. A PHP honlapról a dokumentáció is letölthető különböző formátumokban.

A Zend Engine és más Zend termékek honlapja a <http://www.zend.com/>. Itt híreket, illetve cikkeket olvashatunk a PHP és a Zend világból.

A PHP 4 telepítése Apache webkiszolgálót használó Linuxra

Ebben a részben végigvezetünk egy PHP telepítést egy Apache webkiszolgálót használó Linux rendszeren. A folyamat többé-kevésbé ugyanez más UNIX platformokon is. Elképzelhető, hogy az általunk használt rendszerre készült előre fordított változat is, így még egyszerűbben telepíthetnénk az értelmezőt, a PHP fordítása azonban nagyobb szabadságot ad a feldolgozóba kerülő szolgáltatásokat illetően.

Mielőtt megkezdjük a telepítést, ellenőrizzük, hogy rendszergazdaként (root) jelentkeztünk-e be a rendszerbe. Ha a kiszolgálót nem érhetjük el root felhasználóként, forduljunk a rendszergazdához a PHP telepítésével kapcsolatos kéréseinkkel.

A PHP-t kétféleképpen lehet Apache modulként előállítani. Egyrészt újrafordíthatjuk a webkiszolgálót és beépíthetjük a PHP-értelmezőt, másrészt a PHP-t dinamikusan megosztott objektumként (DSO, Dynamic Shared Object) is fordíthatjuk. Ha Apache kiszolgálónk DSO-támogatással ellátott, képes lesz az új modul használatára anélkül, hogy újrafordítanánk a programot. Ez a legegyszerűbb módja annak, hogy beüzemeljük a PHP-t, ezért ezt az eljárást fogjuk tárgyalni.

Ha ellenőrizni kívánjuk, hogy az Apache rendelkezik-e DSO-támogatással, el kell indítanunk az Apache futtatható állományát (httpd) az `-l` paraméterrel.

```
/www/bin/httpd -l
```

A program ekkor egy listát ad a rendelkezésre álló beépített modulokról. Ha a `mod_so.c` elem szerepel a listában, akkor az Apache alkalmas az alább bemutatott módszerrel történő bővítésre. Egyéb esetben újra kell fordítani, amihez a dokumentáció tartalmazza az összes szükséges információt.

Ha még nem tettük meg, le kell töltenünk a PHP legfrissebb változatát. A `tar` fájl `gzip`-pel tömörített, így első lépésben ki kell csomagolnunk:

```
tar -xvzf php-4.x.x.tar.gz
```

Ha sikeresen kibontottuk a csomagot, lépünk át a keletkezett könyvtárba:

```
cd ../php-4.x.x
```

Ebben a könyvtárban található a `configure` program, melynek a megfelelő paraméterekkel megadhatjuk, milyen szolgáltatásokat építsen be a PHP-be. Ebben a példában csak néhány hasznos parancssori paramétert adunk meg, de természetesen számos más lehetőség is rendelkezésre áll. Később megnézzük néhány további elemet a `configure` paramétereinek közül.

```
./configure --enable-track-vars \  
            --with-gd \  
            --with-mysql \  
            --with-apxs=/www/bin/apxs
```

Lehetséges, hogy a `--with-apxs` paraméternek átadott elérési útnak a rendszerünkön másnak kell lennie, mivel telepítéskor az `apxs` esetleg éppen az Apache futtatható állománnyal megegyező könyvtárba került.

Amikor a `configure` lefutott, elindítható a `make` program. Ennek futtatásához a rendszernek tartalmaznia kell egy C fordítót.

```
make  
make install
```

Ezekkel a parancsokkal a PHP fordítása és telepítése befejeződött, már csak az Apache beállítására és futtatására van szükség.

A configure néhány paramétere

Amikor lefuttattuk a `configure`-t, megadtunk néhány parancssori paramétert, melyek meghatározták, milyen lehetőségekkel ruházzuk fel a PHP-t. Ha a kibontott PHP csomag könyvtárában a következő parancsot adjuk ki, a `configure` megadja a lehetséges paramétereiket:

```
./configure --help
```

Mivel a lista rendkívül hosszú, célszerű előbb egy szövegfájlba irányítani, így kényelmesebben olvasható:

```
./configure --help > lehetosegek.txt
```

Annak ellenére, hogy a fenti parancs kimenete eléggé érthető, néhány fontos lehetőséget meg kell említenünk – mégpedig azokat, amelyek a könyv szempontjából számunkra érdekesek.

--enable-track-vars

Ez a szolgáltatás automatikusan előállítja számunkra a PHP oldalakon kívülről érkező adatokhoz tartozó asszociatív tömböket. Ezek a GET vagy POST kéréssel érkezett adatok, a visszaérkező süti-értékek, a kiszolgálói és környezeti változók. A tömbökkel a hetedik órában foglalkozunk bővebben, a HTTP kapcsolatokat a tizenharmadik órában részletezzük. A fenti rendkívül gyakran használt `configure` paraméter, mivel nagyon kellemes lehetőség a beérkező adatok követésére. A PHP 4.0.2-es változatától kezdve mindig be van kapcsolva, így nem kell külön megadni.

--with-gd

A `--with-gd` paraméter engedélyezi a GD könyvtár támogatását. Amennyiben a GD könyvtár telepítve van a rendszeren, ez a paraméter lehetőséget ad dinamikus GIF, JPEG vagy PNG képek készítésére a PHP programokból. A dinamikus képek előállításáról a tizennegyedik órában írunk. Ha a GD-t korábban nem az alapbeállítású könyvtárba telepítettük, megadható a szokásostól eltérő elérési út is:

```
--with-gd=/eleresi/ut/a/megfelelo/konyvtarhoz
```

--with-mysql

A `--with-mysql` engedélyezi a MySQL adatbázisok támogatását. Ha a rendszeren a MySQL nem az alapbeállítású könyvtárban található, megadható a szokásostól eltérő elérési út is:

```
--with-mysql=/eleresi/ut/a/megfelelo/konyvtarhoz
```

Mint már tudjuk, a PHP támogat számos más adatbázisrendszert is. Az 1.2-es táblázatban ezek közül láthatunk néhányat, a hozzájuk tartozó `configure` paraméterekkel.

2.1. táblázat Néhány adatbázis és a hozzá tartozó configure paraméter

<i>Adatbázis</i>	<i>configure paraméter</i>
Adabas D	--with-adabas
FilePro	--with-filepro
mSQL	--with-msql
Informix	--with-informix
iODBC	--with-iodbc
OpenLink ODBC	--with-openlink
Oracle	--with-oracle
PostgreSQL	--with-pgsql
Solid	--with-solid
Sybase	--with-sybase
Sybase-CT	--with-sybase-ct
Velocis	--with-velocis
LDAP	--with-ldap

Az Apache beállítása

Miután sikeresen lefordítottuk az Apache-t és a PHP-t, módosítanunk kell az Apache beállításait tartalmazó `httpd.conf` fájlt. Ez az Apache könyvtárának `conf` alkönyvtárban található.

A következő sorok hozzáadása szükséges:

```
AddType application/x-httpd-php .php .php3
AddType application/x-httpd-php-source .phps
```

Keressünk rá ezekre a sorokra a `httpd.conf` fájlban! Az újabb Apache kiadásokban ez már szerepel, csak egy megjegyzésjelet kell kitörölnünk a sorok elejéről. Ezzel biztosítjuk, hogy a PHP-elemző fel fogja dolgozni a `.php` és `.php3` kiterjesztéssel rendelkező fájlokat. A `.php3` kiterjesztésre azért lehet szükség, mert számos régebbi program ezt használja, így módosítás nélkül tovább alkalmazhatjuk ezeket is.

A `.phps` kiterjesztéssel rendelkező fájlok PHP forrásként kerülnek a böngészőhöz, ami azt jelenti, hogy a forráskód HTML formátumúvá alakul és a nyelvtani elemek színiemeléssel jelennek meg, ami hasznos segítség lehet a programok hibáinak felderítésében. Ha ügyfeleink miatt esetleg a hagyományos oldalaknál megszokott `.html` kiterjesztést választjuk a PHP számára, a következő beállítást kell alkalmaznunk:

```
AddType application/x-httpd-php .html
```

Tulajdonképpen bármilyen kiterjesztéshez köthetjük a PHP-feldolgozót. Az ajánlott a `.php`, a `.html` kiterjesztés azonban nem feltétlenül jó választás, ugyanis ilyen beállítás esetén minden kiküldött HTML lap áthalad a PHP-elemzőn, ezáltal jelentősen csökkenhet a kiszolgálás sebessége.

Ha a PHP előretelepítve található meg a kiszolgálón és nincs elérésünk az Apache beállításait tartalmazó fájlhoz, létrehozhatunk egy `.htaccess` nevű állományt a saját könyvtárunkban és abban is megadhatjuk a fenti sorokat. A `.htaccess` fájlok hatása kiterjed az adott könyvtárra és annak minden alkönyvtárára is. Ez a megoldás azonban csak akkor működőképes, ha az Apache `AllowOverride` beállítása az adott könyvtárra a `FileInfo` vagy az `All` értéket tartalmazza.

A `.htaccess` az alapbeállítású fájlnev, amit a könyvtár speciális beállításaihoz használhatunk, de az adott rendszeren más is lehet. Ezt a `httpd.conf` állomány `AccessFileName` beállítása határozza meg. Ez a fájl általában akkor is olvasható, ha nem rendelkezünk rendszergazdai jogokkal a kiszolgálón.

A `.htaccess` fájl tökéletes módja annak, hogy testreszabjuk a tárhelyünket, ha a kiszolgáló beállítófájljában nem módosíthatjuk a paramétereket. A PHP működését közvetlenül azonban a `php.ini` szabályozza.

php.ini

A PHP működését a fordítás vagy telepítés után is befolyásolhatjuk, a `php.ini` használatával. UNIX rendszereken az alapbeállítású könyvtár a `php.ini` fájl számára a `/usr/local/lib`, Windows rendszereken a Windows könyvtára. Emellett a feldolgozásra kerülő PHP oldal könyvtárában – a munkakönyvtárban – elhelyezett `php.ini` fájlban felülbírálnak a korábban beállított értékeket, így akár könyvtáranként különböző beállításokat adhatunk meg.

A letöltött PHP csomag könyvtárában található egy minta `php.ini` fájl, amely a „gyári beállításokat” tartalmazza. Ezek az értékek lépnek érvénybe akkor, ha a PHP nem talál egy `php.ini` fájlt sem.

Az alapértékek elegendőek lehetnek ahhoz, hogy a könyv példait futtassuk, ám célszerű néhány módosítást elvégezni; ezeket a huszonkettedik órában tárgyaljuk.

A `php.ini` fájl beállításai egy névből, egy egyenlőségjelből és egy értékből állnak. A szóközöket a feldolgozó figyelmen kívül hagyja.

Ha a PHP előretelepítve áll rendelkezésre a rendszerünkön, ellenőrizzük a `php.ini` fájlban található beállításokat. Ha esetleg nem lenne jogosultságunk a fájl módosítására, a PHP programjaink könyvtárába helyezett saját `php.ini` fájljal felülírhatjuk

az alapbeállítást. Másik lehetőségünk, hogy létrehozunk egy PHPRC nevű környezeti változót, amely kijelöli `php.ini` fájlunkat.

A `php.ini` beállításait bármikor módosíthatjuk, de ha feldolgozunk Apache modulként fut, a változtatások érvénybe léptetéséhez újra kell indítani a webkiszolgálót.

short_open_tag

A `short_open_tag` beállítás határozza meg, hogy használhatjuk-e a rövid `<?>` kód formát a PHP kódblokkok írására. Ha ez ki van kapcsolva, az alábbi sorok valamelyikét láthatjuk:

```
short_open_tag = Off  
short_open_tag = False  
short_open_tag = No
```

Ahhoz, hogy engedélyezzük ezt a beállítást, a következő sorok egyikét kell használnunk:

```
short_open_tag = On  
short_open_tag = True  
short_open_tag = Yes
```

A PHP blokkok kezdő- és záróelemeiről a következő órában lesz szó.

Hibajelentések beállításai

Ha hibákat keresünk programjainkban, hasznos a hibaüzenetek kiírása a HTML oldalba a böngésző számára. Ez alapbeállításban bekapcsolt:

```
display_errors = On
```

Beállíthatjuk a hibajelentési szintet is. Ez azt jelenti, hogy mivel többféle hibaüzenet-típus is rendelkezésre áll, letilthatjuk egyik vagy másik típust, amennyiben nem szeretnénk PHP oldalaink kimenetében látni az abba a csoportba tartozó hibákat. A hibakezelés beállításával alaposabban a huszonkettedik órában foglalkozunk, addig az alábbi értékadás tökéletesen megfelel:

```
error_reporting = E_ALL & ~ E_NOTICE
```

Ezzel a PHP minden hibát jelezni fog a lehetséges problémákat jelölő figyelmeztetések kivételével. Ezek a figyelmeztetések megakadályoznák néhány szokásos PHP módszer alkalmazását, ezért ez az alapbeállítás.

Változókra vonatkozó beállítások

A PHP a GET és POST kérésekből, sütikből, kiszolgálói és környezeti értékekből létrehoz bizonyos változókat. Ennek működését is a `php.ini` fájlban szabályozhatjuk.

A `track_vars` beállítás azt adja meg, hogy létrejöjjenek-e asszociatív tömbök egy HTTP lekérés eredményeként. Ez alapbeállításban engedélyezett, a PHP 4.0.2-es változat óta nem is lehet kikapcsolni:

```
track_vars = On
```

A `register_globals` beállítás azt határozza meg, hogy a HTTP lekéréskor ezek a változók globális változókként jöjjenek-e létre. A PHP fejlődésével egyre inkább azt javasolják a programozóknak, hogy mellőzzék ennek a szolgáltatásnak a használatát, mivel így rendkívül sok változó jöhet létre és ez ütközéseket okozhat, ha nem jól választjuk meg a változók neveit. Ennek ellenére ma a PHP programok legnagyobb része – többek között a könyv számos példája is – arra épít, hogy ez a beállítás be van kapcsolva:

```
register_globals = On
```

Segítség!

A segítség mindig kéznél van az Interneten, különösen a nyílt forráskódú programokkal kapcsolatos problémák esetén. Ezért mielőtt a levelezőprogramunk „Küldés” gombját megnyomnánk, gondolkozzunk el egy kicsit. Akármennyire is működésképtelennek tűnhet a telepített értelmezőnk, beállításunk vagy programozási megoldásunk, jó esélyünk van rá, hogy nem vagyunk ezzel egyedül. Valaki talán már megválaszolta kérdésünket.

Ha falba ütközünk, az első hely, ahol segítséget kereshetünk, a PHP hivatalos honlapja a <http://www.php.net/> címen, különösen az ott található, olvasói kiegészítésekkel ellátott kézikönyv: <http://www.php.net/manual/>. Sok segítséget és információt találhatunk a Zend webhelyén is: <http://www.zend.com/>. A magyar PHP fejlesztők a weblabor.hu webmester honlapon található rengeteg információt: <http://weblabor.hu/php/>. Itt készítjük a magyar PHP dokumentációt is.

Ha nem sikerült megtalálni a megoldást, hasznos segítség lehet, hogy a PHP hivatalos webhelyén keresést is végezhetünk. A tanács, ami után kutatunk, talán egy sajtóközleményben, vagy egy FAQ-fájlban rejtőzik. Egy másik kitűnő forrás a PHP Knowledge Base: <http://www.faqs.com/knowledge-base/index.phtml>. Keresés itt is végezhető.

Még mindig sikertelenek próbálkozásaink? A PHP levelezőlisták kereshető archívumaira mutató hivatkozások megtalálhatóak a <http://www.php.net/support.php> oldalon, számos más hivatkozással együtt. Ezek az archívumok óriási mennyiségű információt tartalmaznak a PHP közösség legjobbjaiktól. Eltölthetünk némi időt pár kulcsszó kipróbálásával.

Amennyiben ezek után is meg vagyunk győződve arról, hogy problémánk még nem merült fel korábban, jó szolgálatot tehetünk a PHP közösségnek, ha felhívjuk rá a figyelmet.

A PHP levelezőlistákra való jelentkezéshez az archívumokat felsoroló oldalon találunk hivatkozásokat. A listák gyakran nagy forgalmúak, de ezt ellensúlyozza, hogy rendkívül sokat lehet belőlük tanulni. Ha érdeklődünk a PHP programozás iránt, legalább egy kötegelte kézbesítésű (digest) listára iratkozzunk fel. A kötegeltség azt jelenti, hogy a listára érkező leveleket nem egyenként, hanem naponta egy-két levélben összefűzve kapjuk meg. Ha sikerül megtalálni az érdeklődési körünknek megfelelő levelezőlistát a számos lehetőség közül, beküldhetjük oda a problémánkat. A PHP honlapján nemzetközi levelezőlisták oldalaira mutató hivatkozásokat is találunk. A magyar PHP levelezőlista és annak archívuma a <http://weblabor.hu/wl-phplista/> címen érhető el.

Mielőtt elküldenénk egy kérdést, gyűjtsük össze a probléma szempontjából fontos információkat, de ne írjunk túl sokat! A következő elemek gyakran szükségesek:

- A használt operációs rendszer
- A telepítés alatt álló vagy futó PHP-változat száma
- A beállításkor használt `configure` paraméterek
- Bármilyen `configure` vagy `make` kimenet, ami előjelezhetette a telepítési hibát
- Egy ésszerűen teljes részlet a kódból, ami a problémát okozza

Miért kell ilyen sok szempontot figyelembe vennünk, mielőtt egy levelezőlistára postáznánk kérdésünket? Először is a fejlesztési problémák megoldásában szerzett tapasztalat előnyös lehet a későbbi munka során. Egy tapasztalt programozó általában gyorsan és hatékonyan tud problémákat megoldani. Egy alapvető kérdés feltevése technikai jellegű listán többnyire egy-két olyan választ eredményez, amelyben felhívják figyelmünket, hogy erre a kérdésre az archívumban megtalálható a válasz.

Másodszor, egy levelezőlista nem hasonlítható össze egy kereskedelmi termék támogatási központjával. Senki sem kap fizetést, hogy megválaszolja kérdéseinket. Ennek ellenére lenyűgöző szellemi erőforráshoz nyújt elérést, beleértve a PHP néhány fejlesztőjét is. A kérdések a válasszal együtt az archívumba kerülnek, hogy később más programozók segítségére lehessenek. Ezért felesleges olyan kérdések feltevése, amelyek már többször szerepeltek a listán.

Ezek megfontolása után ne vonakodjunk kérdéseket küldeni a levelezőlistákra. A PHP fejlesztők civilizált, segítőkész emberek és a probléma felvetésével esetleg másoknak is segíthetünk egy hasonló kérdés megoldásában.

Végül, ha úgy tűnik, hogy a probléma nem a mi kódunkban, hanem a PHP-értelmezőprogramban található, küldjünk egy hibajelentést a fejlesztőknek a `http://bugs.php.net/` címen. Ha a gubanc valóban új, a hibát a PHP következő kiadásában többnyire kijavítják.

Összefoglalás

A PHP 4 nyílt forráskódú. Nyílt abban az értelemben is, hogy nem szükséges egy meghatározott kiszolgálót, operációs rendszert vagy adatbázist használnunk a fejlesztéshez. Ebben az órában láttuk, honnan szerezhető be a PHP és más – a webhelyek szolgáltatásaiban segítő – nyílt forráskódú programok. Megnéztük, hogyan fordítható le a PHP Apache modulként Linux rendszeren. Ha nem a forráskódot töltjük le a hálózatról, hanem egy lefordított változatot, akkor a csomagban részletes információkat találunk a telepítéssel kapcsolatban. Áttekintettünk néhány `configure` paramétert, melyekkel a feldolgozó képességeit befolyásolhatjuk. Tanultunk a `php.ini` fájlról és néhány beállítási lehetőségről, amit megadhatunk benne. Végül megnéztük, hol találhatunk segítséget, ha problémáink akadnak. Ezzel készen állunk arra, hogy megbirkózzunk a PHP nyelvel.

Kérdések és válaszok

A telepítést Apache webkiszolgálót használó Linux operációs rendszeren vezettük végig. Ez azt jelenti, hogy a könyv nem megfelelő más rendszer vagy kiszolgálóprogram használata esetén?

A PHP egyik nagy erőssége, hogy több rendszeren is képes futni. Ha problémák adódnak az adott rendszeren a PHP telepítésével, olvassuk el a csomagban található dokumentációt és a PHP kézikönyv megfelelő fejezetét. Általában széles körű lépésről-lépésre leírt utasításokat kapunk a telepítéshez. Ha továbbra sem sikerül megoldani a problémát, a „Segítség!” című részben ismertetett módszerek célravezetőek lehetnek.

Hogyan érhetők el böngészőből a PHP állományok, ha a gépre telepítettük a webkiszolgálót?

A PHP alkalmazások fejlesztéséhez és teszteléséhez nem szükséges, hogy számítógépünk az Internetre legyen kapcsolva, bár az éles környezetben való ellenőrzés hasznosabb lehet. Bármilyen operációs rendszert is használunk, akár hálózatba kötött gépen dolgozunk, akár nem, a saját gépünk IP-címe `127.0.0.1`, neve akkor is `localhost` lesz. Ezért ha a saját gépünkön lévő webkiszolgáló gyökérkönyvtárában lévő `elso.php` fájlt szeretnénk megnyitni, a `http://localhost/elso.php` címen érhetjük el. Windows operációs rendszeren feltétlenül telepíteni kell a TCP/IP támogatást, hogy ez a lehetőség rendelkezésre álljon.

Műhely

A műhelyben kvízkérdések találhatók, melyek segítenek megszilárdítani az órában szerzett tudást. A válaszokat az A függelékben helyeztük el.

Kvíz

1. Hol érhető el a PHP kézikönyv?
2. UNIX rendszeren hogyan kaphatunk bővebb információkat a beállítási lehetőségekről (milyen paramétereket kell átadni a `configure`-nak)?
3. Hogy hívják általában az Apache beállításait tartalmazó fájlt?
4. Milyen sort kell az Apache beállítófájlhoz adni, hogy a kiszolgáló felismerje a `.php` kiterjesztést?
5. Mi a PHP beállításait tartalmazó fájl neve?

Feladatok

1. Telepítsük a PHP-t. Ha sikerült, nézzük át a PHP beállítófájlt és ellenőrizzük a telepítés helyességét.