

Minőségi dokumentáció egyszerűen phpDocumentor

Az interneten kering egy leírás az „igazi programozóról”, amely ugyan az informatikus humor becses darabja, de mint minden viccben, ebben is van némi komoly tartalom. Mint a cikkben idézet részletből is látható, a dokumentációírás sem az igazi, sem a botcsinálta programozóknak nem kedvenc elfoglaltsága, akármilyen más híresztelést halljunk is.

Az igazi programozó...

A tipikus „igazi programozó” élettere termináljának közvetlen közelére terjed ki. Ezen terminál körül a következők találhatók:

1. Az összes program listája, amelyen az igazi programozó valaha is dolgozott, nagyjából időrendi sorrendben, elfoglalva a szoba összes sík felületét.
2. Vagy fél tucat félig üres csésze hideg kávé. Esetleg csikkek is úszhatnak a kávéban, ha nem lehetett megtalálni a listák között a hamutartót. Néha a csészékben narancslé van.
3. Hacsak nem különlegesen jó, *OS JCL* kézikönyvek és a *Principles of Operation*, néhány érdekesebb részletnél kinyitva.
4. A falon egy sornyomtatóval készült naptár (az 1969-es évre).
5. A padlón szétszórva csokipapírok.
6. A fiók alján egy folyamatábra-rajzoló sablon, amit az előző lakó hagyott itt. (Az igazi programozó programokat ír, nem pedig dokumentációt.)

Forrás: ☞ www.cab.u-szeged.hu/local/doc/UNIX/orlando/igazi.html

Miért készítsünk dokumentációt?

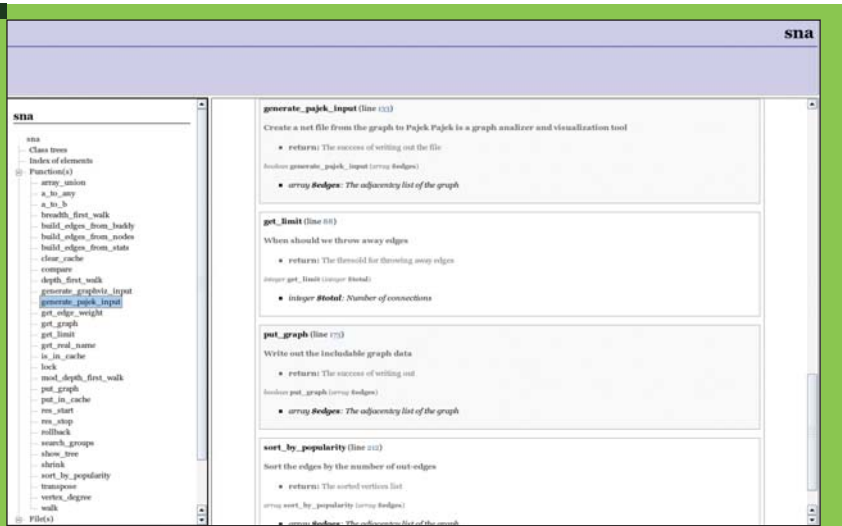
A keretes írásunk zárójeles megjegyzése adja fonák módon cikkünk mottóját. Ugyan manapság talán a hátlagombolós programozók fiókja alján sem találunk folyamatábra-rajzoló sablont, a jó minőségű és részletes dokumentáció elkészítése semmivel sem vált egyszerűbbé. Talán a dokumentációkészítő programok korunk sablonjai. Ha most arra gondolunk, hogy akkor hátlagombolós programozókká degradáljuk magunkat efféle eszközökkel, akkor fontoljuk meg a következőket: „Az igazán dögös programozó hexa számológép nélkül találja meg a hat megabájtos dumpban

a hibát.”. Ha még nem értük el a programozói törzsféjlődés ezen evolúciós lépcsőfokát, akkor inkább legyünk korrekt hátlagombolósok, mintsem csapnivalóak. A *PHP* nyelv nagy népszerűsége tette szert a *LAMP – Linux/Apache/Mysql/PHP* összeállításnak köszönhetően, hisz *LAMP* felállásban működik számtalan webszerver a világhálón. A *PHP* alkalmas arra is, hogy minimális tudással egyszer használatos kódot írjunk szükségmegoldás gyanánt, de olyan népszerű és nagy projektek is ezt használják, mint a *Drupal*, a *PhpMyAdmin* vagy a *Gallery*. Egy nyílt forráskódú programnál

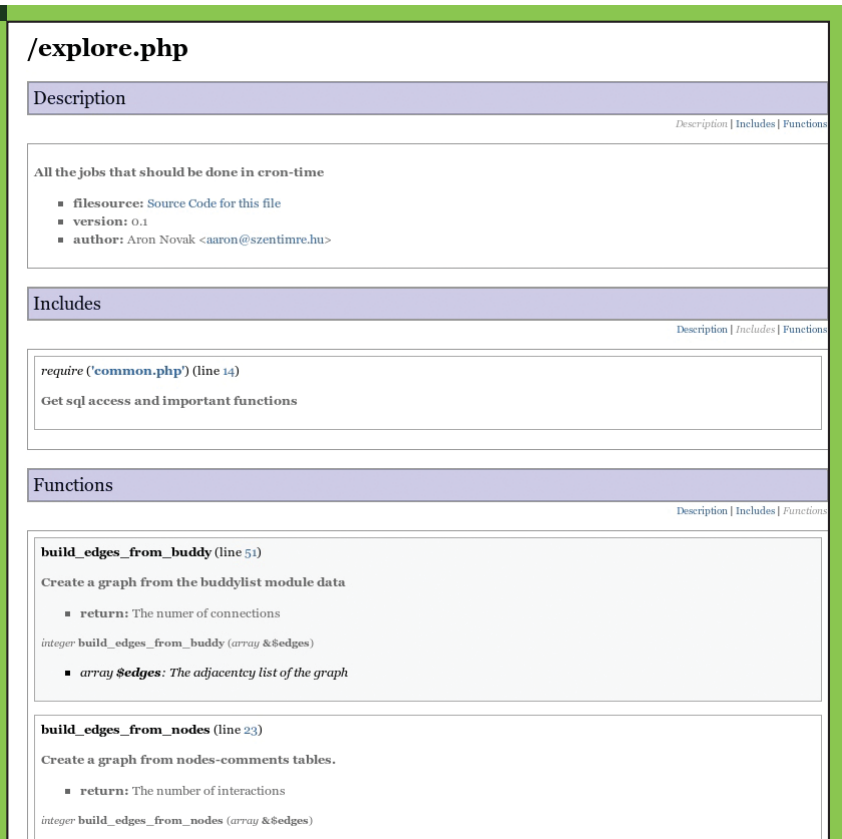
ahhoz, hogy létrejöjjön egy fejlesztőgárda, a dokumentáció egy elengedhetetlen kellék. Hiszen kinek van kedve mindent az alapoktól kezdve a forráskódból kibogarászni? Egy jól megírt dokumentáció nagyban felgyorsítja a beletanulási időszakot. Így már az újonc is képes használható kóddal hozzájárulni a kódbázishoz. Először is kinek szól a dokumentáció? A fejlesztőknek és a felhasználóknak egyaránt, bár a két célcsoportnak különböző mélységű és stílusú leírásokra van szüksége. A fejlesztőknek szóló részét természetesen maguk a fejlesztők írják, a felhasználóknak szólót már nem biztos, hogy célszerű a programozóknak írnia, hiszen ott nagyon fontos, hogy a felhasználó nyelvén szóljon a leírás.

Az eszköz

A *phpDocumentor* egy *PHP* nyelven írt alkalmazás, melyet *GPL* alatt terjesztettek. A projekt honlapjáról (☞ www.phpdoc.org) letölthető a legfrissebb változat, ez a cikk írásának pillanatában az *1.3.0RC6* verziót takarta. A *phpDocumentor* kétféle üzemben is képes működni. Egyfelől hagyományos parancssori alkalmazás, amelyet kapcsolókkal vezérelünk, másfelől egy webes alkalmazás, ahol pár kattintással összeállíthatjuk, mit szeretnénk kérni. Itt az első változatot fogjuk áttekinteni, több okból is. Egyrészt látni fogjuk, hogy a munka kisebbik része magának



1. ábra Ez történik a függvényekkel feldolgozás után



2. ábra Minden forrásfájlról készül összefoglaló

a *phpDocumentornak* a használata, az előkészületek sokkal több energiát igényelnek, másrészt a webes felülettel semmivel nem járunk előrébb, viszont kell hozzá egy webszerver, ami *PHP* szkriptek írásához nem feltétlenül. Amennyiben a második lehetőség jobban érdekel minket, csupán annyi a teendőnk, hogy a letöltött *.tgz* fájlt nem akárhová, hanem a webszerver

könyvtára alá bontjuk ki és máris használatba vehetjük az intuitív kezelőfelületet.

Előkészületek

A *phpDocumentor* ezen verziójába még nem építettek bele magas fokú mesterséges intelligenciát, mely tetszőleges forráskódot képes hasznos leírással ellátni. Mivel a közeljövőben

erre nem is nagyon számíthatunk, érdemes elsajátítani egy speciális kommentezési technikát, melyet számtalan dokumentációkészítő eszköz képes értelmezni. Például a *Doxygen* vagy a *Javadoc* is, így *C++*-hoz és *Javához* is részben alkalmasak az itt leírtak. A *PHP* többféle megjegyzésmódot támogat, *C*, *C++* és *Perl* stílusban is kommentezhetünk. A *phpDocumentor* azt várja, hogy a forráskódunkat úgynevezett *DocBlock* rendszerben lássuk el megjegyzésekkel, amiből már lehet automatikusan generálni a dokumentációt. Egy már meglévő programnál nagy munka ezekkel kiegészíteni a kódot, érdemes már a kezdetektől használni ezt a stílust. Így jelezzük, hogy *DocBlockot* írunk:

```
/**
 *
 */
```

Ez a kerete a blokknak, amit aztán meg kell tölteni értelmes tartalommal. Láthatjuk, hogy ez a forma a *C* stílusú megjegyzés kiegészítésének tekinthető. Ezeket a blokkokat bárhol elhelyezhetjük a kódban, célszerű konszantsok, függvények és osztályok előtt írni egy blokkot, amiben részletezzük az adott elem szerepét. Minden ilyen *DocBlock* három részből tevődik össze ez alapján:

```
/**
 * Rövid leírás
 *
 * Hosszabb leírás
 *
 * Címkék
 */
```

A rövid és hosszabb leírást azt minden köztéttség nélkül leírhatjuk, például egy függvény előtt beleírhatjuk, hogy milyen algoritmust használ, vagy amit épp fontosnak érzünk. A címkéket már egy között formában kell megadni. Ha az aktuális blokk a fájlra magára vonatkozik, akkor e minta alapján készítsük el a megjegyzést a fájl elejére:

```
/**
 * Rövid leírásokra
 *
 * Hosszú leírás
```

Source for file explore.php

Documentation is available at explore.php

```

1. <?php
2. // $Id$
3. /**
4.  * All the jobs that should be done in cron-time
5.  *
6.  * @author Aron Novak <aaron@sztimre.hu>
7.  * @version 0.1
8.  * @package sna
9.  */
10.
11. /**
12.  * Get sql access and important functions
13.  */
14. require 'common.php';
15.
16.
17. /**
18.  * Create a graph from nodes-comments tables.
19.  *
20.  * @param array $edges The adjacency list of the graph
21.  * @return integer The number of interactions
22.  */
23. function build_edges_from_nodes(&$edges) {
24.
```

3. ábra A dokumentációban lévő forráskód

```

*
* @author Szerző neve
* @version Verziószám
* @package Csomagnév
*/
```

Ha pedig egy függvényről van éppen szó, ott nyilván a paraméterek és a visszatérési értékek a fontosak:

```

/**
 * Rövid leírás
 *
 * Hosszú leírás
 *
 * @param integer $param1 Mi ez
 *   a paraméter?
 * @param boolean $param2 Mi ez
 *   a paraméter?
 * @return integer Mit adunk
 *   vissza?
 */
```

Fontos megjegyezni, hogy a leírások és a címkék opcionálisak, készíthetünk blokkot úgy, hogy csak ez egyiket-másikat írjuk bele. Ha mindezekkel elláttuk a *PHP* forráskódot, ami nem kis munka egy nagyobb lélegzetvételű programnál, sőt még értelmes leírásokra is futotta az energiánkból, akkor a *phpDocumentor*on már nem fog múlni a jó minőségű dokumentáció!

Parancssoros használat

Talán ahhoz vagyunk szokva, hogy a *PHP* programoknak webalapúnak kell lennie, ez koránt sincs így. A *PHP 5*-ben már kimondottan a parancssort támogató új lehetőségek jelennek meg. A *phpDocumentor* jelen állapotában egy héjprogramot használ előtétként. Hogyan is használjuk ezt az előtétet?

Ehhez először is tudni kell, hogy mit tud kezdeni egy blokkokkal felokosított forráskóddal a program. Tud belőle számtalan formátumban dokumentációt készíteni (*HTML*, *PDF*, *CHM* és így tovább). A kimenetet lehetőségünk van tetszés szerinti megjelenéssel felruházni, készíthetünk sablonokat, így a projekt arculatához szabhatjuk a dokumentációt. Most mi először a kiegészített forráskódból legyártott dokumentálást – az alapvető működést vesszük szemügyre. Először is gondoskodjunk róla a kényelem érdekében, hogy a kitömörített *phpDocumentor* könyvtár legyen benne az elérési útban és adjunk futtatási jogot a *phpdoc* héjprogramra, mert a kitömörítés után nincs rajta *x* jog. Lépünk be a forráskód könyvtárba és ott adjuk ki az alábbi parancsot:

```
phpdoc --directory . --output
↳ /dokumentacio-utvonala/
```

Ez már elég a programnak, hogy az alapbeállításokkal elkészítse a *HTML* formátumú dokumentációt a megadott könyvtárba. A *phpdoc*, mint minden tisztességes parancssoros program, kilistázza a kapcsolói listáját, egy érdekességre felhívnom a figyelmet. Ha a `--sourcecode` kapcsolót megadjuk, akkor minden forrásfájlból kapunk egy szépen kiszínezett, sorszámozott verziót. Ennek több előnye is van, egyfelől a weben a látogatók ezt tudják böngészni, nagyon látványos. Másrészt pedig emlékezzünk a keretes írásra, a nyomtatott programlista nem öncélú! A forráskód kinyomtatásához nincs más teendőnk, mint generálni egy dokumentációt a megfelelő kapcsolóval és a webböngészőből elindítani a nyomtatást. Az eredményt a 3. ábrán láthatjuk.

DocBlocks – egy univerzális eszköz

Fontos látni, hogy ugyan a cikk a *phpDocumentor* alcímet viseli, viszont ezt a dokumentálási technikát számtalan más helyen fel lehet használni, így a *DocBlocks* használatával megbarátkozni biztosan hasznos tevékenység. Indulásnak a legjobb a projekt honlapján megtalálható példafájlokat megnézni, ott szépen látszik a technika alkalmazása:

☞ manual.phpdoc.org/HTMLSmartyConverter/HandS/phpDocumentor/tutorial_sample3.pkg.html. A felsoroltakon kívül számtalan módon lehet használni a *phpDocumentor*t, például lehetőség van felhasználói dokumentációt is létrehozni. Az nem is kérdés, hogy egy ilyen szoftver dokumentációját mivel készítik. Érdeemes megnézni a ☞ manual.phpdoc.org címet, a legjobb referencia ennek a szoftvernek!



Novák Áron

(aaron@sztimre.hu)

BME-VIK-es hallgató, műkedvelő rendszergazda. Jelenleg legin-

kább a NetBeans-szel és mindenféle hordozható eszközzel foglalkozik, legáltalában mindazokkal, amelyeket meg lehet szólaltatni Linux alatt.