

# Hogyan maximalizálja értékesítési szövegei hatékonyságát?

By MarketingSzoftverek

2005.06.01.

Skycentrum Bt.  
1089 Budapest, Golgota utca 3.

Copyright © 2005, Skycentrum Bt. Minden jog fenntartva

## TARTALOMJEGYZÉK

<b>Hogyan maximalizálja weboldala értékesítési hatékonyságát? .....</b>	<b>3</b>
Találgatás helyett tesztelés .....	3
Mit teszteljen .....	4
A tesztelés alapmódszerei .....	4
Sokszoros hatékonyság .....	7
A tesztelés általános alapelvei.....	7
Hogyan végezze a Párhuzamos tesztelést?.....	8
Meddig futtassa a tesztet? .....	9
Összefoglalás.....	11
1. sz. melléklet - technikai útmutató és kódrészletek az alkalmazáshoz.....	12

## Hogyan maximalizálja weboldala értékesítési hatékonyságát?

### Találgatás helyett tesztelés

Az Internet-marketing és a hagyományos marketing közti egyik legnagyobb különbség, hogy az Internet-marketingben jóval több minden mérhető. A mérési eredmények azonban önmagukban a legtöbb esetben nem adnak elegendő információt ahhoz, hogy a soron következő marketing akcióval kapcsolatos döntéseket meg tudja hozni.

A tesztelés lényege, hogy segítségével több, azonos célú, de különböző módon kivitelezett marketing elemet - oldalcím, szöveg, link, stb. - összehasonlíthat annak érdekében, hogy időben kiderüljön: az „éles” verzióban melyik lesz az eredményesebb.

Mivel az Interneten zajló kampányok költsége a hagyományos kampányokéhoz képest elenyésző, ráadásul az oldalelemeken történő változtatások költsége az Internet sajátosságainak köszönhetően szintén nagyon alacsony, Önnek szinte semmibe nem kerül, hogy a tesztelés segítségével kizárja az összes bizonytalanságot, és csak azokkal a marketing elemekkel dolgozzon, amelyek maximális eredményt hoznak.

## Mit teszteljen

Gyakorlatilag bármit tesztelhet, ami kihat a weboldal eredményeire. Tesztelheti hirdetéseinek címét, szövegét, tesztelhet design elemeket, tesztelheti értékesítési szövegeinek egyes elemeit vagy az egész értékesítési szöveget, linkeket, menü-megoldásokat.

## A tesztelés alapszerei

A tesztelés két alapszere az egyszerű tesztelés és a párhuzamos tesztelés.

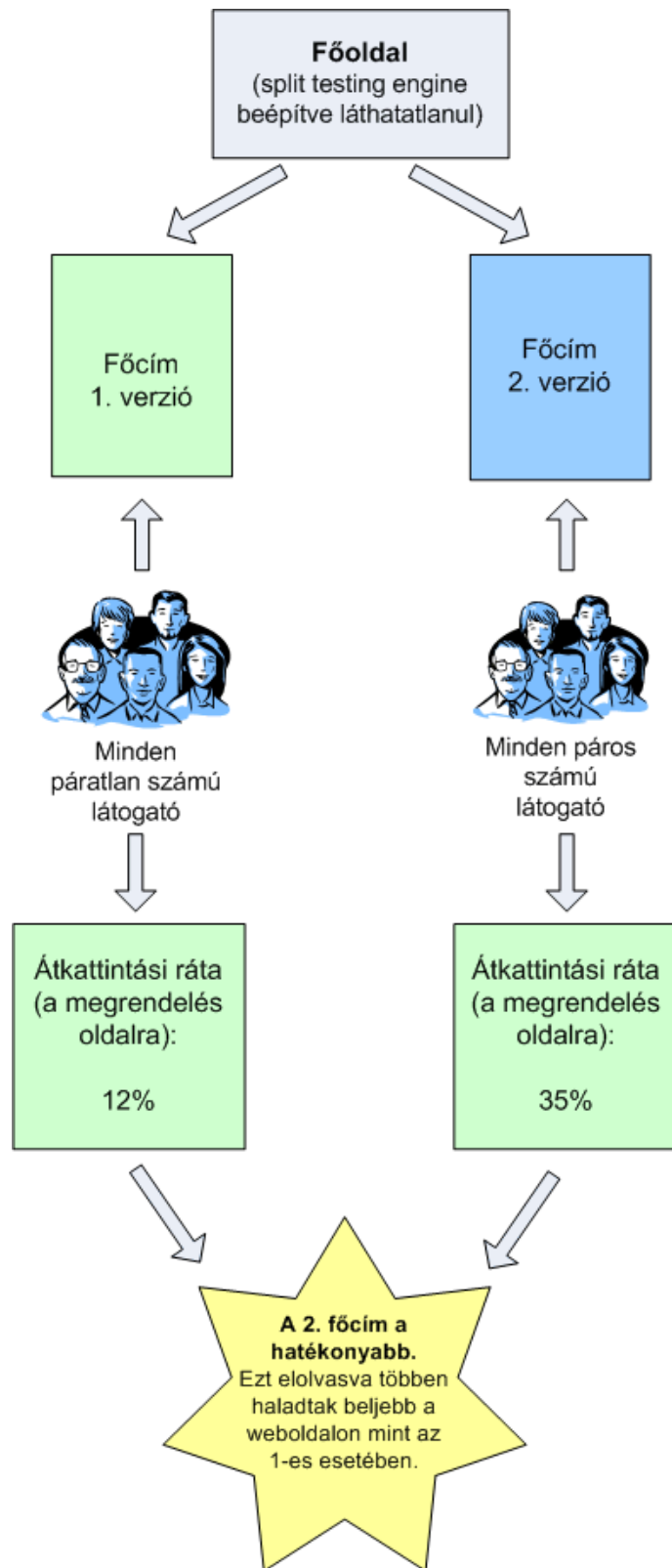
Az egyszerű tesztelés során időben egymás után történik a két lehetséges verzió tesztelése, a párhuzamos tesztelésnél egyidőben. Az egyszerű tesztelés hátránya az, hogy mivel egymás után történik a két különböző verzió vizsgálata, az eredményben mutatkozó különbségeket nem csak a verziók közti különbségek, hanem a tesztelés időpontjára jellemző felhasználói viselkedés is okozhatja.

A párhuzamos tesztelés során a két tesztelendő verzió egyazon időszakban fut, kiküszöbölve ezáltal az időszakok közti lehetséges különbségeket.

A módszer alkalmazása során az értékesítési folyamat egy elemének (cím, alcím, link, design elem, stb.) két vagy több verziója kerül összehasonlításra úgy, hogy minden páratlan sorszámú látogató a tesztelni kívánt elem egyik, minden páros számú látogató a másik változatával találkozik. A tesztelést követően azt kell megvizsgálni, hogy az egyik illetve a másik verzió esetén a látogatók hány százaléka jut el az Ön által megtervezett cselekvésig, mint pl. vásárlás, hírlevél feliratkozás, stb.

A módszert alkalmazva **Ön is könnyedén ki fogja tudni választani a lehetséges weboldal szövegek - címek, alcímek, link szövegek, leírások, stb. - közül azokat, amelyek a maximális eredményt hozzák,** azaz amelyeknél minden más verzió csak rosszabb lenne.

Az alábbi ábra szemlélteti, hogy mi történik a párhuzamos tesztelés során:



Az ábrán egyértelműen látszik, hogy a főcím 2. verziója bizonyult hatékonyabbnak a mérések során. Így a mérést követően a tesztelt verziók közül Önnek csak ezt kell megtartania. A teszteket addig érdemes folytatni, amíg el nem jut egy olyan változatig, amelynél egy következő tesztben minden más verzió rosszabbul teljesít.

Lássunk egy másik példát:

*„Egy online marketing tanácsadóval foglalkozó cég Angliában minden hónapban nagyon nagy számú megrendelést regisztrál. A sikeres működésük egyik alapvető oka tesztelési módszerüknek tulajdonítható. Ha például szeretnének felvenni kínálatukba egy új szolgáltatást, akkor először készítene kétféle weboldal verziót ugyanarról a témáról. Lefuttatnak egy tesztkampányt, mely addig tart, amíg összességében 1000 látogató nem járt az új oldalon. A párhuzamos tesztelési módszernek köszönhetően egy automatizmus megoldja, hogy pontosan 500-500 látogató lássa a két különböző oldal verziót. Az eredmény kiértékelésénél eldől, hogy melyik verzió teljesített jobban, azaz honnan kattintott át több látogató a megrendelés oldalra. A tesztkampány kiértékelése után kezdik el élesben futtatni a jobb teljesítményt nyújtó oldalt. Ráadásul a tesztelést az éles verziónál is mindaddig újra és újra megismétlik, amíg van ötlet a tarsolyukban az oldal javítására. Nem minden ötlet bizonyul jobbnak, mint az előző, de sokszor sikerül javítani az eredményességet. Ezzel a módszerrel eléri, hogy kínálatuk egyre közelebb álljon célközönségük igényeihez, ami egyértelműen a megrendelések számának növekedését is jelenti.”*

A Párhuzamos tesztelés (split testing) egy olyan módszer, **amellyel jól el lehet különíteni és tesztelni azokat a tényezőket, amelyek az online marketing tevékenység eredményeire hatással vannak.** Ezt a módszert manapság is rendszeresen használják a direkt marketingben, amikor a kiküldött boríték nagyságától, a cím szövegének a színéig mindent letesztelnek. Mindenhol gazdaságosan és eredményesen alkalmazható a módszer, ahol (1) a különböző verziók elkészítése nem emeli jelentősen a költségeket (pl. direkt marketingben a boríték nyomdaköltsége szemben egy TV reklám elkészítésének költségével), illetve ahol (2) a két verzióból bejövő megrendelések megkülönböztethetők (pl. direkt marketingben két különböző megrendelőszelvény alkalmazása).

## Sokszoros hatékonyság

Ha online marketing tevékenységének fő elemeit a párhuzamos tesztelés módszerével teszteli, a következő előnyökre tesz szert:

1. Megtalálhatja az értékesítésben részt vevő összes alkotóelem maximálisan hatékony változatát. Így értékesítési folyamatának hatékonysága a sokszorosára nőhet.
2. Amennyiben a tesztelést elvégzi online kampányai előtt - kisebb tesztkampányokon - az éles kampányok hatékonysága is a sokszorosára nő az eddigiekhez képest.
3. Hagyományos marketing tevékenységének hatékonyságát is megsokszorozhatja, ha marketing szövegeit még a használat előtt a párhuzamos tesztelés módszerével tökéletesíti.

## A tesztelés általános alapelvei

Lássuk a két legfontosabb szabályt, ami gyakran elkerüli a figyelmet az online marketing tesztek során.

### 1. Nem csak a reklámon múlnak az eredmények

Gondoljon olyan kampányban amelynek két oldala van. Az egyik a reklám oldal, a másik a reakció oldal. Az első alapszabály, hogy **ki kell terjeszteni a párhuzamos tesztelés alkalmazási területét az egész weboldalra**. Még azok közül is, akik tesztelnek, sokan csak a kreatív tényezőket tesztelik, úgymint értékesítési üzenetek, ajánlatok, design elemek, banner-ek, e-mailek vagy fizetett hirdetések. Elhanyagolják azonban a kampány bejövő csatornáit, azaz a főoldalt vagy fogadó oldalt, illetve az egész site-ot, egészen a megrendelést felvevő oldalig. Így viszont bármilyen hatékonyságot ér el reklám oldalon, a site elveszíti a lehetséges vásárlókat. Ha tehát maximális hatékonyságot szeretne, a teljes site minden egyes pontjának a hatékonyságát maximalizálnia kell!

### 2. Egyszerre csak egy dolgot teszteljen

A következő alapelvet soha ne veszítse szem elől: egyszerre csak egy dolgot teszteljen. Ha ugyanis egyszerre több olyan dolgot is tesztel, ami kihat az eredményekre, nem fogja

tudni, hogy melyik változtatás hozta az eredmények változását. Ha például egyszerre teszteli főoldalának címét, szövegét és elrendezését, előfordulhat, hogy az egyik változaton jobb a cím, a másikon a szöveg, viszont az eredmény a két verzió esetén közel azonos. Ilyen esetben nem fog kiderülni, hogy melyik változattól mit érdemes megtartani.

## Hogyan végezze a Párhuzamos tesztelést?

Ha eldöntötte, hogy elindít egy tesztet, először is el kell készítenie a különböző kreatívokat, amelyeket a teszt segítségével össze szeretne hasonlítani. Ha például értékesítési oldalának főcímét szeretné tesztelni, készítse el az oldal két verzióját - az egyiket az egyik címmel, a másikon a másik címmel.



Példa: A MarketingSzoftverek korábbi két nyitólap verzióján csupán az oldal alcíme volt a különbség. Mégis az derült ki a mérések alapján, hogy az egyik esetben kb. 30%-kal jobbák az eredmények!

Ezek után van szükség arra a kis programra, amelyik elosztja az oldal látogatóit az oldal két változata közt. (A mellékletben található programkód felhasználásával Ön vagy webmestere is pillanatok alatt el tudják indítani saját tesztjüket. Amennyiben azonban a [MarketingSzoftverek Exkluzív tárhely](#) szolgáltatását használja - elkezdni használni -, munkatársaink gyorsan, szakszerűen és DÍJMENTESEN beindítják első párhuzamos tesztjét!)

A tesztelést természetesen a látogatók nem érzékelik, minden későbbi látogatásukkor is ugyanazt a verziót látják, amelyiket először látták (Technikai részletek a mellékletben.)



Ezután a módszer kivitelezéséhez már csak egy dolog hiányzik: egy látogatottságmérő szoftver, amely megmutatja, hogy az egyik vagy a másik verzió esetén merre mennek tovább a látogatók. Erre tökéletesen alkalmas az általunk kifejlesztett WebMonitor szoftver ( [www.marketingszoftverek.hu/webmonitor](http://www.marketingszoftverek.hu/webmonitor) ), de több külföldi eszköz közül is választhat.

## **Meddig futtassa a tesztet?**

**Hány látogató szükséges ahhoz, hogy nagy biztonsággal el tudja dönteni, melyik verzió a jobb? Honnan tudhatja, hogy elegendő adattal rendelkezik a döntéshez?**

A szakértők is vitatkoznak hogy van-e precíz képlet rá, de egy dologban egyet értenek: **minél kevesebb tesztelendő tényező van, annál kisebb látogatói szám szükséges hozzá.**

Sok szakértő szerint egyetlen változó tesztelésénél egy 25 főből álló mintacsoport már 95%-os megbízhatóságú eredményt produkál, azaz 95% esély van arra, hogy a mintacsoport eredménye megegyezik az összes látogatótól várható eredménnyel.

Néhány tényezőt azonban sokkal könnyebb kontrollálni, mint más típusúakat. Sokkal kevesebb ellenőrzés szükséges, ha nagyobb mintánk van, mert sokkal könnyebben elérjük a magas megbízhatósági rátát.

Nagyon könnyű például szintesztet végezni, ha a weboldal leendő hátterének a design-ja a kérdés, mert majdnem mindenki egyformán érzékeli a színeket.

Sokkal nehezebb viszont kontrollálni az értékekkel kapcsolatos állítások igaz voltát, mert az emberek szubjektíven értelmeznek dolgokat. Ilyenek például a „takarítson meg...” és a „most bevezető áron...” típusú szlogenek.

**Létezik azonban egy megbízható módszer annak eldöntésére, hogy meddig kell még folytatnia a tesztelést (használja a mellékletben található 2. kódrészt):** Ebben az esetben a program a látogatók felének a tesztelt elem „A” verzióját jeleníti meg, a másik felét pedig még két részre osztja. Az első 25% a „B1” a másik 25% a „B2” verziót látja az oldalon. A módszer lényege az, hogy mivel a „B1” és a „B2” verzió tökéletesen megegyezik egymással, ha elég ideig végzi a tesztet - tehát ha elég nagy minta áll rendelkezésre -, a „B1” és „B2” verzióknak ugyanazt az eredményt kell produkálniuk.

A mérés befejezése után két dolgot kell megvizsgálnia:

1. Az „A” vagy a „B” („B1” + „B2”) verzió eredményei a jobbak. Amelyik verzió eredményei jobbak, azt a verziót kell a jövőben használni.
2. Hány százalék az eltérés a „B1” és „B2” verzió eredményei közt. Ez a szám ugyanis éppen a mérési pontatlanság. Ha ugyanis a mérés tökéletes lenne, „B1” és „B2” azonos lenne, tehát az eltérés - azaz a mérési pontatlanság - 0% lenne.

A döntésnél a következő az alapelv: Abban az esetben döntsön „A” vagy „B” verzió mellett, ha a két verzió közti eltérés jelentősen nagyobb (személy szerint azt mondanám, hogy legalább ötszörös), mint a mérési pontatlanság. Ha pl. „A” és „B” eredményei közt 40% különbség van, akkor mindez elég ok a döntéshez, ha „B1” és B2” között nincs több, mint 8-10% különbség. Ha azonban ez utóbbi szám pl. 35%, akkor még túl nagy a mérési pontatlanság, tehát érdemes tovább folytatni a mérést.

## Összefoglalás

**Bátran állíthatjuk, hogy a párhuzamos tesztelés a leghatékonyabb marketing fegyver ami manapság elérhető az online marketing piacon.**

- Lehetővé teszi, hogy weboldala minden fontos részét sorban letesztelje: főcímet, szöveget, árakat, grafikákat, az egész design-t egyben, garanciákat, színeket , stb.
- Elmondja Önnek, hogy egyazon elem két tesztelt verziója közül, melyiknek van nagyobb hatása a látogatók vásárlási döntésére.
- Megspórolhatja mindazt az időt és fáradságot, amit azzal töltene, hogy „öszöneire” hallgatva próbálja számokba önteni, megbecsülni a várható eredményeket.
- Hihetetlen versenyelőnyt ad az Ön kezébe a módszer, hiszen jószerivel senki nem használja.
- Megelőzi a találgatást - világosan megmutatja mi működik és mi nem.
- Diszkréten működik - az Ön oldalára látogatók közül senkit nem zavar össze a teszteléssel.

**„Ez nem egy extra lehetőség, ez kell, hogy az alapszere legyen annak, aki internetes értékesítéssel foglalkozik.,,**

## 1. sz. melléklet - technikai útmutató és kódrészletek az alkalmazáshoz

Az alábbiak segítségével a weblap karbantartója üzembe tudja helyezni a párhuzamos tesztelést. Amennyiben Ön saját maga tartja karban a weboldalt, ez a rész Önnek szól. Amennyiben másvalaki végzi ezt az Ön számára, juttassa el neki ezt a dokumentációt.

Ha Ön a MarketingSzoftverek [Exkluzív Tárhelyszolgáltatását](#) használja, az Ön számára díjmentesen elvégezzük első párhuzamos tesztjének elindítását <sup>1,2</sup>.

1 : A tesztelendő oldal két verzióját Önnek kell elkészítenie

2 : Amennyiben nem a MarketingSzoftverek Exkluzív Tárhelyszolgáltatását használja, Önnek kell gondoskodnia róla, hogy PHP 4.xx futtatható legyen a webserveren

### Technikai megvalósítás feltételei

A párhuzamos tesztelés technikai megvalósításakor az alábbi két tényezőre kell különösen figyelmet fordítani:

1. A látogatók 50%-a lássa a weboldal egyik és 50%-a a másik verzióját.
2. A visszatérő látogatók ugyanazt a weboldalt lássák, mint amit az első látogatásukkor.

Az első tényező egyszerűen kezelhető egy megfelelő eloszlású véletlenszámokat előállító függvénnyel. Elegendően nagy szám minta esetén a véletlenszámok eloszlása egyenletes lesz.

A visszatérő látogatók felismerésének legbiztosabb módszere a cookie alapján történő azonosítás. Némely böngésző az alapértelmezett beállítás szerint elutasítja a nem megfelelő cookie-kat, de sok látogató maga is beállítja bizonyos cookie-k elutasítását.

A cookie elutasításának kivédésére két technológiát alkalmazunk:

1. Első kézből származó cookie (1st party cookie), ami azt jelenti, hogy a cookie-t az ön weboldal helyezi el a látogató böngészőjében. Így, ha a látogató le is tiltotta a harmadik féltől származó cookie-k (3rd party cookie) elfogadását, a párhuzamos tesztelés által használt cookie akkor is elhelyezésre kerül a látogatónál.

2. Tanúsítvánnyal rendelkező cookie használata. Az Internet Explorer 6-os verzióját használó látogatók így a böngésző alapbeállításai mellett, biztosan elfogadják a cookie-t.

A cookie törlését ezen a szinten nem lehet kivédeni. Ezt a kérdést a felhasználók felvilágosítása illetve a következetes, egyértelmű kommunikáció oldhatja meg.

A cookie-k problémakörének részletesebb ismertetése meghaladja e tanulmány kereteit, erről a témáról egy későbbi cikkünkben fogunk bővebben írni.

A párhuzamos tesztelés hatékony megvalósításához az Internet legelterjedtebb szkript nyelvét a [PHP](#)-t használjuk. A PHP használatával egyszerűen megvalósíthatók a fent említett követelmények.

Fontos szempont továbbá az is, hogy a programlogika teljesen elkülönüljön a megjelenítéstől. Így változtatáskor csak a HTML fájlt kell változtatni, a programkódhoz nem szükséges hozzányúlni.

Ennek megvalósításához kétféle módszert mutatunk be.

Az egyik a sablonkezelő rendszerrel történő megoldás, amelyhez a [Smarty](#) sablonkezelő rendszert használjuk.

Azért tartjuk fontosnak e módszer alkalmazásának bemutatását a párhuzamos tesztelésben, mert tapasztalatunk szerint a hatékony online marketinghez elengedhetetlen a gyors és költséghatékony weboldal karbantartás megoldása. A Smarty Template Engine lehetővé teszi, hogy a mind a gyakori, apró változtatásokat, mind a folyamatos nagyobb fejlesztéseket hatékonyan végezzük el. Azon ügyfeleinknél, akiknél ezt a rendszert használjuk, jóval alacsonyabb költségek mellett tudjuk karban tartani oldalait, illetve végezni a folyamatos fejlesztéseket, mintha nem használnánk a rendszert.

A másik megoldás egyszerű fájlkezelést használ a megfelelő verzió megjelenítésének kezelésére. Amennyiben Ön nem használ sablonkezelő rendszert, akkor válassza ezt a megoldást.

## Párhuzamos tesztelés módszerei

A párhuzamos tesztelésnek két módszere van, egy **egyszerűbb**, amely csak az A és B verzió megfelelő megjelenítését kezeli, illetve egy **összetettebb**, amely alkalmas a mérési pontatlanság kimutatására, ezáltal annak eldöntésére, hogy kell-e még tovább folytatni a tesztelést. Az alábbiakban mindkét verzió forráskódját közzétesszük.

### Az egyszerű módszer Smarty sablonkezelő rendszer használatával

A párhuzamos tesztelést végző kód két jól elkülöníthető részből áll:

1. PHP kód, amely a program logikáját tartalmazza. Itt kerül eldöntésre, hogy a lehetséges verziók közül melyik jelenjen meg az adott látogatónak.
2. Tesztelendő verziók HTML fájljai. Ezek közül kerül kiválasztásra a megjelenítendő verzió.

Az egyszerű módszer PHP forráskódja Smarty sablonkezelő rendszer használatával:

```
<?php
require 'libs/Smarty.class.php';

/* Header */
header("P3P: CP='ALL DSP COR LAW CUR ADMi DEVi TAIi PSDi IVDi CONi
OUR BUS PHY ONL PUR FIN COM NAV'");

$stCookieName = 'stc1'; // a split testing cookie neve
$fileName      = 'teszt'; // fájl nevének első része
$expirationDay = 2; //cookie lejáratának ideje, napok száma mától

if (isset($_COOKIE[$stCookieName])) { //ha a látogatónak már van
split testing cookie-ja
    $stcCookieValue = $_COOKIE[$stCookieName];
} else {
    mt_srand((double)microtime()*1000000);
    $stcCookieValue = mt_rand(1,2);
    setcookie($stCookieName,$stcCookieValue,time() + (3600 *
$expirationDay),"/",$_SERVER['HTTP_HOST']);
}

$smarty = new Smarty;
```

```
$smarty->display("$fileName-$stcCookieValue.tpl");  
?>
```

### A kód rövid magyarázata

Először definiálásra kerül a Smarty könyvtár a `require 'libs/Smarty.class.php'` kifejezéssel. Természetesen a párhuzamos tesztelés megvalósítható Smarty sablonkezelő rendszer nélkül is, ennek is megmutatjuk az alkalmazását.

A `header` definiálja az úgynevezett [P3P](#) Compact Policy-t, amely a cookie visszautasításának megelőzésére szolgál.

A `$stCookieName` változóban kerül definiálásra a visszatérő látogatók azonosítását lehetővé tevő úgynevezett párhuzamos tesztelés (split testing) cookie. Ha Ön a website-ján több helyen is kívánja alkalmazni a párhuzamos tesztelést, akkor ügyeljen arra, hogy a cookie neve különböző legyen az egyes tesztelt oldalaknál.

A `$fileName` változó a HTML fájlok nevének első tagja. A különböző verziókat tartalmazó HTML fájlok elnevezéséről részletesebben lejjebb olvashat.

Az `$expirationDay` változó meghatározza a cookie élettartamát. Ezt alapértelmezetten 30 nap. A cookie, lejáratát után törlődik a látogató böngészőjéből, így ezután már lehet azonosítani visszatérő látogatóként.

A kód további része tartalmazza a tulajdonképpeni működési logikát. Megvizsgálásra kerül, hogy a látogató rendelkezik-e már az adott párhuzamos teszteléshez tartozó cookie-val. Ha igen akkor egyszerűen kiolvasásra kerül a cookie értéke, ha nem akkor a véletlenszám generátor előállít egy 1 és 2 közötti értéket, illetve elhelyezi a cookie-t a látogató böngészőjében.

A kód utolsó része végzi a most vagy korábban előállított véletlenszám alapján a megfelelő verzió megjelenítését, amelyek Ön által előre elkészített HTML fájlok.

### Különböző verziók HTML fájljai

Az egyszerű párhuzamos tesztelésnél két HTML fájlt kell előállítania. Mindkét fájl tartalmazza a teljes oldal forráskódját. Az egyik fájlban a tesztelendő elem egyik verziója, a másik fájlban a másik verziója szerepel.

A két HTML fájlt az alábbi konvenció alapján nevezze el:

[fájlnév]-[verzió].tpl

A fájlnev tetszőleges lehet, de mindenképp úgy válassza meg, hogy egyedi legyen. A választott nevet állítsa be a `$fileName` változó értékének. A verzió értéke 1 vagy 2 lehet, attól függően, hogy a weboldal A vagy B verziójáról van szó. A fájl kiterjesztése `.tpl`, amely arra utal, hogy ez a Smarty sablonkezelő rendszer által használt fájl.

Így például az A verzió neve: `teszt-1.tpl`; a B verzió neve pedig: `teszt-2.tpl`.

A PHP programfájlt mentse tetszőleges, egyedi néven, `.php` kiterjesztéssel és töltsse fel a web kiszolgálóra a megfelelő helyre. Ha a weboldal, amelyet párhuzamos tesztelés alá kíván vetni korábban már letézett más statikus HTML fájlként, akkor Önnek minden hivatkozást amely a statikus fájlra mutat át kell írnia. Ilyen esetben a PHP programot célszerű ugyanazon a néven elnevezni, mint az eredeti fájl, azzal a különbséggel, hogy a fájl kiterjesztése `.php` lesz.

A weboldal két verziójának `tpl` fájlját is töltsse fel a szerverre. A `.php` fájl mellé hozzon létre egy `templates` nevű könyvtárat és ebbe másolja be a különböző verziók `tpl` fájljait.

A Smarty sablonkezelő rendszernek szüksége van még egy munkakönyvtárra. Hozzon létre egy `templates_c` könyvtárat is a `templates` könyvtár mellé. A `templates_c` könyvtárra adjon írásjogot a webservernek.

### **Párhuzamos tesztelés eredményének mérése**

Ahhoz, hogy a párhuzamos tesztelés mérhető legyen, gondoskodjon a weboldal különböző verziói látogatottságának monitorozásáról.

Mindkét verzió HTML forráskódjában helyezzen el egy-egy weboldal monitorozó kódot. A weboldal monitorozó kódnak képesnek kell lennie a két különböző verzió letöltését egyedileg mérni, vagyis meg tudja mondani, hogy mely látogatók, mely verziót látták.

Ezen felül olyan látogatottság monitorozó rendszert válasszon, amely képes az adott oldalra érkező látogatók viselkedését elemezni látogatás következő oldalának viszonyában. Így tudja ugyanis mérni, hogy mely verzió milyen eredménnyel működött - pl. hányan mentek tovább a rendelés oldalra. A fenti feltételeknek maradéktalanul megfelel a [WebMonitor](#) látogatottság elemző rendszer.

### **Az összetett módszer Smarty sablonkezelő rendszer használatával**

Az összetett párhuzamos tesztelés módszer a forráskódjában különbözik az fentebb bemutatott egyszerű módszertől és abban, hogy három HTML fájlt kell elkészíteni, amelyből kettő megjelenése azonos.

A forráskód Smarty sablonkezelő rendszer használatával:

```
<?php
require 'libs/Smarty.class.php';

/* Header */
header("P3P: CP='ALL DSP COR LAW CUR ADMi DEVi TAIi PSDi IVDi CONi
OUR BUS PHY ONL PUR FIN COM NAV'");
```



```

$stCookieName = 'stc'; // a split testing cookie neve
$fileName     = 'teszt'; // fájl nevének első része
$expirationDay = 30; // cookie lejáratának ideje, napok száma mától

if (isset($_COOKIE[$stCookieName])) { // ha a látogatónak már van
split testing cookie-ja
    $stcCookieValue = $_COOKIE[$stCookieName];
} else {
    mt_srand((double)microtime()*1000000);
    $randValue = mt_rand(1,100);

    if ($randValue >= 1 && $randValue <=50) {
        $stcCookieValue = 1;
    } else if ($randValue >= 51 && $randValue <=75) {
        $stcCookieValue = 2;
    } else {
        $stcCookieValue = 3;
    }

    setcookie($stCookieName,$stcCookieValue,time() + (3600 *
$expirationDay),"/",$_SERVER['HTTP_HOST']);
}

$smarty = new Smarty;
$smarty->display("$fileName-$stcCookieValue.tpl");
?>

```

Az egyszerű verzióhoz képest annyi különbség van, hogy nem kettő, hanem három verzió van, amelyből kettő megjelenésre azonos. Az A verzió a látogatók 50%-ának jelenik meg, a B1 ill. B2 verzió látogatók 25-25%-ának.

A fent említett [fájlnév]-[verzió].tpl névadási konvenciónak megfelelően el kell nevezni a három HTML fájlt és bemásolni a PHP fájl mellett lévő templates könyvtárba.

A tpl fájlok neve így például az alábbiak lehetnek: teszt-1.tpl, teszt-2.tpl, teszt-3.tpl

A három verzió látogatottságának monitorozásáról szintén gondoskodnia kell.

### Az összetett módszer eredményének monitorozása

Az összetett módszer monitorozásában annyi különbség van az egyszerű módszer monitorozásához képest, hogy mind három verzió HTML forrásába különböző monitorozó kódot kell elhelyeznie.

A látogatottság monitorozó rendszernek képesnek kell lennie összehasonlítható formában megjeleníteni a B1 és B2 verzió megjelenésének számát, hogy a mérési pontatlanság megállapítható legyen.

### Az egyszerű módszer Smarty sablonkezelő rendszer használata nélkül

Ha Önnek nincs lehetősége a Smarty sablonkezelő rendszer használatára, akkor is tudja működtetni a párhuzamos tesztelést. Ebben az esetben a program logikának tartalmaznia kell a megfelelő verzió beolvasásának illetve megjelenítésének kezelését. A programkód ebben az esetben egy kicsit összetettebb.

Az A-B verziós egyszerűbb párhuzamos tesztelés forráskódja Smarty sablonkezelő rendszer használata nélkül:

```
<?php
    /* Header */
    header("P3P: CP='ALL DSP COR LAW CUR ADMi DEVi TAIi PSDi IVDi CONi
OUR BUS PHY ONL PUR FIN COM NAV'");

    $stCookieName = 'stc1'; // a split testing cookie neve
    $fileName      = 'teszt'; // fájl nevének első része
    $expirationDay = 2; //cookie lejáratának ideje, napok száma mától

    if (isset($_COOKIE[$stCookieName])) { //ha a látogatónak már van
split testing cookie-ja
        $stcCookieValue = $_COOKIE[$stCookieName];
    } else {
        mt_srand((double)microtime()*1000000);
        $stcCookieValue = mt_rand(1,2);
        setcookie($stCookieName,$stcCookieValue,time() + (3600 *
$expirationDay),"/",$_SERVER['HTTP_HOST']);
    }
    $filename = "templates/$fileName-$stcCookieValue.tpl";
    $handle   = fopen($filename, "r");
```

```

$stVersionFile = fread($handle, filesize($filename));

fclose($handle);

echo $stVersionFile;
?>

```

A különbség észrevehető a kód végén. A fájl beolvasását és megjelenítését beépített PHP függvényekkel végezzük. A különböző verziókat tartalmazó tpl fájlok elkészítése és feltöltése a fent leírtakkal megegyezően történik.

### Az összetett módszer Smarty sablonkezelő rendszer használata nélkül

Az összetettebb, A, B1,B2 verziót kezelő párhuzamos tesztelés forráskódja Smarty sablonkezelő rendszer használata nélkül:

```

<?php
/* Header */
header("P3P: CP='ALL DSP COR LAW CUR ADMi DEVi TAIi PSDi IVDi CONi
OUR BUS PHY ONL PUR FIN COM NAV'");

$stCookieName    = 'stc'; // a split testing cookie neve
$fileName        = 'teszt'; // fájl nevének első része
$expirationDay   = 30; // cookie lejáratának ideje, napok száma mától

if (isset($_COOKIE[$stCookieName])) { // ha a látogatónak már van
split testing cookie-ja
    $stcCookieValue = $_COOKIE[$stCookieName];
} else {
    mt_srand((double)microtime()*1000000);
    $randValue = mt_rand(1,100);

    if ($randValue >= 1 && $randValue <=50) {
        $stcCookieValue = 1;
    } else if ($randValue >= 51 && $randValue <=75) {
        $stcCookieValue = 2;
    } else {
        $stcCookieValue = 3;
    }
}

```

```
        setcookie($stCookieName,$stcCookieValue,time() + (3600 *
$expirationDay),"/",$_SERVER['HTTP_HOST']);
    }

    $filename = "templates/$fileName-$stcCookieValue.tpl";
    $handle    = fopen($filename, "r");
    $stVersionFile = fread($handle, filesize($filename));
    fclose($handle);
    echo $stVersionFile;

?>
```

By MarketingSzoftverek

2005.06.01.

Skycentrum Bt.  
1089 Budapest, Golgota utca 3.

Copyright © 2005, Skycentrum Bt. Minden jog fenntartva