

.htaccess

.htaccess fájl beállításával kapcsolatos leírások

- [Hogyan hozhatok létre .htaccess fájlt?](#)
- [Hogyan módosíthatom a fő domain névhez tartozó dokumentum root könyvtárat?](#)
- [Hogyan módosíthatom az alapértelmezett kezdő \(index\) oldalt .htaccess fájl segítségével?](#)
- [Mi az a .htaccess fájl és mire használható?](#)
- [URL átirányítás/átírás .htaccess fájl segítségével](#)
- [HTTP fejléc \(header\) beállítások módosítása .htaccess fájl segítségével](#)
- [suPHP használata a tárhelyen](#)
- [Access-Control-Allow-Origin \(CORS\) header beállítása .htaccess segítségével](#)
- [Fájl létrehozása és módosítása SSH-n keresztül](#)

Hogyan hozhatok létre .htaccess fájlt?

A .htaccess fájl létrehozható a tárhelyen FTP kliens segítségével, a cPanel beépített fájlkezelőjével illetve SSH-n keresztül. Az alábbi két cikkben található leírás alapján lehet létrehozni fájlt:

- [Fájl létrehozása és módosítása FTP-n keresztül](#) (egyszerűbb a kezdők számára)
- [Fájl létrehozása és módosítása SSH-n keresztül](#) (haladók számára)

Abban az esetben ha FTP kliens program segítségével csatlakozik a tárhelyhez, győződjön meg arról, hogy a rejtett fájlok meg vannak jelenítve. Erre azért van szükség, mert a .htaccess fájl ponttal kezdődik.

A fájl létrehozása során ügyeljen arra, hogy a .htaccess fájlneve NEM tartalmaz semmilyen kiterjesztést. A fájl szerkezeti felépítése szerint .htaccess névre hallgat.

MILYEN JOGOSULTSÁGOT ÁLLÍTSAK BE A .HTACCESS FÁJLRA?

Általában a 644-es jogosultság megfelelő szokott lenni, azonban a .htaccess fájl helye befolyásolni fogja a .htaccess fájl alatti könyvtárakban elhelyezett fájlok működését.

HOVÁ HELYEZZEM EL A .HTACCESS FÁJLT?

A .htaccess fájlt jellemzően az adott domain név dokumentum root könyvtárába kell elhelyezni. Azonban a .htaccess fájl hatással lesz az elhelyezési mappa alatt található összes fájlra mappára.

Hogyan módosíthatom a fő domain névhez tartozó dokumentum root könyvtárat?

Alapértelmezésben a fődomain név esetében a tartalom a *public_html* mappából töltődik be. A *public_html* mappa a fődomain névhez tartozó dokumentum root könyvtár amely nem módosítható.

Abban az esetben ha a weboldal a *public_html* mappán belül egy almappából jelenne meg akkor lehetőség van arra hogy egy *.htaccess* fájl segítségével módosítsuk a fő domain név kezdőkönyvtárát.

A módosításhoz a *public_html* mappában kell létrehozni egy *.htaccess* fájlt, és az alábbi tartalmat kell elhelyezni benne:

```
RewriteEngine on
RewriteCond %{HTTP_HOST} ^domain-name.com$ [NC,OR]
RewriteCond %{HTTP_HOST} ^www.domain-name.com$
RewriteCond %{REQUEST_URI} !folder/
RewriteRule (.*?) /folder/$1 [L]
```

A fenti kódrészletben két változót kell mindenképpen módosítania, ezek a következők:

domain-name.com - Írja be a saját domain nevét.

folder - Itt adja meg azt a könyvtárat amelyre szeretné átirányítani a weboldalt.

Hogyan módosíthatom az alapértelmezett kezdő (index) oldalt .htaccess fájl segítségével?

A szerver alapértelmezésben megpróbálja megtalálni a weboldal kezdő oldalát amelyet index oldallak is neveznek. Az index fájl nevek alapértelmezett sorrendje index.php, index.htm, index.html és végül default.htm. A .htaccess fájl módosításával megváltoztathatja a tárhelyén a keresett index fájl nevét, és sorrendjét. Ha egyedi fájlnevet szeretne megadni az index fájlhoz, vagy esetleg másik kiszolgálóról költözött, és az index oldalt ott másként nevezték el. Így a belső hivatkozásokat nem törli az index fájl átnevezése.

[Mi az a .htaccess fájl és mire használható?](#)

AZ ALAPÉRTELMEZETT INDEX FÁJL MÓDOSÍTÁSA .HTACCESS FÁJL SEGÍTSÉGÉVEL

Abban az esetben ha még nincs létrehozva a domain névhez tartozó dokumentum_root könyvtárban .htaccess fájl akkor azt létre kell hoznia. A fájl létrehozását, vagy módosítását el lehet végezni a cPanel beépített fájl kezelőjével, vagy bármilyen FTP kliens segítségével, illetve SSH protokollon keresztül is. Ennek menetéről az alábbi hivatkozásokon keresztül található részletesebb leírást:

- [Fájl létrehozása és módosítása a cPanel beépített fájlkezelőjével](#)
- [Fájl létrehozása és módosítása FTP-n keresztül](#)
- [Fájl létrehozása és módosítása SSH-n keresztül](#)

A fájlban az alábbi kódrészletet kell beilleszteni:

```
#Alternate default index page  
DirectoryIndex elso.html
```

A példában a kezdő oldalt elso.html fájlra hívják. Ezt módosítsa arra a fájlra amelyet Ön be kíván állítani a weboldalának kezdőoldalaként.

Lehetősége van több fájlnev megadására is, ezeket az alábbi formában adhatja meg:

```
#Alternate default index pages  
DirectoryIndex elso.html index.htm index.html index.php
```

Ebben az esetben a webkiszolgáló a tárhelyen megpróbálja megnyitni az elso.html fájlt, majd ha azt nem találja akkor próbálkozik a felsorolásban szereplő többi fájlnevvvel.

Mi az a .htaccess fájl és mire használható?

A htaccess fájl egy szöveges rejtett fájl amelynek segítségével lehetőség van a weboldal működésének befolyásolására. A .htaccess fájl segítségével lehetőség van a tartalom blokkolására is. Alapértelmezésben a .htaccess fájl nem látható a tárhelyen, mivel ponttal kezdődik. Az UNIX alapú kiszolgálókon a ponttal kezdődő fájlok rejtett fájlok. A .htaccess fájlnek nincs neve, a kiterjesztését tekintve a .htaccess a kiterjesztése. Ahhoz, hogy a .htaccess fájl láthatóvá váljon a cPanelen a fájlkezelőben engedélyezni kell a Rejtett fájlok megjelenítése lehetőséget.

MIRE HASZNÁLHATÓ A .HTACCESS FÁJL?

Lehetőség van például a weboldal egyes részeinek jelszavas védelmére. Ezt a típusú jelszavas védelmet is a .htaccess fájl segítségével lehet beállítani. A .htaccess fájlban található legtöbb funkciót a cPanel segítségével is elő lehet állítani ilyen például a könyvtárak jelszavas védelme. Ezt a cPanel kezelőfelületén a **Könyvtárak jelszavas védelme** menüpontban találhatja meg. Abban az esetben ha jelszavas védelmet állít be egy könyvtárra vonatkozóan akkor egy .htaccess direktívát készít.

Egyéb funkciók is definiálhatóak a .htaccess fájlban, ilyen például a **HotLink védelem**, az [URL átirányítás/átírás](#), az [alapértelmezett kezdőoldal beállítása](#), a 301-es és a 302-es típusú átirányítások, és még sok egyéb. Fontos kiemelni, hogy a .htaccess fájl hatással lehet a teljes cPanel fiók működésére. Ezért javasoljuk, hogy mielőtt a .htaccess fájlt szerkesztené az eredeti verzióról készítsen biztonsági mentést.

A .HTACCESS FÁJLBAN HELYTELENÜL MEGADOTT BEÁLLÍTÁSOK ELHÁRÍTÁSA

Abban az esetben ha 500-as hibát jelenít meg a böngésző akkor érdemes átnézni a .htaccess fájlt ugyanis gyakran a fájlban helytelenül megadott direktívák okozzák a problémát.

Ilyen esetben a probléma rendszerint gyorsan orvosolható, mert elegendő átnevezni a .htaccess fájlt .htaccess.txt névre, így a fájlban található szabályokat a webszerver már nem fogja figyelembe venni.

URL átirányítás/átírás .htaccess fájl segítségével

Alapértelmezésben az Ön weboldala elérhető www előtaggal illetve a nélkül is. A Google keresési eredményeiben sajnos rosszabb helyezést ér el az a domain név amelyiknél a tartalom elérhető www-vel is és a nélkül is. Mivel egyes hivatkozásokat a keresők beindexeltek ezért szükség van ezen URL címek átírására, átirányítására.

- [Átirányítás cPanel segítségével](#)

ÁTIRÁNYÍTÁS WWW-RŐL, VAGY WWW-RE

Hogyan irányíthatjuk át az össze hivatkozást a www.pelda.hu címről a pelda.hu címre?

301-es típusú átirányítás segítségével az összes http kérés át lehet irányítani a www.pelda.hu címről a pelda.hu címre vagy fordítva

- **a pelda.hu átirányítása a www.pelda.hu címre**

```
RewriteEngine On
RewriteCond %{HTTP_HOST} !^www.pelda.hu$ [NC]
RewriteRule ^(.*)$ http://www.pelda.hu/$1 [L,R=301]
```

- **a www.pelda.hu átirányítása a pelda.hu címre**

```
RewriteEngine on
RewriteCond %{HTTP_HOST} ^www\.pelda\.hu$
RewriteRule ^/?$ "http://\./pelda.hu/" [R=301,L]
```

MAGYARÁZAT

Az első sorral utasítjuk az Apache-ot hogy használja a rewrite modult.

A következő sor határozza meg feltételt:

```
RewriteCond %{HTTP_HOST} !^www.pelda.hu$ [NC]
```

Csak a http alapú kérésekre vonatkozik az átirányítás `%{HTTP_HOST}`.

Ezt követően adjuk meg azt hogy milyen URL címre vonatkozzon a szűrés, jelen esetben ha az URL címben nem szerepel a www akkor irányítjuk át. Ezt a `!` segítségével állítjuk be.

A `$` karakter határozza meg azt, hogy ha `www.pelda.hu` szerepel az URL címben akkor a következő sorban szereplő szabály alapján kerül átírásra az URL cím.

Az `[NC]` jelentése „no case”, azaz ezen flag használata esetén a kisbetű és a nagybetű között nincs különbség. Több flag esetében azokat a kapcsos zárójelen belül vesszővel választjuk el egymástól, azaz pl. `[NC,OR]` jelölést alkalmazhatunk.

Az utolsó sorban adjuk meg az átirányítási feltételt.

```
RewriteRule ^(.*)$ http://www.pelda.hu/$1 [L,R=301]
```

A `^(.*)` egy reguláris kifejezés ahol a `^` jelöli az „illesztendő” kifejezés elejét.

A `()` segítségével adhatjuk meg hogy az adott karaktereket egységként kezelje.

A `.` karakterrel bármilyen karaktert helyettesíthetünk.

A `*` karakterrel adható meg az előző karakter vagy zárójeles kifejezés legalább nullaszor.

A `$` karakter határozza meg a kifejezés végét.

Ezt követi az az URL cím amire át kívánjuk irányítani a címet. Az utolsó `[]` között található karakterek jelentése:

`L` - az utolsó szabályt jelöli

`R` - Az átirányítás típusát határozza meg.

HOGYAN IRÁNYÍTHATOM ÁT A LÁTOGATÓKAT HTTPS PROTOKOLLRA?

Abban az esetben ha érvényes tanúsítvány lett beállítva a domain névnél akkor javasolt a látogatókat átirányítani a nem biztonságos `http://` címről a biztonságos `https://` címre.

Az átirányítást legegyszerűbben az adott domain névhez tartozó `document_root` mappájában található `.htaccess` fájl módosításával oldható meg.

A példában szereplő sorok segítségével az adott domain név minden felhasználóját át fogja irányítani a nem biztonságos (`http://`) URL-ről a biztonságos (`https://`) URL-re.

A példában szereplő pelda.hu domain nevet cserélje ki a saját domain nevére.

```
RewriteEngine On  
RewriteCond %{HTTPS} off  
RewriteRule (.*) https://pelda.hu/$1 [R=301,L]
```

HTTP fejléc (header) beállítások módosítása .htaccess fájl segítségével

Első lépésként ha még nem létezik .htaccess fájl az érintett domain név dokumentum root könyvtárában akkor hozza létre. Ehhez az alábbi leírásban találhat segítséget:

[Hogyan hozhatok létre .htaccess fájlt?](#)

Abban az esetben ha a fájl már létezik akkor az alábbi leírások alapján tudja őket szerkeszteni:

- [Fájl létrehozása és módosítása FTP-n keresztül](#)
- [Fájl létrehozása és módosítása SSH-n keresztül](#)

MI AZ A HTTP FEJLÉC (HEADER)?

A HTTP fejléc (header) része egy HTTP kérésnek illetve válasznak. Ez határozza meg a HTTP tranzakció működési paramétereit. Bővebb információt az alábbi hivatkozáson találhat angol nyelven:

[List of HTTP header fields](#)

A .htaccess fájl segítségével megváltoztathatóak, vagy kiegészíthetőek a HTTP válasz fejlécek (header).

CHARSET HEADER

A .htaccess fájlban az alábbiakat lehet használni az adott tartalomtípusú fejléc kényszerítésére. A [karakterkészlet fejléce](#) meghatározza a dokumentum karakterkódolását. Így a fejléct a metatag használata nélkül adhatja hozzá:

```
<IfModule mod_headers.c>
    AddDefaultCharset UTF-8
    AddDefaultCharset ISO-8859-2
</IfModule>
```

CONTENT-LANGUAGE HEADER

A .htaccess fájlban a következő használatával lehet beállítani egy nyelvi fejléct. Így a fejléct a metatag használata nélkül adhatja hozzá:

```
<IfModule mod_headers.c>
    DefaultLanguage hu-hu
</IfModule>
```

CACHE-CONTROL HEADER

A Cache-Control az egyik leggyakoribb fejléc, amelyet a weboldalaknál használnak. Ez határozza meg azt, hogy mennyi ideig tárolja a fájlt a böngésző

Ha például a Cache-Control headerben 5 percet állít be, akkor a látogató böngészője letölti az oldalt, majd 5 percig gyorsító tárazza azt. 5 perc elteltével az oldalt újra le kell kérni a kiszolgálóról.

Például:

A következő példában beállítjuk, hogy a weboldalt 5 percig tárolhassák látogatók böngészői.

```
<IfModule mod_headers.c>
    Header set Cache-Control "max-age=300, public"
</IfModule>
```

Szintaxis

a **max-age** másodpercben van beállítva.

A gyorsítótárazási irányelv lehet „public”, „private” vagy „no-store”.

‘VARY’ HTTP HEADEREK HASZNÁLATA A MOBIL OLDALAKHOZ

A következő Google által készített cikk (angol nyelvű) leírja a Vary headerek használatát mobil oldalak esetében:

- [The Vary HTTP Header](#)

BIZTONSÁG

CONTENT-SECURITY-POLICY

A Content-Security-Policy header segít csökkenteni az XSS kockázatokat. További részletekért tekintse meg a következő oldalakat:

- <https://content-security-policy.com/> (angol nyelven)
- <https://www.cert.hu/cross-site-scripting-xss>

STRICT-TRANSPORT-SECURITY (HSTS)

Meghatározza, hogy a böngészők a HTTP helyett csak a HTTPS protokollon keresztül kommunikáljanak. További részletekért tekintse meg az alábbi oldalakat:

- https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security (angol nyelven)
- https://www.owasp.org/index.php/HTTP_Strict_Transport_Security_Cheat_Sheet (angol nyelven)

Engedélyezhető az alábbi tartalommal a .htaccess fájlban:

```
<IfModule mod_headers.c>
    Header add Strict-Transport-Security "max-age=31415926;includeSubDomains;"
</IfModule>
```

Tesztelhető az alábbi parancs kiadásával:

```
curl -I https://example.com
```

A kimenet a következőképp néz ki:

```
[server]$ curl -I https://example.com
HTTP/1.1 200 OK
Date: Tue, 05 Jun 2018 20:05:52 GMT
Server: Apache
Last-Modified: Tue, 05 Jun 2018 16:26:52 GMT
ETag: "2f9-56de78493cbc8"
Accept-Ranges: bytes
Content-Length: 761
Strict-Transport-Security: max-age=31415926;includeSubDomains;
Content-Type: text/html
```

A parancs kimenetében látható a Strict-Transport-Security header

suPHP használata a tárhelyen

MI A SUPHP?

A suPHP egy eszköz a PHP parancsfájlok futtatásához a tulajdonosok engedélyével. Ez egy Apache modulból (mod_suphp) és egy setuid gyöker binárisból (suphp) áll, amelyeket az Apache modul hív meg a PHP interpreter végrehajtó folyamat uid megváltoztatására.

A cPanel felületen a PHP verzió választás menüpontban elérhető PHP verziókkal működik.

Abban az esetben ha valamelyik weboldal esetében szeretné használni a suPHP-t, akkor az adott weboldalhoz tartozó kezdőkönyvtárban (document_root) kell létrehozni egy .htaccess fájlt, majd abban a következő sorok egyikét kell elhelyezni. A beállítást követően rekurzívan, az összes alkönyvtárra is érvényes lesz a beállítás.

PHP 4.4-es verzió esetében:

```
AddHandler application/x-httpd-suphp44 .php .php5 .php4 .php3
```

PHP 5.1-es verzió esetében:

```
AddHandler application/x-httpd-suphp51 .php .php5
```

PHP 5.2-es verzió esetében:

```
AddHandler application/x-httpd-suphp52 .php .php5
```

PHP 5.3-as verzió esetében:

```
AddHandler application/x-httpd-suphp53 .php .php5
```

PHP 5.4-es verzió esetében:

```
AddHandler application/x-httpd-suphp54 .php .php5
```

PHP 5.5-ös verzió esetében:

```
AddHandler application/x-httpd-suphp55 .php .php5
```

PHP 5.6-os verzió esetében:

```
AddHandler application/x-httpd-suphp56 .php .php5
```

PHP 7.0-ás verzió esetében:

```
AddHandler application/x-httpd-suphp70 .php .php5
```

PHP 7.1-es verzió esetében:

```
AddHandler application/x-httpd-suphp71 .php .php5
```

PHP 7.2-es verzió esetében:

```
AddHandler application/x-httpd-suphp72 .php .php5
```

PHP 7.3-as verzió esetében:

```
AddHandler application/x-httpd-suphp73 .php .php5
```

PHP 7.4-es verzió esetében:

```
AddHandler application/x-httpd-suphp74 .php .php5
```

PHP 8.0-ás verzió esetében:

```
AddHandler application/x-httpd-suphp80 .php .php5
```

PHP 8.1-es verzió esetében:

```
AddHandler application/x-httpd-suphp81 .php .php5
```

PHP 8.2-es verzió esetében:

```
AddHandler application/x-httpd-suphp82 .php .php5
```

PHP 8.3-as verzió esetében:

```
AddHandler application/x-httpd-suphp83 .php .php5
```

PHP 8.4-es verzió esetében:

```
AddHandler application/x-httpd-suphp84 .php .php5
```

PHP 8.5-ös verzió esetében:

```
AddHandler application/x-httpd-suphp85 .php .php5
```

A verziókénti bekapcsolt modulok globális beállítások, de a php.ini beállításokat le lehet cserélni egy .php.ini fájl elhelyezésével, illetve az 5.4-es PHP verziótól kezdődően elhelyezhet egy .user.ini fájlt, amiben egyesével felülbírálhatóak a PHP beállítások az adott könyvtárra (és annak alkönyvtáira).

Access-Control-Allow-Origin (CORS) header beállítása .htaccess segítségével

Első lépésként ha még nem létezik .htaccess fájl az érintett domain név dokumentum root könyvtárában akkor hozza létre. Ehhez az alábbi leírásban találhat segítséget:

[Hogyan hozhatok létre .htaccess fájlt?](#)

Abban az esetben ha a fájl már létezik akkor az alábbi leírások alapján tudja őket szerkeszteni:

- [Fájl létrehozása és módosítása FTP-n keresztül](#)
- [Fájl létrehozása és módosítása SSH-n keresztül](#)

Íme egy gyorsan beilleszthető kód részlet, amelyet akkor használhat, ha be kell állítania az Access-Control-Allow-Origin fejléceket a .htaccess fájljában.

Az Access-Control-Allow-Origin meghatározása:

Biztonsági okokból a böngészők korlátozzák a szkripteken belül kezdeményezett, több forrásból származó HTTP-kéréseket. Például az XMLHttpRequest ugyanazt a származási házirendet követi. Tehát egy XMLHttpRequest alkalmazást használó webalkalmazás csak HTTP kéréseket küldhet a saját domain nevére. A webes alkalmazások fejlesztése érdekében a fejlesztők arra kérték a böngészőgyártókat, hogy engedélyezzék az XMLHttpRequest számára, hogy domain nevek közötti kéréseket engedélyezzék.

[...]

A CORS a webszerverekhez a domain nevek közötti hozzáférés-vezérlést biztosít, amelyek lehetővé teszik a biztonságos domain nevek közötti adatátvitelt.

Access-Control-Allow-Origin

A CORS BEÁLLÍTÁSA

A használatához tehát be kell állítania a megfelelő fejléceket. A `.htaccess` fájlban adja hozzá az alábbi kódrészletet:

```
Header Set Access-Control-Allow-Origin "https://az.on.kulso.eroforrasanak.cime.vegzodes"
```

A fentiek lehetővé teszik a fejléceket küldő webhely számára, hogy erőforrásokat (például AJAX kéréseket vagy webes betűtípusokat) kérjen a „`https://az.on.kulso.eroforrasanak.cime.vegzodes`” domain név alatt elhelyezett weboldalról. Vegye figyelembe a protokollt, ez – ebben az esetben – csak a HTTPS kéréseket teszi lehetővé. A HTTP -kérések továbbra is blokkolva lesznek.

A CORS LETILTÁSA

Ha teljesen le szeretné tiltani a CORS-t (amit nem javasolunk, azonban tesztelés esetében hasznos lehet) akkor az alábbi kódrészletet helyezze el a `.htaccess` fájlban:

```
Header Set Access-Control-Allow-Origin ""
```

Mint fentebb említettük, biztonságosabb úgy beállítani, hogy az `Access-Control-Allow-Origin` tartalmazza azokat a domain neveket, amelyekről az alkalmazás adatokat kérhet (vagy adatokat küldhet).

A CORS ENGEDÉLYEZÉSE TÖBB DOMAIN NÉVRE VONATKOZÓAN

Ha több domain neve van, és CORS fejléceket szeretne beállítani az adott domain név alapján, akkor használhatja az alábbi kódrészletet a `.htaccess` fájlban:

```
SetEnvIf Origin "http(s)?://(www\.)?(google.com|staging.google.com|development.google.com)$" Acc  
Header add Access-Control-Allow-Origin %{AccessControlAllowOrigin}e env=AccessControlAllowOrigin
```

Mivel a fejlécben csak egy CORS domain név lehet, ezért létre kell hoznia egy változót, ha ezt több helyen szeretné használni.

Fájl létrehozása és módosítása SSH-n keresztül

SSH-n keresztül többféle módon van lehetőség fájl létrehozására. Ebben a cikkben kétféle szövegszerkesztő segítségével mutatjuk be a fájl létrehozást.

- **nano** – kezdő felhasználók számára
- **vim** – haladó felhasználók számára

Ha a fájlok szerkesztése nehézkes parancssorból akkor javasoljuk, hogy FTP-n keresztül hozzon létre fájlt, ebben az alábbi leírás lehet segítségére:

Fájl létrehozása és módosítása FTP-n keresztül

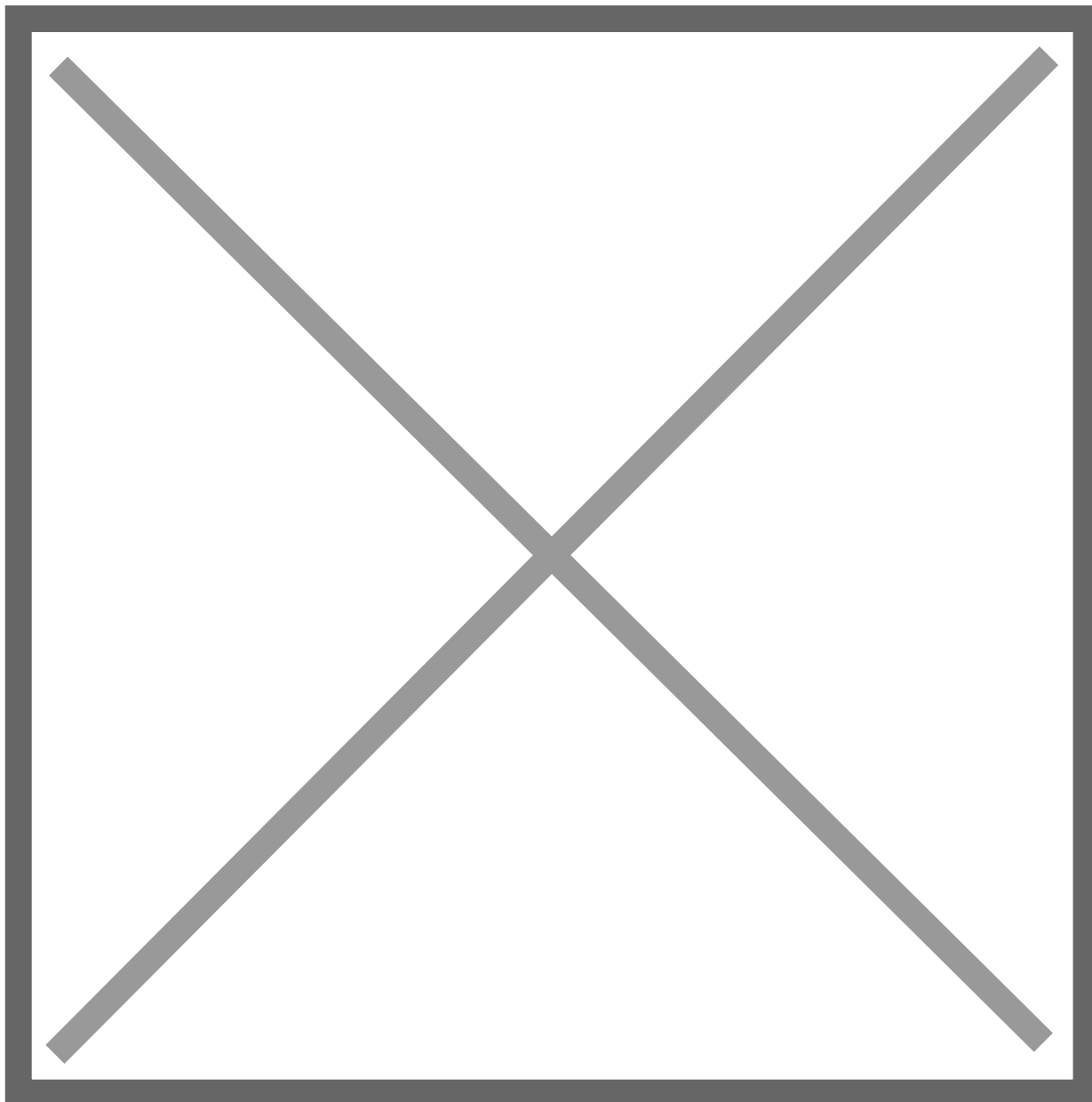
A következő leírás csak példaként szolgál arra hogy hogyan tud létrehozni vagy módosítani fájlt SSH segítségével. A leírást követve győződjön meg arról, hogy a megfelelő helyen hozza létre a fájlt. Módosítás esetén pedig azt a fájlt módosítja amelyet szeretne.

FÁJL LÉTREHOZÁS NANO SEGÍTSÉGÉVEL

1. Jelentkezzen be a szerverre SSH-n.
2. Navigáljon abba a könyvtárba ahol a fájlt létre szeretné hozni, vagy szerkeszteni szeretné.
3. Ezt követően adja ki a **nano** parancsot, majd szóközzel elválasztva folytatólagosan írja be a létrehozni/szerkeszteni kívánt fájl nevét.

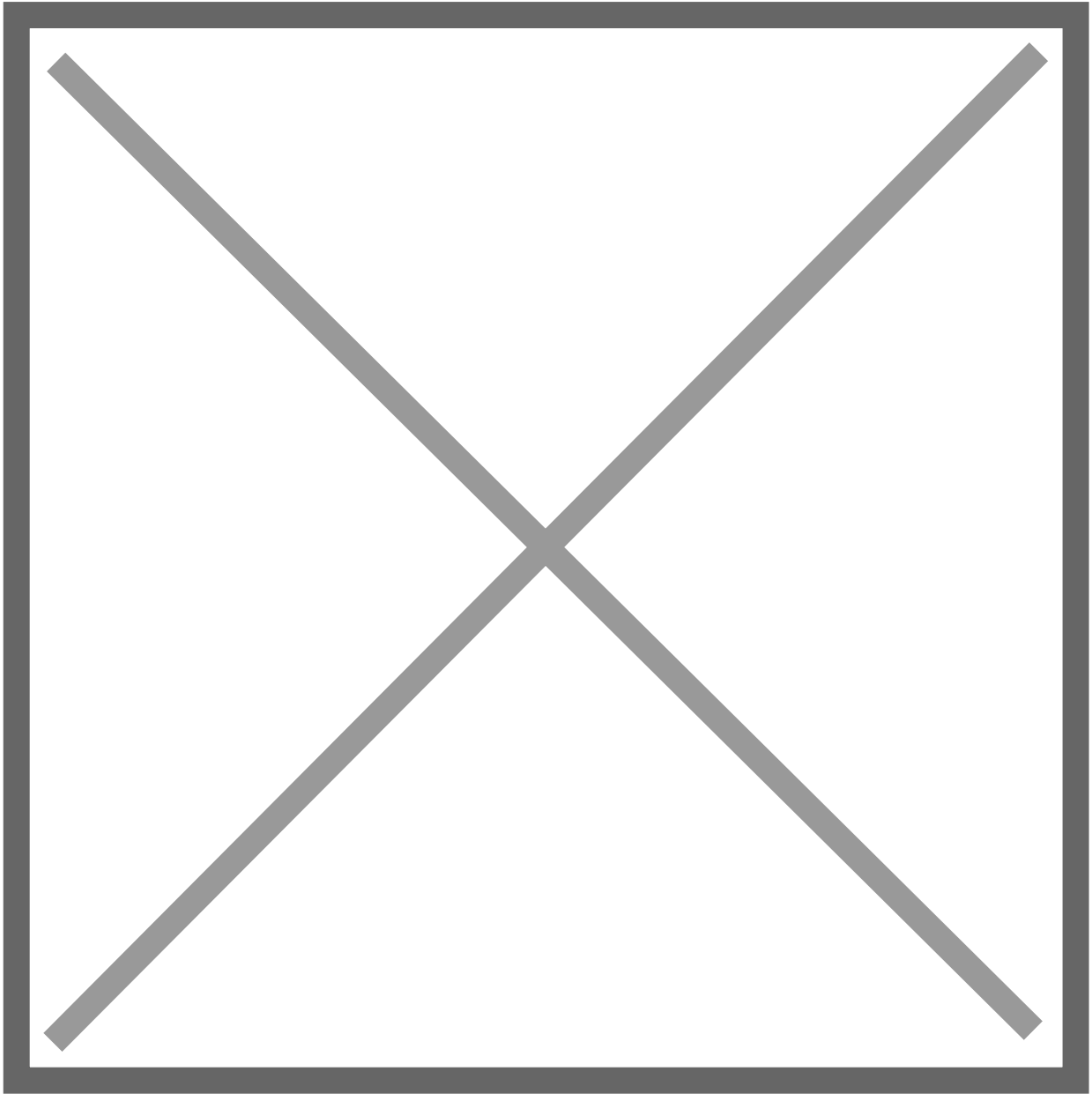
```
[server]$ nano .htaccess
```

4.



Ezt követően kezdje el begépelni a tartalmát a fájlnek.

5. Amikor a szükséges tartalommal feltöltötte a fájlt akkor tartsa lenyomva **Ctrl** billentyűt, majd nyomja le az **O** billentyűt (Ctrl+O).



A képernyő alsó részén a szerkesztő rá fog kérdezni még egyszer az előzőleg megadott fájl nevére.

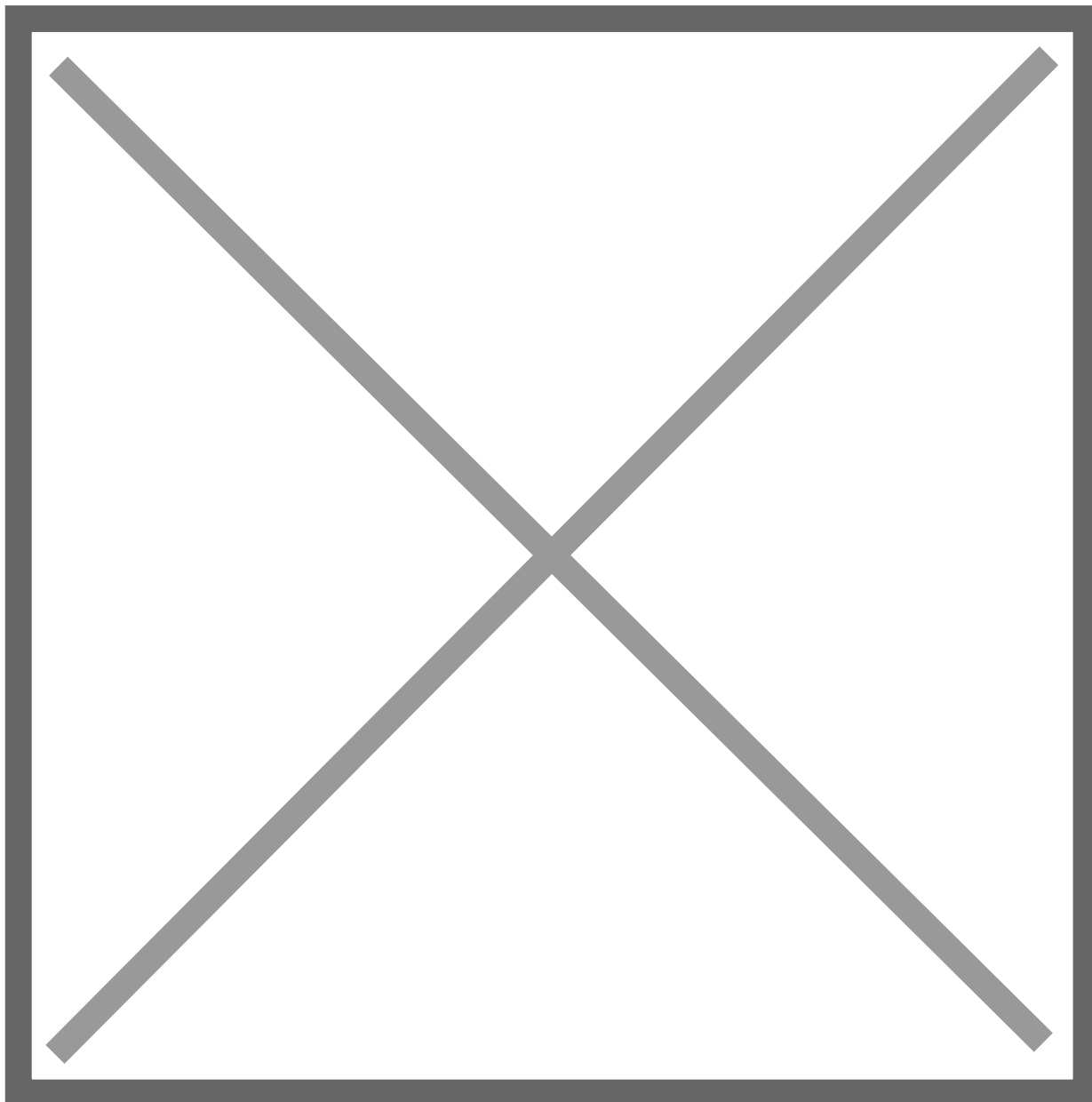
6. A fájl mentését az Enter billentyű lenyomásával hagyhatjuk jóvá.
7. A szövegszerkesztőből a **Ctrl+X** lenyomásával léphet ki.

FÁJL LÉTREHOZÁS VIM SEGÍTSÉGÉVEL

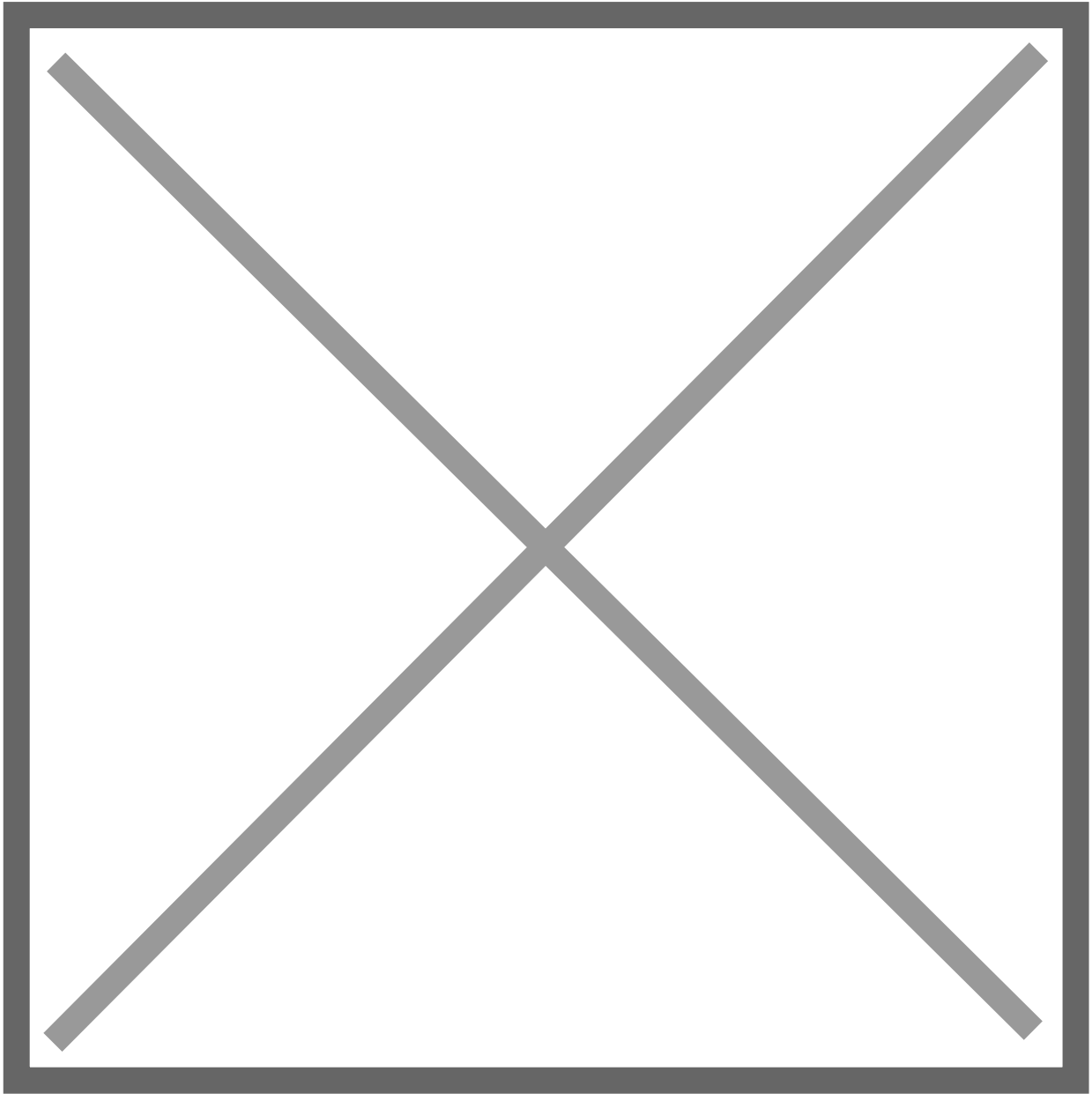
1. Jelentkezzen be a szerverre SSH-n.
2. Navigáljon abba a könyvtárba ahol a fájlt létre szeretné hozni, vagy szerkeszteni szeretné.
3. Ezt követően adja ki a **vim** parancsot, majd szóközzel elválasztva folytatólagosan írja be a létrehozni/szerkeszteni kívánt fájl nevét.

```
[server]$ vim .htaccess
```

4.

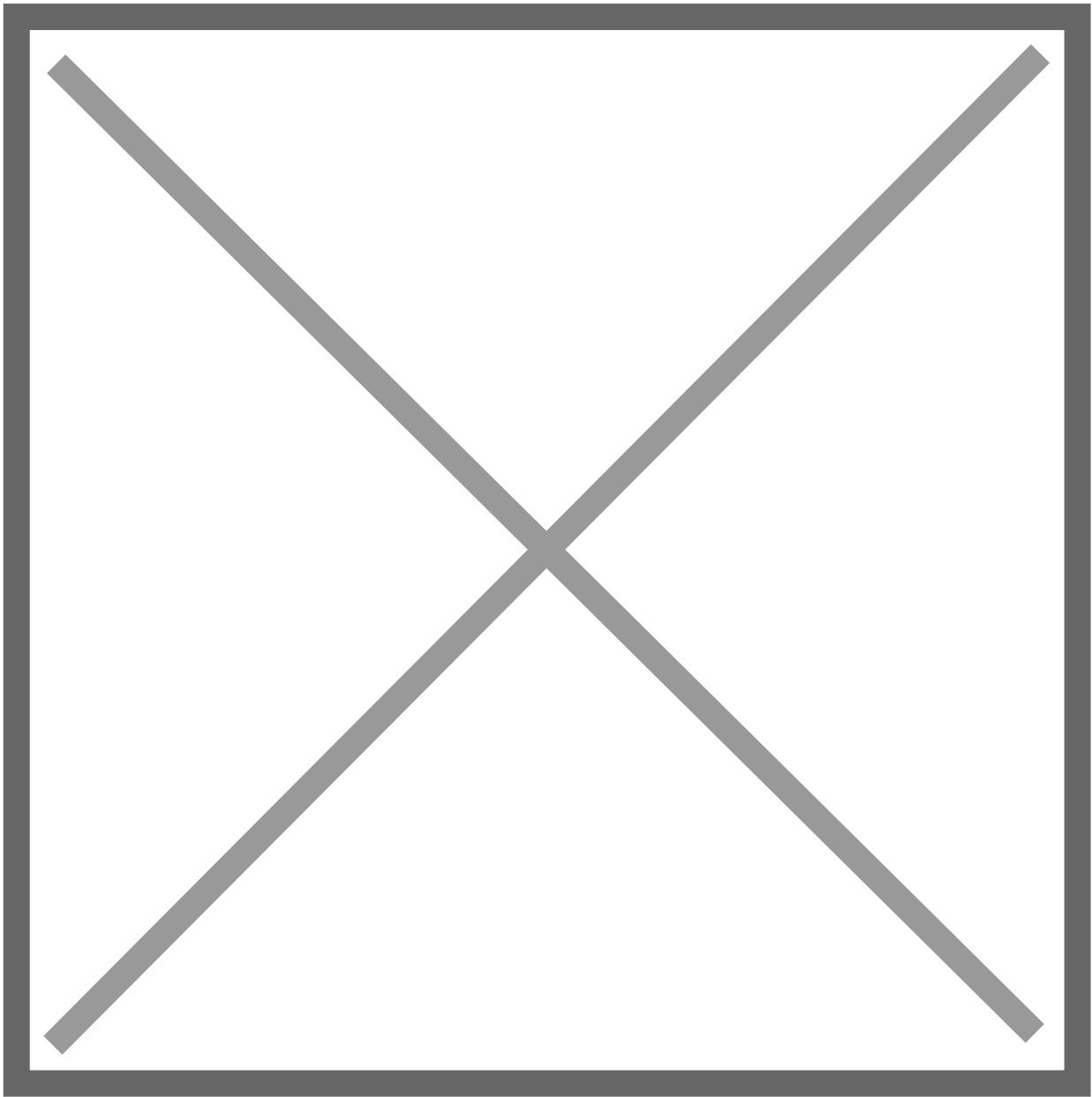


A szövegszerkesztő elindulását követően akkor tudja adattal feltölteni a fájlt ha lenyomja az **I** billentyűt, ugyanis ezzel vált át beszúrás (INSERT) módba.



5. Amikor a szükséges tartalommal feltöltötte a fájlt akkor az **Esc** billentyű segítségével lépjen ki a beszúrási módból.

6.



A fájl mentéséhez írja be a **:wq** parancsot, végül az **Enter** billentyű segítségével mentse a fájlt.