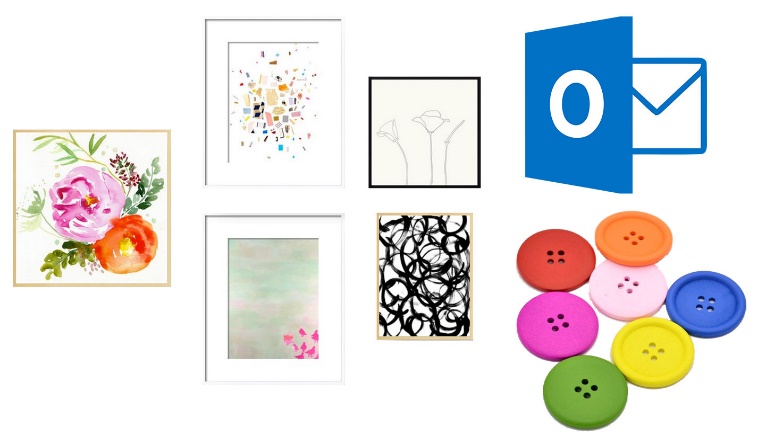
# Pár dolog, amit a HTML levelekről tudni érdemes



## Háttérképek, gombok és az Outlook

Ez a dokumentum elsősorban a HTML levelek háttérkép kérdésköréből indult el, de végül került bele pár más érdekesség is.

(fejlesztés alatt álló verzió, még majd bővülhet)

* Lehet-e háttérképet használni HTML levélben? – Igen, lehet, megoldható
* Használjunk-e háttérképet HTML levélben? – Csak ha nem okoz problémát!

**Mindenekelőtt fontos megjegyezni, hogy Veeva levelek esetében a VML és úgy általában az HTML megjegyzésekben használt MSO IF vezérlők nem működnek, a Veeva rendszere ezeket kitörli!**

Éppen ezért a dokumentumban foglalt VML és más trükkös megoldások maximum csak hagyományos leveleknél használhatók!

A mellékelt példák közé most éppen ezért készült egy teljesen különálló, leegyszerűsített HTML sablon (*simplified-email-template.html*), amely nagyon jól működik és használható akár Veeva levelek esetében is – sőt, egy moustache-template-es unsubscribe elem is van benne – de megfelelő lehet más, szabvány 600 pixel széles (30-30 tartalmi margóval, így 540 pixel széles tartalmi képekkel) HTML levelekhez!  
További általános okosságokat a 4. oldalon látható „további okosságok” cím alatt találni.

No, de visszatérve a háttérkép és gombok problémájához:

A problémát elsődlegesen – és természetesen – az Outlook okozza, amelyben sem a háttérkép-használat, sem az inline típusú elemek blokk típus szerinti használata nem támogatott a CSS módszerekkel. Ez utóbbi a gombok miatt érdekes, ám nem csak a lekerekített, hanem a szögletes gombok miatt is! **Nagyon fontos kiemelni, hogy a háttérkép-problémakörbe tartozik a speciális háttérszín is, így a színátmenetes felületek is!**

Szerencsére akad megoldás mind a háttérkép, mind a gombok problémára az Outlookban is: a VML (Vector Markup Language), azaz a Vektoros Leírónyelv. Ezt a Microsoft dolgozta ki és elsősorban az Office számára tervezték – így az Outlook is használja. Nagyszerű és igazából viszonylag egyszerű megoldást ad az említett problémákra, ám van egy nagy gond vele: a VML elemeket nem lehet egymásba ágyazni, így az alapvetően nem megoldható, hogy egy háttérképes dobozban van egy kattintható gomb is.

Az egymásba ágyazási problémát vizuálisan azért meg lehet oldani azzal a kis trükkel, hogy a gomb kódszinten pl. a háttérképet tartalmazó wrapper doboz előtt van, ám abszolút pozícionálással mégis a dobozba van helyezve. Mivel ez a pozícionálás csak az Outlookot érinti (többi levelezőnél a CSS megoldás érvényesül), így viszonylag biztonsággal lehet pozícionálni, ám a megoldás jellegéből adódóan ez mégsem igazán ajánlott, különösen, ha mondjuk egy hosszabb levél, több szöveges és egyéb blokk után következik a gomb!

Bizonyos esetekben a háttérképek használatánál – ha amúgy sem a szöveg alatt látható, hanem egyszerű háttérszínes szövegdobozokon kívül, pl. két oldalt elhúzódóan – lehet esetleg olyan megoldásokat is találni, amely a háttérkép hatását kelti, ám egyszerű képek használatáról van szó.

**Konklúzió:**

Ha szükséges gombot is tenni a hátteres részbe, akkor meg kell vizsgálni, hogy mennyire szükséges a hátteret úgy megoldani, nincs-e lehetőség valami olyan, sima képet tartalmazó megoldásra, amely látványban hasonló hatást ér el, de mégsem háttér. A gomb működése fontosabb szempont, mint a design, kivéve, ha az ügyfél ragaszkodik a háttérhez és inkább tud engedni a gomb megoldásából.

## Gombok

A gombok beillesztéséhez több megoldás is lehetséges, ám ebből egy van, ami igazán jónak mondható, ez pedig a CSS+VML megoldás. Ebből adódóan, ha egy levélben gomb van és egyúttal háttérképet is szeretnénk használni, akkor a gomb tekintetében már mindenképp valami rosszabb megoldást kell találni. Kérdés, hogy mi a fontosabb? Háttérkép, vagy egy bombabiztos gomb?

Megoldások gomb beillesztéshez:

* Az A tag inline-block (vagy simán block) típusként definiálása CSS-ben: az Outlook nem támogatja, ott a link szövege lesz csak gomb és lesz csak színes.
* Képként használt gomb: mindenhol jól jelenik meg, de csak akkor, amikor már a képeket is betöltötte a felhasználó. Addig viszont elég tragikus megoldásnak minősül.
* Külső wrapperes elemmel létrehozott gomb: amikor pl. egy kis táblázatot hozunk létre, azon belül az egy cellája a gomb, ez kap háttérszínt is. Vizuálisan mindenhol jó megoldásnak tűnik, ám csakis a szöveg lesz benne kattintható, nem az egész gomb!
* CSS+VML megoldás: ez a legbiztosabb, mivel minden helyen, képbetöltés nélkül is megjelenik a gomb, az egész felülete kattintható, ráadásul egy másik problémát, a lekerekített sarkokat is megoldja!

## Padding, margin

A térköz és margó CSS tulajdonságok támogatottsága igencsak esetleges. Az közismert, hogy HTML levél esetében nem lehet a modern CSS és blokk elemek építkezésére hagyatkozni – sajnos vissza kell menni egészen 2000-be és a táblázatokkal, cellákkal és „spacer” képekkel megoldott váz-építést kell használni. Ezt webes környezetben már az 1999-ben megjelent 4.01-es HTML szabvány sem ajánlotta (igaz, akkoriban még gyakori volt, mivel a böngészők megfelelő CSS támogatottsága borzalmas volt), ám HTML leveleknél mégis ehhez az eszközhöz kell nyúlnunk.

A HTML táblákkal – ha nem is korszerűen – igen biztos keretet lehet építeni a levélhez. Éppen ezért érdemes munka előtt alaposabban megnézni a levél design grafikát, hogy elemezzük és értelmezzük azt. Akár gyorsan papíron is felvázolhatjuk a felépítést: hol vannak külön blokkok, amelyek egy-egy TD-be kerülnek, hol vannak összevont sorok vagy oszlopok, illetve hol lehet egyszerűbb elemeket használni.

A margók és térközök kezelése táblázatoknál különösen rosszul megoldott: például Outlook esetében, ha egy TD-re rakunk CSS padding térközt, akkor az egyszerű szövegnél működik. De ha egy beágyazott táblázatot rakunk bele (például, hogy ikonos felsorolást készítsünk), akkor különös módon ez a térköz eltűnik!

Külön fontos megemlíteni, hogy a „shorthand” megoldások sem mindig támogatottak! Például egy padding esetében a Padding: 5px 15px; nem jó megoldás, célszerű ezt külön egyedi attribútumokra szétszedni: Padding-top: 5px; Padding-right: 15px; Padding-bottom: 5px; Padding-left: 15px;

## A példa

A mellékelt példa levél HTML bizonyos szintig még „template-ként” is használható, de elsődlegesen a lehetséges megoldások illusztrálására, egy segédanyagnak készült. Ezért is sűrűn kommentelve van, amiket természetesen a végső, készítendő levélbe nem kell átmásolni.

**Háttérkép:**

Ez a megoldás két részből áll. Egy wrapper TD lesz az elem, amelyik a hátteret kapja. A TD rendelkezik a megszokott CSS háttérképes attribútumokkal (a példában csempézett megoldással, de a pozícionálás és az ismétlés funkciók is használhatók). Fontos, hogy itt a shorthand CSS attribútumokat használjuk! (azaz mint a példában is, „background: ……”)

A TD után közvetlenül nyílik a VML-es megoldású wrapper doboz. A példában egy abszolút pozíciós gomb is itt kapott helyett, de ettől most eltekinthetünk.

Az hátteres doboz lezárása értelemszerűen a </TD> elemnél ér véget és így a VML-es lezárást is itt kell, a </TD> előtt beilleszteni.

**Gomb:**

A példában egy CSS+VML megoldású gomb található, ráadásul abszolút pozícionálással. Ha egyszerűbb módon, helyben szeretnénk csinálni, akkor a példához képest annyiban módosul a kód, hogy a CSS gomb (ROUNDED BUTTON kommentnél lévő TABLE, azon belül is elsődlegesen az A elem) után közvetlenül jöhet a VML kód (ami a példában a „NESTED” VML BUTTON kommentnél található, a háttérképes megoldás nyitó VML része előtt). A VML kódban lévő abszolút pozícionáló CSS paramétereket ilyenkor el kell hagyni!

A működésének lényege: A CSS változat az MSO (Outlook) megjelenítőben nem látszik, helyette a VML jelenik meg, míg más levelező programokban meg fordítva.

A gomb ezen felül lekerekített változatot tartamaz, sima, szögletes gombhoz a CSS változatból a kerekítés attribútumokat kell kihagyni, a VML változatnál pedig „v:roundrect” helyett „v:rect” elemet kell használni (ügyelve a nyitó és záró elemre is), továbbá nincs szükség az „arcsize” paraméterre sem.

**Térközök:**

A levél bal és jobb oldalán (illetve felül) húzódó térközt margin vagy padding megoldással nem lehetett volna rendesen elkészíteni (hogy az mindenhol működne is), így a nagy wrapper táblázat lett úgy kialakítva, hogy a bal és jobb oldalon végighúzódik egy-egy üres cella, amely ráadásul méretre állított térköz-tartó képet is tartalmaz (mivel előfordulhat, hogy a beállított szélesség mégis elveszne).

**A térközökre általánosan bevált megoldás, ha üres, méretre állított cellákkal (akár oldalt, akár függőlegesen) állítjuk be azt. Kritikus esetben spacer kép is alkalmazható.**

## Naptár

Gyakran előfordul, hogy olyan gombot vagy linket kell a levélben elhelyezni, amelyik a felhasználó naptárába elhelyez egy eseményt. Ez több rendszer és program esetében a szabvány .ICS fájlokkal megoldható (Windows és Mac rendszereken is felismeri és a beállított naptár programhoz hozzáadja), ám előfordulhat, hogy felmerül az igény, hogy GMAIL alatt olvasva a levelet a Google naptárhoz adja hozzá, ahol egy megfelelő linkkel történik a művelet. Hasonlóképp lehet eljárni Yahoo levelezőn és naptár esetében is. A példában erre található megoldás, hogy csak egy gomb jelenjen meg, attól függően, hogy hol nézzük a levelet.

Persze, előfordulhat az a reális eset, hogy bár GMAIL-ben nyitottuk meg a levelet, de mégis a Windows naptárhoz szeretnénk hozzáadni. Ilyen lehetőségek miatt lehet, hogy célszerűbb úgy megtervezni az oldalt, hogy a naptárhoz hozzáadás kis ikonokkal van megoldva és a felhasználó maga választhatja ki, hogy hova szeretné hozzáadni.

## További okosságok

**Hosszú szövegek (különösen URL szövegek)**

Az az ilyen szövegeket a megjelenítő túlnyomó többsége automatikusan töri, ám – természetesen – az Outlook nem így jár el. A sor széttolja a wrapper elemet és az egész levél szétesik. Egyszerű és gyors megoldás a word-break: break-all; CSS szabállyal megoldható, ám csak olyan esetben, ha a hosszú szöveg nem egy bekezdés szöveg része, mivel ez minden folyó szöveget félbetör a sor végén a bekezdésen belül.

Szerencsére az ilyen hosszú linkek általában külön sorban kezdődnek, vagy ha van is előtte valami szöveg (pl. „További információk:”), akkor az sem olyan hosszú, hogy a sor végén gondot okozzon és csak a hosszú link szöveg kerül törésre.

Amennyiben ez a törés mégsem alkalmazható, mert olyan jellegű a folyószöveg, akkor – mivel úgyis fix szélességgel dolgozunk – végső megoldásként a link szöveg részében elhelyezett szóközök is jó megoldást adhatnak.

**Felsorolások, különösen másmilyen színű bulletpointtal, számmal**

Egyszerűbb és biztosabb megoldás táblázatba szervezni ezt is.

<table cellspacing="0" cellpadding="0" border="0" width="475">

    <tbody>

        <tr>

            <td width="13" style="width: 13px; color:#00abc7; font-size:13px; text-align: left;" valign="top">

                &bull;

            </td>

            <td width="462" style="width: 462px; color: #301946; text-align: left;" valign="top">

                Lorem Ipsum Dolores Sith Ramen

            </td>

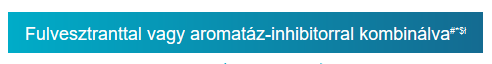
        </tr>

    </tbody>

</table>

**A gomb téma újra, csak mégsem az: színes-szagos szövegdobozka**

Igazából ez is olyan, mint a gomb-téma, különösen, ha ez a szövegdoboz speciális hátteret kap – képi háttér vagy színátmenet.



A megoldás itt a VML használata (így ügyelni kell, hogy külső konténeren a háttér már ütközést okozhat!), de mivel itt nincs linkelés, így lehet kicsit egységesebb megoldást alkalmazni, amelyik egyben tartalmazza a CSS és a VML megoldást.

Példaként lásd a spec-boksz.inc.html mellékelt fájlt.

A doboz szerkezetének lényege: Egy TABLE szervezi a doboz alapját, amiben egy TD-n belül van a doboz maga (ám a TD rendelkezik egy egyszerű háttérszínnel – ez csak praktikus fallback olyan esetekre, amikor sem a CSS gradient, sem a VML nincs támogatva).

Ezen belül kezdődik a VML rész, amely csak Outlook alatt valósítja meg a kívánt színátmenetes felületet. Fontos, hogy ebben az esetben a VML textbox stílusdefiníciójában a „mso-fit-shape-to-text” mindenképp „false” legyen, mivel a hátteret tartalmazó VML elcsúszik, ha igazítani próbáljuk az itt bele ágyazott táblázatba. A gradiens színt a VML fill elemmel lehet beállítani (vízszintes átmenetnél 90 fokkal elfordítva).

Ezután következik a belső, content táblázat, amelyik a doboz méretét hivatott pontosan beállítani. Sajnos margók meg térközök használata – főleg Outlook esetében – nem garantál biztos és / vagy pontos megoldást, így a méretre állított belső táblázat TD-je a legbiztosabb. Ezen a TD-n van a CSS-ben megadható színátmenet, amelyet az Outlook nem tud értelmezni (számára már ott van a VML), de a többi megjelenítő ezt használja (leszámítva esetleg párat, amikhez meg ott a korábban említett fallback szín).

Ezek után már csak a bezáró elemek jönnek, a megfelelő sorrendben, visszafelé.

**Egy-egy vonallal határolt tartalom**

Nem ritka grafikus megoldás, amikor egy szövegblokkot vagy címet egy-egy vonal határol.



A látszólag egyszerű elem mégis számos problémát, elcsúszást tud okozni vagy a tartalmi rész körül lévő térközök vagy a két vonal miatt (vagy mindkettő).

A példaként mellékelt one-border-box.inc.html tartalmaz egy erre vonatkozó kódrészletet, amely a táblázatos felépítést alkalmazza, sőt, spacer képek is kerültek bele, hogy az egyes cellák méretét (különösen magasságát) pontosan szabályozzuk. A felépítés nagyon egyszerű: egy 5 soros, 1-1 cellával egy sorban. Az első és utolsó sor 1 pixel magas, a vonal színével, mint háttérszín. A második és negyedik sor a pontosan beállított térköz. Végül, a harmadik cella a tartalom.

**Fájlnevek**

Fontos! Csakis csupa kisbetű, szám, kötőjel, aláhúzásjel, a végén szabvány kiterjesztéssel!

Mindenféle szóköz meg össze-vissza megoldásnál előfordulhat, hogy pl. GMail alatt nem jelenik meg egy-két kép!