# JavaScript 2.

## Bevezetés

Ebben a leckében külső fájlba tesszük a JavaScript kódot, és egy gomb megnyomásakor módosítjuk az oldal egy bekezdésének szövegét.

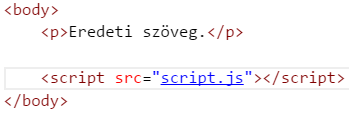
## Előkészületek

1. Nyisd meg a *proba* mappát, majd abból a *proba.html* fájlt a Visual Studio Code-ban!
2. Nyisd meg az oldalt a böngészőben is!
3. Nézd át az eddig elkészített HTML kódot!

## Külső JavaScript

A weblapok készítésénél általában az a jó, ha elkülönítjük egymástól a HTML, a CSS és a JavaScript kódokat. Ráadásul így ugyanazt a kódot több oldalon is használhatjuk.

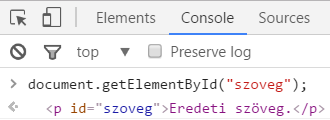
Ezért (a CSS fájlokhoz hasonlóan) a továbbiakban a JavaScript kódokat is külön fájlba fogjuk írni. A JavaScript fájlokat .js kiterjesztéssel fogjuk menteni.

1. Hozz létre egy új fájlt *script.js* néven! Írj bele egy megjegyzést!
2. Csatold a HTML fájlhoz így:  
     
   Figyeld meg, hogy most is a body rész végére tettük a script elemet!

Megjegyzés: Ha megadunk egy JavaScript fájlt, akkor a script tagbe már ne írjunk kódot, mert nem fog futni!

## Oldal elemének megadása

Ha módosítani szeretnénk a weblap egy elemét, először meg kell adnunk, melyik elemről van szó. Most a 9. sorban lévő bekezdést szeretnénk megadni. Ehhez először egy azonosítót rendelünk a bekezdéshez, majd erre hivatkozunk a scriptben.

1. Rendeld hozzá a „szoveg” azonosítót a bekezdéshez, majd frissítsd az oldalt!  
   
2. Nyisd meg a böngészőben a fejlesztői eszközöket, és lapozz a konzolra!
3. Az éppen betöltött weblapra document néven hivatkozhatunk a scriptekben. Kezdd el írni, hogy document, majd a Tab billentyű megnyomásával válaszd ki a listából!
4. Tegyél utána pontot, majd a listából válaszd a getElementById metódust! Ezzel hivatkozhatunk egy adott azonosítójú elemre.
5. Add meg a "szoveg" azonosítót, majd nyomj Entert!  
   

## Elem tartalmának módosítása

Egy elem tartalmát az innerHTML tulajdonsággal tudjuk elérni.

1. Hívd vissza a konzolon az előző parancsot (fel nyíl), és egészítsd ki így:  
   
2. Az elem tartalmát módosíthatjuk egy értékadó utasítással:  
     
   Próbáld ki! Figyeld meg, hogy az oldalon is megváltozott a szöveg!

## Függvények

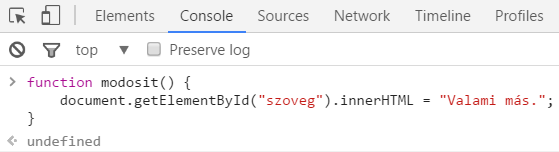
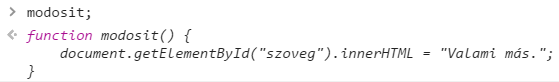
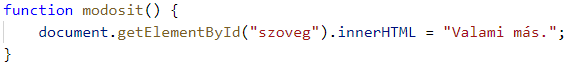
A függvényekben megadhatunk egy vagy több utasítást, amelyeket azután egyszerre, a függvény hívásával lehet lefuttatni.

Ez hasznos, ha ugyanazokra az utasításokra több helyen is szükség van egy programban. A függvényt csak egyszer kell megírni, és akárhány helyről meg lehet hívni.

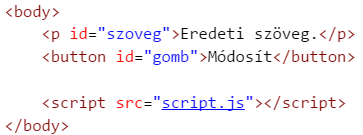
A függvény formája:

function *nev*() {  
 *utasítások*  
}

Próbáljunk meg először a konzolon létrehozni egy függvényt!

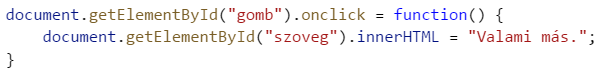
1. Írd be a következő parancsot, majd indítsd el!  
     
   Az undefined szó azt jelzi, hogy a modosit függvény nem számol ki, nem ad vissza értéket. Mit csinál a függvény?
2. Próbáld ki, mi történik, ha beírod a függvény nevét!  
   
3. Ha meg akarod hívni a függvényt, akkor a zárójeleket is meg kell adni. Próbáld ki!  
     
   Figyeld meg, hogy most megváltozott az oldal szövege!
4. Másold be a JavaScript kódba a függvény!  
   

Azt szeretnénk, hogy egy gombra kattintáskor fusson le a függvény. Ezért először elhelyezzük a gombot, majd utána a gombra kattintáshoz hozzárendeljük a függvényt.

1. Helyezz el egy gomb azonosítójú button elemet az oldalon!  
     
   Nézd meg a böngészőben!

## Események

Amikor a felhasználó megnyom egy billentyűt, vagy rákattint valamire, az egy esemény, amelyhez hozzárendelhetjük egy függvény végrehajtását.

1. Egészítsd ki a JavaScript kódot így:  
     
   Próbáld ki!
2. Ha csak egy eseményhez szeretnénk hozzárendelni egy függvényt, akkor így rövidíthetünk:   
   Ebben az esetben a függvénynek nem lesz neve. Próbáld ki!

## Önálló feladatok

1. Készíts egy másik gombot Töröl felirattal (gomb2 azonosítóval), amely törli a weblapon a bekezdés szövegét! (Vagyis helyettesíti azt az üres szöveggel, amely két egymás utáni idézőjellel adható meg.)
2. Próbáld ki úgy is, hogy nem üres szövegre cseréled a bekezdés szövegét, hanem nem törhető szóközre (&nbsp;)! Mi a különbség?