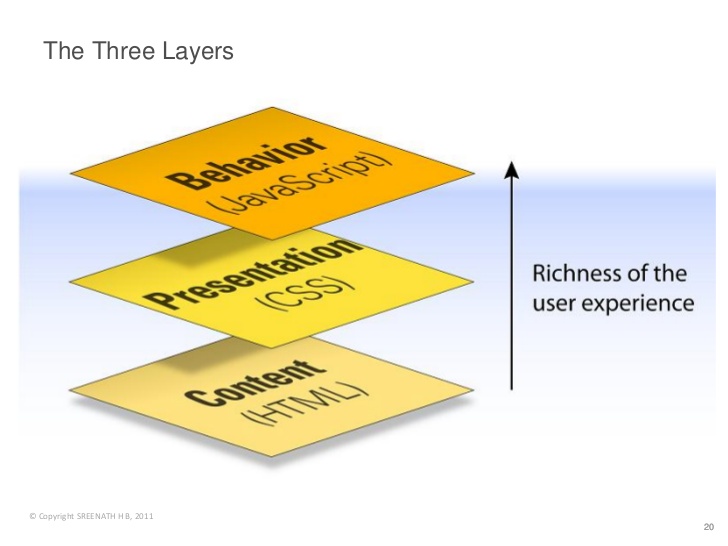
# JavaScript 1.

## Bevezetés

Az eddig elkészített weblapok statikusak, függetlenek a felhasználó tevékenységétől. A következőkben a JavaScript segítségével olyan weblapokat készítünk, amelyek interaktívak, reagálnak a felhasználó tevékenységére.

A weblapok tartalmát a HTML nyelv segítségével írtuk le, a formázást pedig a CSS használatával adtuk meg. A JavaScript az oldal viselkedését, működését határozza meg.

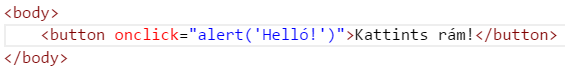


## Előkészületek

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!

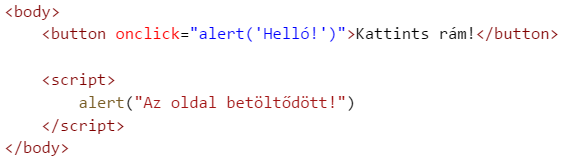
## Inline JavaScript

Első példánkban egy HTML elembe fogjuk írni a JavaScript kódot. Ezt nevezik inline JavaScriptnek.

1. Módosítsd a button elemet így:  
     
   Frissítsd az oldalt, majd kattints a gombra!

## Belső JavaScript

Következő példánkban is a HTML kódba fogjuk írni a JavaScript kódot, de nem egy HTML elem belsejébe, hanem külön. Ez jobb megoldás, mint az előző.

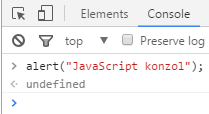
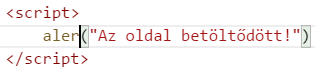
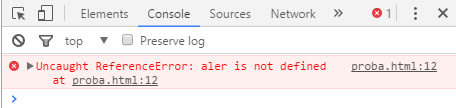
1. Egészítsd ki a HTML kódot így:  
     
   Próbáld ki! Mikor jelenik meg az üzenet?

A script elemet lehetne az oldal head részébe is tenni, de a body rész végén jobb, mert így csak a többi elem betöltődése után hajtja végre a JavaScript kódot.

A szövegek jelölésére az idézőjelet és az aposztrófot is lehet használni. Az első példában mindkettőt használtuk. Figyeld meg, hogyan!

## A JavaScript konzol

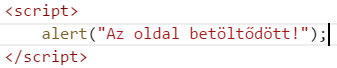
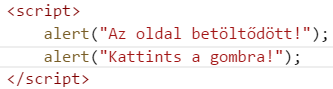
A JavaScript konzol a böngésző fejlesztői eszközei között található. Két dologra lehet használni: JavaScript parancsok kipróbálására és hibakeresésre.

1. Nyisd meg a böngészőben a fejlesztői eszközöket (F12), és lapozz a Console-ra!
2. Írd be a következő parancsot, majd indítsd el!  
     
   Az undefined szó azt jelzi, hogy az alert függvény nem számolt ki, nem adott vissza értéket.
3. Állíts elő egy hibát az oldal JavaScript kódjában, majd próbáld meg frissíteni az oldalt!  
     
   Az oldal betöltődését jelző üzenet nem jelenik meg, de az oldalon ezen kívül nem látszik semmi.
4. Nézd meg a konzolt! Mit lehet látni benne?  
   
5. Javítsd ki a hibát!  
   Ha egy oldalon nem működik a JavaScript program, akkor a hibaüzenetet a knzolon kell keresni!

## Utasítások elválasztása

Ha több utasítást szeretnénk egymás után írni, az utasítások elválasztására két lehetőségünk van. Vagy külön sorba írjuk őket, vagy pontosvesszőt teszünk közéjük.

Ha külön sorba írjuk őket, akkor is kitehetjük a pontosvesszőt. Ez nem kötelező, de ajánlott, ezért mi minden utasítás után fogunk pontosvesszőt írni.

1. Tedd ki a pontosvesszőt az alert utasítás után!  
   
2. Egészítsd ki a programot még egy utasítással!  
     
   Próbáld ki!

Még egy fontos dolog: a JavaScriptben nem számít, hogy az egyes utasításokat behúzással írjuk-e vagy sem, hagyunk-e közöttük üres sorokat vagy sem. Ezeket csak a program áttekinthetősége érdekében fogjuk alkalmazni.

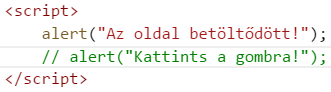
## Megjegyzések

A kódban elhelyezett megjegyzések két esetben is hasznosak:

* Segítenek a kód megértésében (másoknak is, és később nekünk is).
* A megjegyzésbe tett utasításokat ideiglenesen kihagyhatjuk a programokból.

Egysoros megjegyzéseket a // segítségével készíthetünk. A // után a sor végéig minden megjegyzés lesz.

Többsoros megjegyzésnél az elejére /\*-ot, a végére pedig \*/-t írunk. (Ugyanúgy, mint a CSS-ben.)

1. Kapcsold ki ideiglenesen az oldal betöltődése után megjelenő második üzenetet!  
   
2. Próbáld ki, majd kapcsold vissza az üzenetet!

## Javascript kód futtatása

A böngésző elolvassa a kódot, lefordítja futtatható gépi kódra, majd futtatja azt. Nincs szükség arra, hogy a futtatás előtt egy külön programmal lefordítsuk a kódot. Az ilyen programokat script-eknek nevezik.

Bár a JavaScript programokat a böngészőn belüli futtatásra tervezték, ma már külön is futtathatók. Ehhez telepíteni kell a gépre a Node.js programot, amely ezt lehetővé teszi.

A böngészőn belüli és kívüli futtatás között vannak különbségek. A böngészőn belül a biztonságra törekednek, ezért például nem lehet tetszőleges fájlokba írni, alkalmazásokat futtatni.

Ezzel szemben a külső futtatáskor nincsnek ilyen korlátozások. Így a Node.js segtségével akár szervert is készíthető JavaScriptben.