# Visszaszámláló

## Bevezetés

Ebben a leckében egy tárhelyszolgáltató akciós ajánlatait bemutató oldalt készítünk:



Az oldal alján a visszaszámláló azt mutatja, hogy mennyi idő van még hátra az akcióból.

## Előkészületek

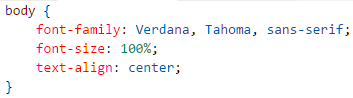
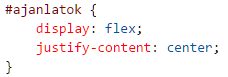
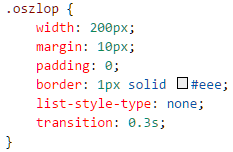
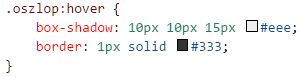
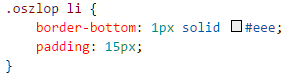
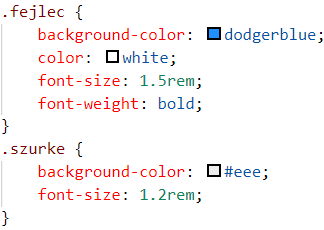
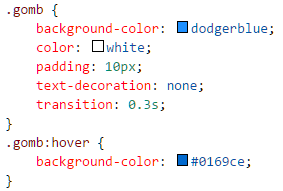
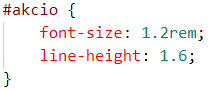
1. Nyisd meg az *akcio* mappát, majd az abban lévő fájlokat a Visual Studio Code-ban!
2. Nyisd meg az oldalt a böngészőben is!
3. Nézd át az eddig elkészített HTML kódot! Figyeld meg, milyen részekből áll az oldal! Milyen azonosítók és osztályok vannak az oldalon? Javítsd ki a dátumot olyanra, ami még nem múlt el!

## Formázás

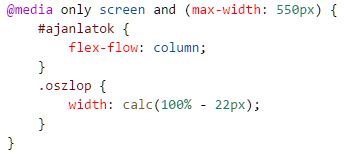
Gondold végig, milyen formázási beállításokra van szükség az egyes elemeknél!

Most az asztali változatot készítjük el először (amely a fenti ábrán is látszik), majd a utána az 550 képpontnál keskenyebb kijelzőkhöz szánt változatot.

A következő formázásokat a *stilus.css* fájlban végezd el, majd próbáld ki őket különböző ablakszélességekkel!

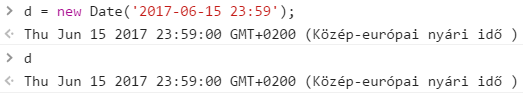
1. A szöveg Verdana, Tahoma vagy talpatlan betűkkel jelenjen meg az alapértelmezett méretben, és legyen középre igazítva!  
   
2. Az ajanlatok rész legyen flex tároló, középre igazított tartalommal!  
      
   Próbáld ki! Mi változott?
3. Az oszlop osztályra állítsd be a következőket:  
     
   Mit jelentenek ezek a beállítások? Próbáld ki!
4. Amikor az egérmutatót valamelyik oszlop fölé visszük, így változzon meg:  
     
   Próbáld ki!
5. Az oszlopokban lévő felsorolások elemeire a következőket állítsd be:  
     
   Próbáld ki!
6. Az oszlopok tetején van egy-egy fejlec és szurke osztályú elem. Ezek stílusai:  
     
   Próbáld ki őket!
7. A gombokat az alábbiak szerint formázd meg! Az egérrel rámutatva változzon sötétebbre a háttérszínük:  
     
   Próbáld ki!
8. Az akcio azonosítójú rész formázása:  
     
   Írd be, majd próbáld ki!

550 képpontos viewport szélességtől lefelé a következők változzanak meg:

1. Az oszlopok ne egymás mellé, hanem egymás alá kerüljenek!
2. Az oszlopok a margókkal és a szegélyekkel együtt töltsék ki a teljes szélességet!  
     
   Figyeld meg, hogyan számoltuk ki az oszlopok szélességét!

## A script elkészítése

Az előző leckében azt tanultuk, hogy egy új Date típusú objektum létrehozásakor az éppen aktuális dátum és idő kerül bele. Ha viszont a zárójelek között megadunk neki egy dátumot és időt, akkor az kerül bele.

Próbáld ki a következőket a konzolon:  


A getTime() metódus egy Date objektumban tárolt dátumot és időt egy számmá alakít. Ez a szám azt adja meg, hogy hány ezredmásodperc (ms) telt el 1970.01.01. 0 órától az adott időpontig.

Próbáld ki a konzolon:  


A jelenlegi idő lekérdezésének van egy rövidebb formája is, amely nem hoz létre egy új objektumot, csak megadja az eltelt időt, így gyorsabban fut:



Próbáld ki ezt is!

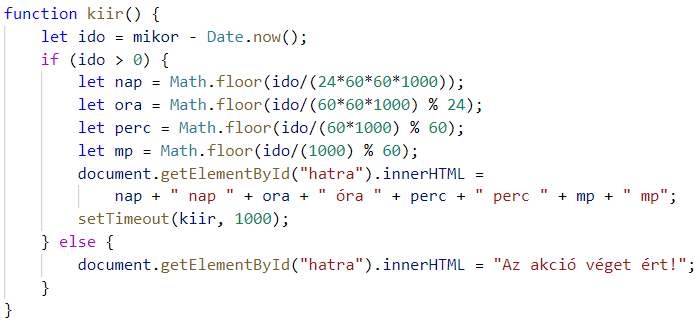
Az akcióból hátralévő időt úgy fogjuk kiszámítani, hogy:

* a jelenlegi időt és az akció végének idejét is átszámíttatjuk ezredmásodpercekre, és
* vesszük a kettő különbségét, majd
* a különbségből kiszámoljuk a hátralévő napokat, órákat, perceket és másodperceket.

Az időzítést úgy fogjuk megoldani, hogy:

* készítünk egy függvényt (kiir),
* ebben megvizsgáljuk, hogy van-e még hátra idő az akcióból,
* ha van hátra idő, akkor azt kiíratjuk, és 1 másodperc múlva a függvény újra meghívja magát,
* ha letelt az idő, akkor ezt kiírjuk, és nem hívja meg magát többször a függvényt.

A scriptet a *vissza.js* fájlban készítsd el!

1. A sriptben szükség lesz egy globális változóra, amely az akció végének idejét tárolja. Ide **olyan dátumot és időt adj meg, amely még nem járt le**:  
   ****
2. A kiir() függvény a következő lesz:  
     
   Gondold végig lépésenként a függvény működését! Figyeld meg, hogyan számoltuk át az ezredmásodperceket napokra, órákra, … !
3. Az oldal betöltésekor induljon el a kiir függvény!  
     
   Próbáld ki, és ellenőrizd a hátralévő időt!