# Objektumok

## Bevezetés

Korábban már használtunk beépített JavaScript objektumokat (Math, Date, Audio). Ebben a leckében megnézzük, hogyan készíthetünk saját objektumokat.

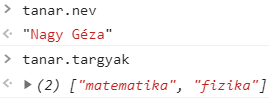
Tegyük fel, hogy egy JavaScript programban egy tanár adatait szeretnénk tárolni. Szükségünk van a nevére (szöveg), a születési évére (szám), arra, hogy van-e egyetemi végzettsége (logikai érték) és az általa tanított tárgyakra (tömb).

A tanár adatait együtt szeretnénk tárolni. Ehhez létrehozhatunk egy objektumot (próbáld ki a konzolon):



Ahogy az ábra alsó részén látható, az objektumra a változó nevével hivatkozhatunk. A változó valójában csak az objektum címét tartalmazza.

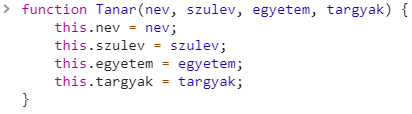
A tanar objektum egyes tulajdonságaihoz így férhetünk hozzá:



Van egy másik módszer is. Ezt ritkábban használjuk, de néha előnyös, mert a tulajdonság neve egy kifejezés eredménye is lehet:



Sok ugyanolyan objektum létrehozását megkönnyíthetjük egy konstruktor függvénnyel:



Figyeld meg, hogy a konstruktor függvény nevét nagy kezdőbetűvel kell írni!

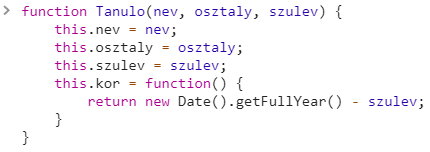
A konstruktor függvénnyel így hozunk létre új objektumot:



Az objektumoknak lehetnek saját függvényeik is, amelyeket metódusoknak nevezünk.

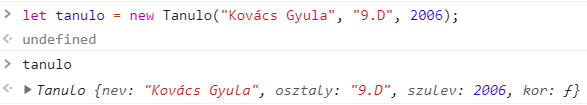
Például most olyan objektumokat készítünk, amelyek egy-egy tanuló adatait tárolják (név, osztály, születési év), és lesz egy olyan metódus is, amely a tanuló korát adja meg években.

Ehhez a következő konstruktor függvényre van szükség:



Figyeld meg, hogyan határozzuk meg az aktuális évet! Létrehozunk egy új Date objektumot (ennek konstruktorát beépítve tartalmazza a JavaScript), és meghívjuk ennek getFullYear() metódusát.

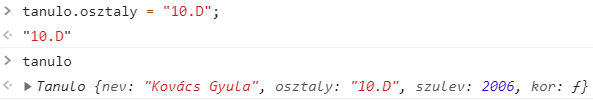
Hozz létre egy új tanulót a konstruktor függvény segítségével:



Írasd ki az előbb létrehozott tanuló korát (az eredmény változhat az aktuális évtől függően):



Az objektum tulajdonságait a létrehozás után módosíthatjuk. Ha a tanuló a következő évben a 10.D osztályba kerül, akkor ezt így adjuk meg:



Új tulajdonságot is megadhatnánk, és ez hozzáadódna az eddigiekhez, de ez nem egy jó gyakorlat, ezért nem tesszük.

A JavaScriptben a tömbök is objektumnak számítanak, ezért nincs külön tömb adattípus.

A HTML oldal elemei is objektumok, ezért tudtuk a metódusaikat használni (pl. innerHTML, …).

## Előkészületek

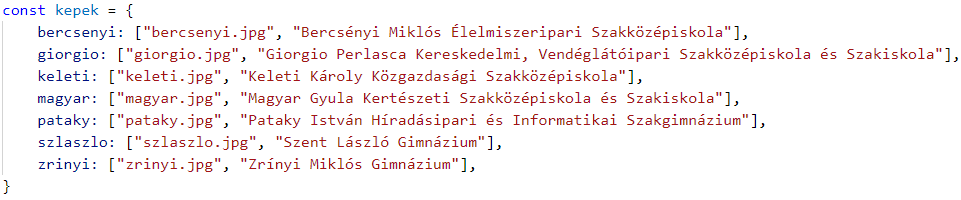
1. Nyisd meg a *kobanya* mappát a Visual Studio Code-ban, és nyisd meg a fájlokat a mappában!
2. Nézd át a HTML kódot! Figyeld meg a jQuery importálását! Melyik iskola képe látszik alaphelyzetben?
3. Nézd meg az oldalt a böngészőben!

## Képek cseréje

Azt szeretnénk, hogy ha az egérmutatót egy iskola neve fölé visszük, akkor a lenti kép azt az iskolát mutassa.

Ehhez először létrehozunk egy objektumot, amely azt tárolja, hogy az egyes azonosítókhoz milyen képfájl, és milyen alternatív szöveg tartozik.

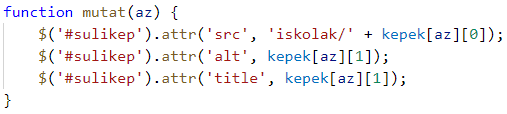
Írd ezt a *kobanya.js* fájlba:



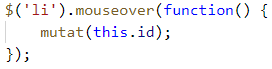
Figyeld meg, hogy itt az objektumot egy tömbhöz hasonlóan használjuk, csak az egyes elemekre nem indexekkel hivatkozunk!

Az egyes azonosítókhoz tartozó értékeket egy-egy tömbbe tettük.

Ezután készítünk egy mutat nevű függvényt, amely a sulikep azonosítójú image elem attribútumait módosítja. Paraméterként megkapja a mutatott elem azonosítóját.



Végül minden eseményhez, amikor egy li elem fölé visszük az egérmutatót, hozzárendeljük eseménykezelőként a mutat függvényt, és parméterként átadjuk az elem azonosítóját:



Próbáld ki a böngészőben! Szükség esetén javítsd a hibákat!

Kapcsold be a fejlesztői eszközöket, és az Elements lapon jelöld ki a képet! Után vidd az egérmutatót az egyes iskolák neve fölé, és nézd meg, hogyan változik a HTML kód!