

Szkriptnyelvek - JavaScript ismertető

- A programot JavaScript nyelven kell megírni.
- A benyújtandó fájl neve: `feladat.js`
 - Egy JavaScript nyelven írt, szöveges fájl (nem zip, rar, stb.)
 - Ez csak a feladatban kért dolgokat tartalmazza! Amennyiben saját inputtal teszteléd a kódot lokálisan, úgy feltöltés előtt a tesztelő kódrészletet kommenteld ki!
- A megoldást Bíró2 webes felületén (<https://biro2.inf.u-szeged.hu>) keresztül kell benyújtani és a megoldást a Bíró fogja kiértékelni.
 - A Feladat beadása felületen a *Feltöltés* gomb megnyomása után ki kell várni, amíg lefut a kiértékelés. **Kiértékelés közben nem szabad az oldalt frissíteni vagy a Feltöltés gombot újból megnyomni** különben feltöltési lehetőség veszik el!
- Feltöltés után a Bíró a programot **Node** interpreterrel fogja futtatni, és különböző tesztesetekre futtatja.
- A program működése akkor helyes, ha a tesztesetek futása nem tart tovább 2 másodpercnél és hiba nélkül fejeződik be, valamint a program működése a feladatkiírásnak megfelelő.
- Ha 3 teszteset futási ideje túllépi a fenti időkorlátot, a tesztelés befejeződik, a pontszám az addig szerzett pontszám lesz.
- A riport.txt megtekinthető az alábbi módon:
 1. Az Eredmények megtekintése felületen a vizsgálandó próba új lapon való megnyitása
 2. A kapott url formátuma:
https://biro2.inf.u-szeged.hu/Hallg/IB370G/FELADAT_SZAMA/hXXXXXX/4/riport.txt
 3. Az url-ből visszatörölve a 4-esig (riport.txt törlése) megkaphatók a 4-es próbálkozás adatai
- A programot 25 alkalommal lehet benyújtani, a megadott határidőig.
- A munkád során figyelj arra, hogy pontosan kövesd a feladatban leírtakat, az elnevezéseket!
- A fájl elejére kommentbe írd be a neved, Neptun és h-s azonosítód az alábbi formában:

```
// Nev: vezeteknev keresztnév  
// Neptun: NEP4LF  
// h: h123456
```

Nyuszi a vendéglátó

A Százholdas Pagonyban Nyuszi szereti a barátait vendégül látni és minden vendége 1 répát kap. A répát viszont ültetni kell. Amennyiben nincs elegendő répája, akkor a vendégeknek várakoznia kell.

Hozz létre egy `Nyuszi` nevű osztályt, amely a következőt csinálja:

- Legyen két adattagja: **_repa**, amely egy egész számot tárol (a jelenlegi répák számát jelzi), valamint egy **vendegek** nevű tömb.
- Nyuszi példányosításakor meg lehet adni, hogy épp hány répája van. Ha nem adnánk meg, akkor 0 répája van.
- A `_repa` adattagjához hozzuk létre a gettert, amely visszaadja épp hány répája van.

- Nyuszi tud répát ültetni az `ültet` nevű függvénnyel, amelynek paraméterként megadhatjuk, hogy hány répát ültetett. Ha nem adjuk meg, akkor egyet ültetett. Az ültetés során megadott mennyiség egyből rendelkezésre áll.
- A `vendeg` függvény segítségével új vendéget fogad Nyuszi. Ez csak string paramétert fogadhat és a vendég nevét tartalmazza. Egy vendég akár többször is betérhet.
- Nyuszi az `etet` függvénnyel kedveskedik a vendégeknek. Ekkor, ha van nála vendég, megpróbál nekik répát adni (amennyi rendelkezésre áll). Aki kapott répát, az el is köszön Nyuszitól és haza megy. Mindig a legrégebben érkező vendég kap először répát. Ha elfogyott a répa, akkor a vendégeknek várniuk kell. Ha nincs vendég, akkor a répa megmarad és nem fogy.

Példa

```
// nincs répája Nyuszinak.  
var nyusz = new Nyuszi(0);  
//ültet egy répát.  
nyusz.ültet(1);  
console.log(nyusz.repa); // 1.  
console.log(nyusz.vendegek); //még nincs vendége az eredmény [].  
nyusz.vendeg('Robert Gida'); // Róbert Gida megérkezik Nyuszihoz.  
nyusz.vendeg('Malacka'); // Malacka megérkezik Nyuszihoz.  
console.log(nyusz.vendegek); ['Robert Gida', 'Malacka']  
nyusz.etet(); // Mivel van egy 1 répa, Robert Gida kap egyet és utána haza megy.  
Malackának már nem jutott répa, ezért neki várnia kell.  
console.log(nyusz.repa) // 0  
console.log(nyusz.vendegek); // ['Malacka']
```