



## Félkarú rabló – 27 pont


A következő feladatokban egy félkarú rabló játékot kell készítened! A megoldás során használd fel az `index.html` és a `style.css` kiinduló állományokat, és csak a `script.js` fájlba dolgozz!


A feladatok megoldása során a változók neveit (a tiszta kód elveit lehetőség szerint követve) tetszőlegesen megválaszthatod, de a **függvényeket pontosan a leírt módon nevezd el**. Ügyelj rá továbbá, hogy minden függvény **paraméterezése** és **visszatérési értéke** szintén megfeleljen a leírásnak! (A paraméterek neve nem fontos, a számuk és sorrendjük viszont igen!)


 **Félkarú rabló** 

Tét:

Játék indítása







Statisztikák mutatása/elrejtése!

Legutóbbi nyeremény: 0    Aktuális pénz: 11400

 **Félkarú rabló** 

Tét:

Játék indítása







Statisztikák mutatása/elrejtése!

Legutóbbi nyeremény: 500    Aktuális pénz: 11800

A játék lényege, hogy a felhasználó választ valamekkora tétet, majd a „Játék indítása” gombra kattint. Ezután a gép sorsol 3 darab gyümölcsöt. Ha egyformákat kapott a felhasználó, akkor a feltett tétje 5-szörösét kapja nyereményként, minden egyéb esetben semmit sem kap.

Például ha 12 000 Ft-já van, amiből feltesz 300 Ft-ot, amellyel nyer, akkor utána  $12\,000 - 300 + 1500 = 13\,200$  Ft lesz a teljes pénze.

1. Érdemes néhány változót globálisan deklarálni, hogy minden függvény során elérhetőek legyenek. Ezek a következők:

- `slots`: Tömb, amely a sorsolt gyümölcsök neveit tárolja stringként. Kezdetben üres.
- `totalMoney`: Mennyi pénzünk van jelenleg? Kezdeti értéke: 12000.

Javasolt még a weboldalon található egyetlen `<ul>` elemet globális konstansként eltárolni, mert több függvénynek is szüksége lehet rá!

**1 pont**

2. A “Játék indítása” gombra kattintáskor egy `startGame()` nevű eseménykezelő függvény fusson le! Olvassa ki a megadott tétet, és ellenőrizze annak helyességét! Ne fogadj el olyan tétet, ami kisebb 100-nál vagy több a jelenlegi pénzünknel!

Hiba esetén írd ki a konzolra, hogy hiba történt, és mást ne tegyél! Helyes tét esetén csökkenjen a felhasználónál lévő összeg, és hívj meg egy `rollSlots(bet)` nevű eljárást, amelynek átadod a tétet paraméterként! (Az eljárás törzsének megírására majd a 4. feladatban kerül sor.) **6 pont**

3. Készíts `getFruit()` néven egy paraméter nélküli függvényt, amely visszatérési értéke egy gyümölcs neve a “cherry”, “lemon” és “orange” szövegek közül véletlenszerűen választva (egyforma valószínűséggel). **3 pont**

4. Írj eljárást `rollSlots(bet)` fejléccel, amely 3-szor meghívja a `getFruit()` függvényt, és a kapott gyümölcsökhöz tartozó képeket megjeleníti egy-egy lista elemként a felhasználó számára! A gyümölcsök neveit tárold el a `slots` tömb elemeként is!

Ügyelj rá, hogy akkor is helyesen működjön az eljárás, ha egymás után többször szeretnénk játszani! **5 pont**

5. Csinálj `calculateWinning(bet)` függvényt, amely megadja a nyeremény értékét! Ha nem nyertünk, akkor térjen vissza 0 értékkel, egyébként a tét 5-szörös értékével! Hívd meg a függvényt a megfelelő helyen, és tárold el az értékét! Ne felejtse el növelni a nálunk lévő pénz értékét sem a megnyert összeggel! **2 pont**

6. Hozz létre egy `showStats(winning)` eljárást, amely paraméterként megkapja az aktuális játék nyereményét, és beírja az oldal alján található statisztikák megfelelő részébe a szükséges adatokat (aktuális nyeremény és aktuális pénzünk).

Szintén ezen eljárás feladata, hogy győzelem esetén az `<ul>` elemre alkalmazza a `winner` css osztályt! (Vigyázat: csakis győzelem esetén legyen rajta az osztály!) **3 pont**

7. Írj `saveGame(bet, winning)` eljárást, amely egy JavaScript objektum segítségével egységbe zárja az aktuális játékot leíró változókat. A JS objektumnak 3 adattagja legyen: `bet`, `winning` és `money`.

Például ha 1000 Ft tétet tettünk fel, nyertünk 5000 Ft-ot és így most van 17 000 Ft pénzünk, akkor: `game = { bet: 1000, winning: 5000, money: 17000 }`;

Az aktuális játék adatait “game” kulcs alatt tárold el `localStorage`-ban! **3 pont**

Az 5-7. feladatok függvényeit hívd meg a `rollSlots` függvény végén! Erre külön pontszám nem jár, de a program helyes működéséhez szükséges!

8. Az oldal betöltésekor olvasd ki a `localStorage`-ban előzőleg eltárolt értéket, majd jelenítsd meg a statisztikákat a betöltött értékekkel! (Ehhez természetesen az előzőleg megírt `showStats` eljárást felhasználhatod.) Ügyelj rá, hogy az oldalra történő első látogatás esetén se kapjunk hibát! **4 pont**

**Bónusz! Az utolsó feladat csak bónusz! Ha a dolgozat jeles és a bónusz feladat is helyes, akkor dupla webprog jeles jár!**

9. Az oldal alján található “*Statisztikák mutatása/elrejtése!*” gombra kattintva az alatta lévő statisztikák megjelenése változzon meg. Ha eddig láthatóak voltak, a továbbiakban ne legyenek (és fordítva)! A megoldáshoz felhasználható a `hidden` nevű css-ben definiált osztály!