AKTUÁLIS ADATOK MEGJELENÍTÉSE

Feladat

Egészítsd ki az időjárás widgetet úgy, hogy hőmérséklettől függően különböző ajánlatokat is jelenítsen meg!

A hamburgerező úgy döntött, hogy hőmérséklettől függően minden nap más ajánlattal várja a vendégeket. Ezek a következők:

* 0 °C alatt: forró csoki
* 0 °C - 15 °C: meleg tea
* 15 °C - 20 °C: finom süti
* 20 °C - 25 °C: fagyi
* 25 °C-tól: jéghideg limonádé

1. Vegyél fel két új tömböt a JavaScript fájlodba! Az első tartalmazza a hőmérséklet kategóriák felső határait, a második a megjelenítendő ajánlatokat. Ez utóbbiakat tedd vendégcsalogató mondatokba!
2. A hőmérséklet alatt jelenítsd meg az adott napi hőmérséklethez tartozó ajánlatot!

Ciklusok

Folyamatosan ismétlődő programrészletek. Egy ciklus mindaddig ismétlődik amíg a feltétele igaz. Olyan, mintha azt mondanánk, addig csináld ezt, amíg igaz, hogy ...

For

A for ciklus minden programnyelvben megtalálható.

Három paramétere van:

* ciklusváltozó: egy Number típusú változó, minden futás után módosítod az értékét.
* feltétel: Boolean, amíg ez a feltétel igaz (a ciklusváltozó értékvizsgálata), addig fut a ciklus.
* léptetés: minden ciklus végén lefut, itt kell a ciklusváltozót növelni/csökkenteni.

**Ciklusmag:** az az utasítás vagy utasítások, amelyek a feltétel teljesülése esetén lefutnak.

Az alábbi példa egy for ciklust mutat be:

Elöltesztelős ciklus

Először megvizsgálod, hogy a feltételed igaz-e, és ha igen, akkor lefut a ciklusmag. Tehát a for egy tipikusan elöltesztelős ciklus.

Fontos megjegyezni, hogy egy for ciklust bármikor átírhatsz while ciklusba. Tehát amit forral megvalósíthatsz azt while-lal is, csupán a szintaxis más. A while ciklussal később majd foglalkozunk, addig is a legtöbb feladatra tökéletes lesz a for.

Tömb bejárása ciklussal

Ha szeretnéd egy kollekció összes elemét végigjárni, és valamilyen vizsgálatot vagy műveletet végezni velük, akkor általában az első elemtől indulsz, lefut a ciklusmag, és növeled az iterátort. Ezt mindaddig kell ismételned, amíg a kollekció végére nem érsz, tehát ismerned kell hozzá a hosszát.

Tömb bejárása for ciklussal:

continue és break

continue : ha valamiért úgy döntesz, hogy mégsem akarod lefuttatni a teljes ciklusmagot, akkor a continue utasítással tovább tudsz ugrani a következő iterációra.

break : a break utasítás nem a következő iterációra ugrik, hanem megszakítja a ciklust, azaz a ciklusmag többet nem kerül végrehajtásra, úgy is mondhatnánk, a program továbbmegy.

Continue és break használata:

# **Összetett operátorok**

## Compound operators

Ez azt jelenti, hogy összetett operátorok. Itt arra utalnak, hogy a több operátort raksz egymás után és ezzel gyorsítod a kódolást.

Lista

A fontosak, amiket gyakran használsz majd

* Hozzáadás: x = x + 10  helyett x += 10
* Kivonás: x = x - 10  helyett x -= 10
* Osztás: x = x / 2 helyett x /= 2
* Modulus (maradék): x = x % 2 helyett x %= 2
* Szorzás: x = x \* 10  helyett x \*= 10

Exotikum (nem kötelező tudni, csak agybővítésre)

Nézzünk egy példát a bitwise azaz bitenkénti és műveletekre. Át kell váltani 2-es számrendszerbe és 32 biten ábrázolni a számokat majd elvégezni rajtuk az and & műveletet:

* a 9: 00000000000000000000000000001001
* a 14: 00000000000000000000000000001110
* ott lesz egy, ahol mindkettőben egy van: 00000000000000000000000000001000
* ha ezt visszaváltod 10-es számrendszerbe akkor az eredmény: 8
* leírva ha x = 9 és y = 14: x &= y az 8 lesz. Ennyi.

A ternary operátor

Egyszerűsítsünk!

Ha már az értékadásnál összevontuk az operátorokat, miért ne vonnánk össze az if...else esetén is?

Ternary operator

Ezt magyarul három operandusos kifejezésnek is szokták hívni.

Szintaxis:

let variablename = (condition) ? value1: value2

Szóval ha a kondíció igaz, akkor az első, ha nem akkor a második érték kerül bele a változóba.

Részei

* **változó neve:** a változó neve
* **értékadás:** = ezt ismered, értéket adunk egy változónak
* **feltétel:** 4 > 5 lehet bármilyen kifejezés, amit Boolean-ként ki lehet értékelni
* **?** a kérdőjel után jön az az érték, ami a kifejezés true értékéhez tartozik
* **:** a kettőspont után épedig az, ami a false értékhez tartozik

Példák

Hagyományos if...else módszerrel

let age = 33;

let voteable;

if (age < 18) {

voteable = "Too young";

} else {

voteable = "Old enough";

}

console.log( voteable );

Ternary operátorral

Az értékadás 5 sor helyett kemény 1 lett:

let age = 33;

let voteable = (age < 18) ? "Too young": "Old enough";

console.log( voteable );

Azt fogja visszaadni, hogy "Old enough" azaz elég idős. Azért, mert az age nem kisebb mint 18, a feltétel nem igaz és így a második érték kerül a voteable változóba.