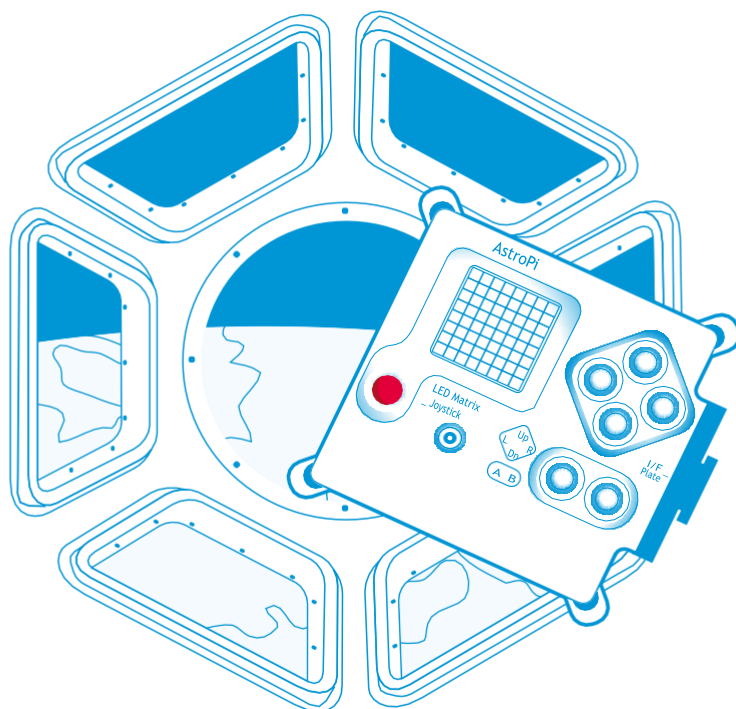


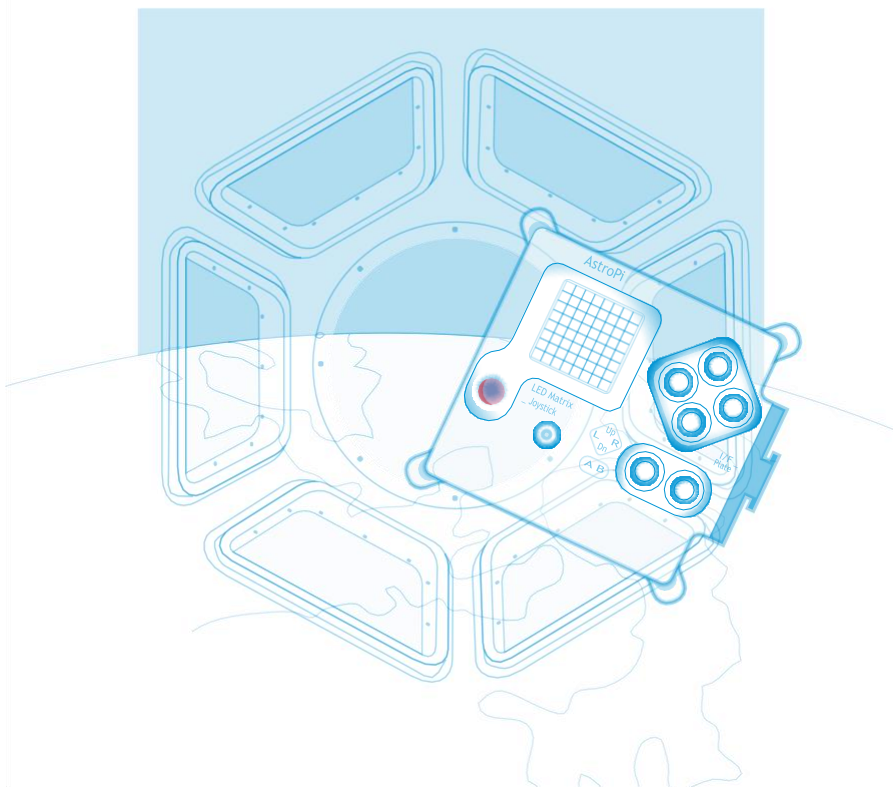


Tanítsunk a világűrrel!

→ Astro Pi kezdőknek

Tanuljunk programozni a Raspberry Pi-vel!





Az Európai Astro Pi Kihívás

3. oldal

1. tevékenység – Mi is az a Raspberry Pi?

4. oldal

2. tevékenység – Ismerd meg Pythont!

6. oldal

3. tevékenység – Készíts kvízt Pythonnal!

8. oldal

4. tevékenység – Rajzolj geometriai alakzatokat Pythonnal!

10. oldal

Üdvözlünk az Európai Astro Pi Kihíváson!

Az Astro Pi egy kis számítógép neve, amelyet a Raspberry Pi Alapítvány fejlesztett ki az Egyesült Királyság Űrügynökség közösen az Európai Űrügynökséggel (European Space Agency, ESA) együttműködésben.

Két nagyon különleges Astro Pi létezik. A nevük Ed és Izzy, és űrrepülésre alkalmasak. Most a Nemzetközi Űrállomás (International Space Station, ISS) fedélzetén vannak, és hozzád hasonló tanulók használják őket.

A tevékenység során először összeállítod a saját Raspberry Pi számítógépedet, majd elsajátítod az Astro Pi programozásához szükséges alapvető programnyelvet. Ezáltal lehetőséged nyílik arra, hogy lefuttasd a saját programodat a Nemzetközi Űrállomáson.



↑ Az Astro Pi logója.

Eszközök

- Astro Pi készlet
- monitor
- billentyűzet USB-csatlakozással
- egér USB-csatlakozással

Alapadatok

Korosztály: 10–14 év

Nehézségi fok: könnyű

Helyszín: tanterem

Eszközök: Astro Pi készlet, monitor, billentyűzet és egér USB-csatlakozással

Rövid ismertetés

A tevékenység keretében a tanulók összeállítanak és beprogramoznak egy Raspberry Pi számítógépet. A cél az alapvető programozási készségek elsajátítása.

Tanulási célok

- Megismerkedünk a Raspberry Pi alapvető tulajdonságaival, összeállításával és használatával.
- Megtanuljuk, mi a különbség hardver és szoftver között.
- Megtanuljuk, mi az a programnyelv.
- Gyakoroljuk a programozást Python használatával.
- Megismerkedünk a bemeneti és kimeneti üzenetek, változók, adattípusok, „if” (magyarul: ha) parancsok és ciklusok használatával.
- Gyakoroljuk a geometriai alakzatok rajzolását a Python Turtle moduljának használatával.

→ 1. TEVÉKENYSÉG: MI IS AZ A RASPBERRY PI?

A Raspberry Pi egy olyan számítógép, amely méretét és kinézetét leszámítva nagyban hasonlít azokra a számítógépekre, amelyeket mi is ismerünk. Lehet vele internetezni, e-maileket küldeni, és akár át is alakíthatod MP3-lejátszóvá, hogy feldobd valami jó kis zenével az ISS űrhajósainak szabadidejét. A Raspberry Pi azonban ennél sokkal többre is képes! Ezzel a számítógépes eszközzel ugyanis tudományos kísérleteket és adatelemzést is végezhetsz, akárcsak az igazi űrkutatók!

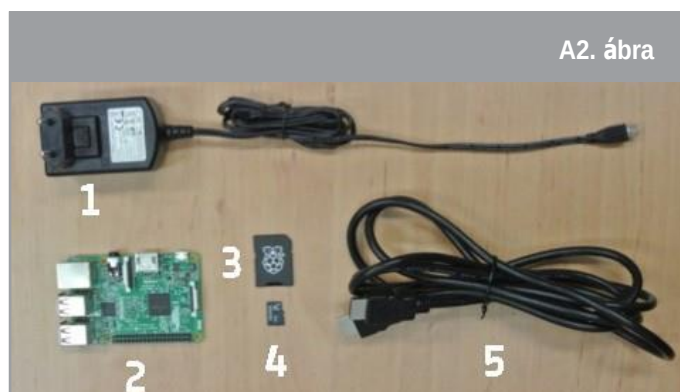
A tevékenység során összeállítod a saját Raspberry Pi számítógépedet, és felfedezed, mi mindenre használható.

Gyakorlat

1. Az Astro Pi küldetésed megkezdéséhez különleges eszközökre lesz szükséged, amelyeket megtalálsz az Astro Pi készletben. Kelleni fog továbbá egy monitor, egy billentyűzet és egy egér. Az A2. ábrán látható az összes szükséges felszerelés.

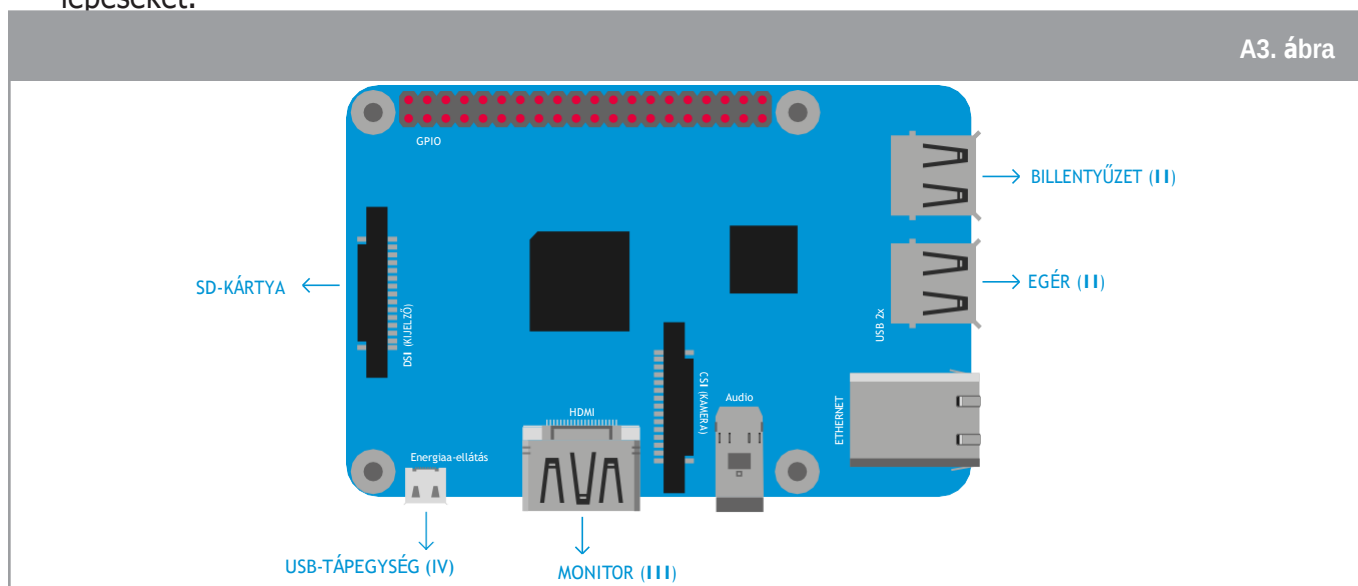
Ezeket a tartozékokat hardvernek nevezzük. A 2. ábra segítségével nevezd meg őket (Micro SD-kártya, SD-kártya adapter, Raspberry Pi 3, tápegység, HDMI-kábel).

1. _____
2. _____
3. _____
4. _____
5. _____



↑ A Raspberry Pi tartozékai.

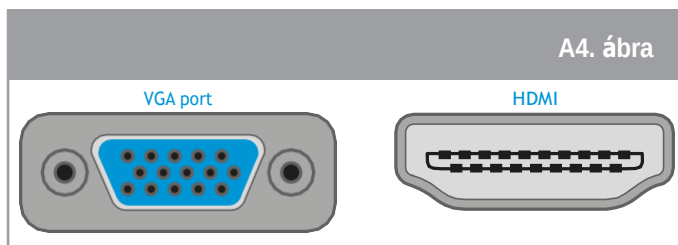
2. Most pedig összeállítjuk a Raspberry Pi-t! Ehhez figyeld meg az A3. ábrát, és kövesd az I–IV. lépéseket.



↑ Így kell összeállítani a Raspberry Pi-t.

- I. Vedd ki az SD-kártyát az SD-kártya adapterből, és helyezd a Raspberry Pi SD-kártya foglalatába. Csak egyféleképpen fog illeszkedni.
- II. Csatlakoztasd a billentyűzetet és az egeret az USB-portokhoz.
- III. Csatlakoztasd a monitor (vagy TV) HDMI-kábelét a Raspberry Pi HDMI-portjához, és kapcsold be a monitort.

TIPP: Ennél a lépésnél mindenképp győződj meg róla, hogy a megfelelő bemenetet választottad ki (pl. HDMI, DVI stb.). Ha a monitor csak VGA-val rendelkezik (lásd az A4. ábrát), akkor használj HDMI-VGA-adaptert (a készletben).



- IV. Csatlakoztasd a micro USB-tápegységet. [↑ VGA- és HDMI-portok](#)

Ezzel bekapcsolod és elindítod a Raspberry Pi-t.

Ez a művelet akár 5 percig is eltarthat, miközben a Raspberry Pi villogni fog, és a monitoron szöveg jelenik meg. Amikor a Raspberry Pi befejezte a betöltést, és a szöveg már nem fut a képernyőn, az A5. ábrán megjelenített grafikus felületet fogod látni.



[↑ Az asztali környezet vagy grafikus felhasználói felület.](#)

3. Most már kipróbálhatod a Raspberry Pi szoftvert. Megnyithatsz például egy webböngészőt, és böngészhetsz az interneten. Ehhez kattints a képernyő tetején lévő főmenüre (a Raspberry logóra), és válaszd az Internet > Chromium Web Browser lehetőséget. Az internetezéshez wifi hozzáférés szükséges. Keresd meg és írd le az ISS aktuális helyét!

4. Ismerkedj meg a Raspberry Pi szoftverrel: válassz ki és nézz meg különféle opciókat a programozási menüben. Mindezek alapján el tudnád magyarázni, mi a különbség hardver és szoftver között?

Megjegyzés: a Raspberry Pi-t nem szabad egyszerűen a csatlakozó kihúzásával kikapcsolni. Amikor befejezted vele a munkát, a főmenüben található Shutdown (Kikapcsolás) gombra kattintva tudod helyesen leállítani.

Kiegészítő feladat

A látszat néha csal! Írj egy listát azokról a digitális eszközökről, amelyeket a mindennapjaid során használsz és számítógépnek tekintesz. Hasonlítsd össze a listád a többiekével, és beszéljétek meg, mitől számítógép egy számítógép.

→ ISMERD MEG PYTHONT!

Ahhoz, hogy a Raspberry Pi segítségével kommunikálhass az ISS űrhajósaival, meg kell adnod neki a szükséges utasításokat. Mint minden számítógép, a Raspberry Pi is csak azt fogja csinálni, amit megtanítasz neki, vagy amire beprogramozod. Ehhez el kell sajátítanod egy programozási nyelvet ugyanúgy, ahogy az anyanyelvedet is meg kellett tanulnod ahhoz, hogy szót érthess a környezetteddel. Az alábbi tevékenység során megismered a Python programnyelv alapvető szókincsét, és megírod az első programodat.

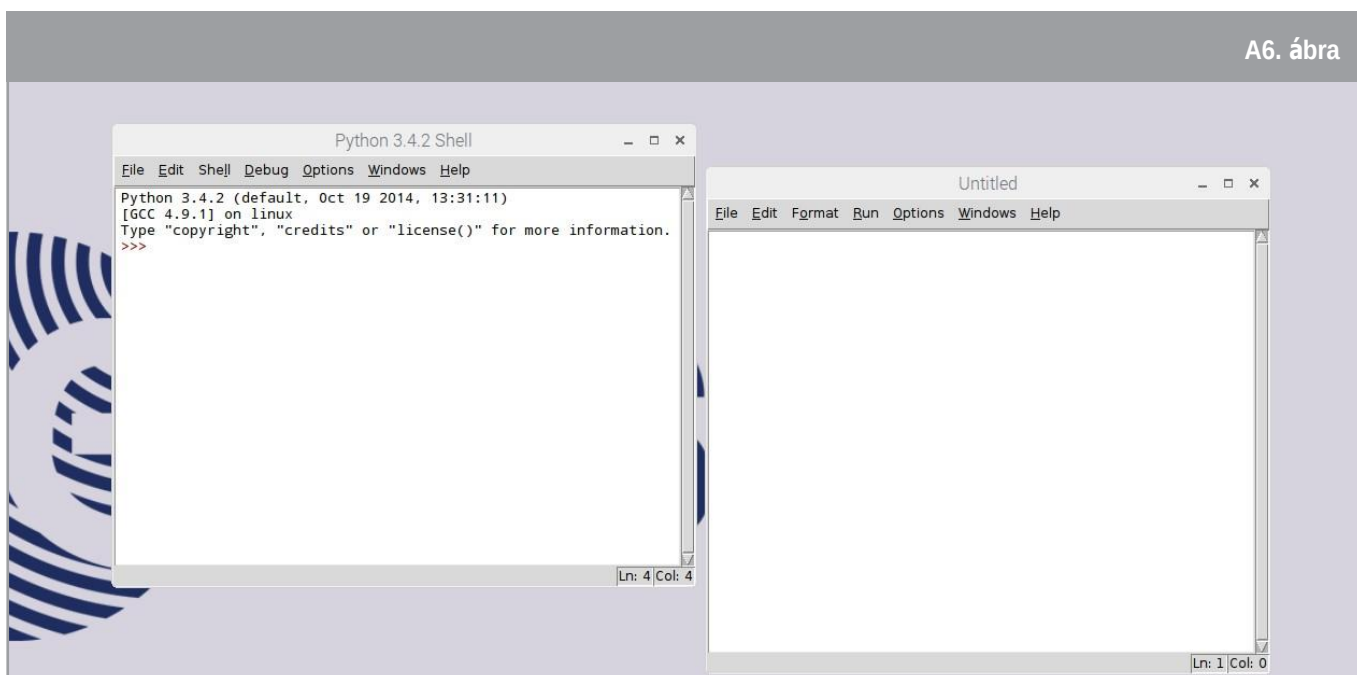
Tudtad?

Mielőtt felmehetnének az űrbe, az űrhajósoknak is el kell sajátítaniuk különböző nyelveket. Például minden űrhajósnak meg kell tanulnia oroszul, mert a Szojuz űrhajó vezérlése oroszul történik. Jelenleg a Szojuz az egyetlen űrhajó, amellyel embereket lehet szállítani az ISS-re és vissza. A Szojuz szó oroszul azt jelenti, „szövetség”. Bár az űrkutatás korszakának hajnalán épült a Szovjetunióban, az azóta korszerűsített Szojuz űrhajót ma is használják.



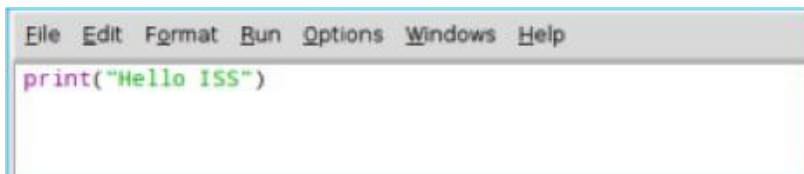
Gyakorlat

1. A Python elindításához kattints a Raspberry Pi-n a képernyő tetején található főmenüre (Raspberry logó), és válaszd a Programming (Programozás) > Open Python 3(IDLE) menüpontot. Ez megnyitja a Python Shell nevű kimeneti ablakot (A6. ábra). Ebben a shellbe közvetlenül is beírhatod a programodat, de egyszerűbb egy szerkesztőablakban dolgozni. Új ablak megnyitásához válaszd a File (Fájl) > New File (Új fájl) lehetőséget. Most már két ablak van nyitva. Az egyikbe a programot írod (a szerkesztőablakba), a másik a kimenet (a Python Shell), például az eredmények, hibaüzenetek stb. megjelenítésére szolgál. Győződj meg róla, hogy mindkét ablak látható.



↑ Python Shell (balra) és szerkesztőablak (jobbra).

2. Kezdheted is a kódolást. Írd be az alábbi kódot a szerkesztőablakba.

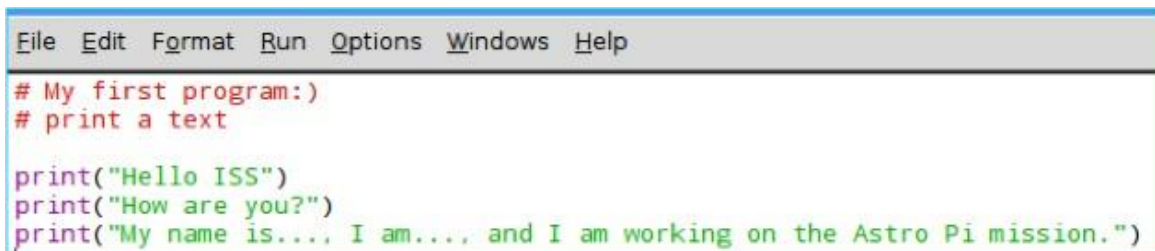


```
File Edit Format Run Options Windows Help
print("Hello ISS")
```

3. Ahhoz, hogy lásd az eredményt, válaszd a Run (Futtatás) > Run Module (Modul futtatása) lehetőséget. A program meg fog kérni, hogy mentsd el a kódot. Nevezd el a fájlot, és futtasd le. Mi történik?
-
-

Megjegyzés: ne felejtse el rendszeresen menteni a munkáját! A mentéshez válaszd a File (Fájl) > Save (Mentés) menüpontot.

4. Itt az ideje, hogy bemutatkozz az ISS űrhajósainak! Másold be az alábbi kódot a fájlodba, és írd be az űrhajósoknak szánt üzenetedet.



```
File Edit Format Run Options Windows Help
# My first program:)
# print a text

print("Hello ISS")
print("How are you?")
print("My name is..., I am..., and I am working on the Astro Pi mission.")
```

Gratulálunk! Elkészítetted az első programodat a Python programnyelven.

Megjegyzés: a számítógép megpróbál mindent megérteni, amit az ablakba írsz. Ha csak magadnak írnál egy rövid megjegyzést, használd előtte a # jelet. A # után írt szöveget a program figyelmen kívül hagyja, és nem próbálja meg lefordítani számítógépes nyelvre.

→ 3. TEVÉKENYSÉG: KÉSZÍTS KVÍZT PYTHONNAL!

Eddig te adtál egyszerű utasításokat a számítógépnek. Most ráveheted a számítógépet, hogy kérdéseket tegyen fel neked, és reagáljon a válaszaidra.

Gyakorlat

1. Nyiss egy új Python 3 ablakot, és írd be az alábbi kódot.

```
File Edit Format Run Options Windows Help
# whenever a word or number follows the '=' symbol, it is called a variable.
# variables are used to store information that the computer must remember.
orbits = input (" How long does it take the ISS to orbith the Earth?")
```

2. Válaszd a Run (Futtatás) > Run Module (Modul futtatása) lehetőséget a kód lefuttatásához. Mi történik?

3. A Python az input () függvényt használja kérdések feltevésére. Tudod, mennyi idő alatt kerüli meg az ISS a Földet? Írd be a Python shellbe.

4. A Python shellt számológépként is használhatod. Íme a példa:

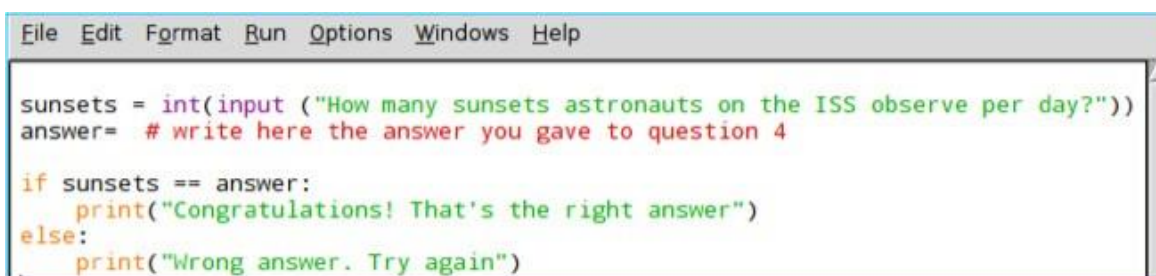
```
File Edit Shell Debug Options Windows Help
Python 3.4.2 (default, Oct 19 2014, 13:31:11)
[GCC 4.9.1] on linux
Type "copyright", "credits" or "license()" for more information.
>>> 12+12
24
>>> 12*2
24
>>> 12/2
6.0
>>> 12*2-12
12
```

Próbálg válaszolni az alábbi kérdésre úgy, hogy a szükséges számításokat a Python Shell segítségével végzed. Írd le, milyen számításokat alkalmaztál.

Naponta hány naplementét látnak az ISS űrhajósai?

5. Most pedig készíts egy tudományos kvízt a Raspberry Pi segítségével. Ehhez ki kell választanod a helyes és a helytelen válaszokat. A Python programnyelvben ezt **kiválasztási** utasításokkal teheted meg, amelyeket **„if (ha) utasításnak”** is nevezünk. Nyiss egy új fájlt, írd be az alábbi kódot és a választ. Szerinted mit fog csinálni a program? Futtasd le a modult, és nézd meg.

Megjegyzés: az **„int”** kódot azért írjuk az **„input”** elé, hogy tudassuk a Pythonnal, hogy a válasz integer szám lesz. Az integer szám egy egész szám, például 5 vagy 10. Ha Pythonban az információt számtípusként akarod kezelni, akkor ezt az utasítást kell megadnod a számítógépnek. Az egész számokon kívül előfordulhatnak lebegőpontos (float) számok is: a tizedesjegyekkel rendelkező számokat hívjuk így, például 5,1 vagy 10,99.



```
File Edit Format Run Options Windows Help
sunsets = int(input("How many sunsets astronauts on the ISS observe per day?"))
answer = # write here the answer you gave to question 4

if sunsets == answer:
    print("Congratulations! That's the right answer")
else:
    print("Wrong answer. Try again")
```

6. A kvízben megadhatsz némi extra segítséget. Közvetlenül az **„else”** (különben) sor előtt írd be az alábbi parancsot, és futtasd le újra a modult. Írj egy megjegyzést, hogy megmagyarázd, mit csinál az **„elif”** (különben, ha) kulcsszó.

#

```
elif sunsets >13 and sunsets <16:
    print("Almost there! Try again")
```

Kiegészítő feladat

Az **„if utasítások”** több dolgot is ellenőrizhetnek, nem csak egyet. Ehhez csak annyi kell, hogy a szükséges számú **„elif”** utasítást használnod. Írj egy programot, amely arra kéri a felhasználót, hogy adjon meg egy 45 és 60 közötti számot. Adj meg egy olyan választ, amely megmondja a felhasználónak, hogy a szám **„tartományon belül van”, „túl magas”** vagy **„túl alacsony”**.

→ 4. TEVÉKENYSÉG: RAJZOLJ GEOMETRIAI ALAKZATOKAT PYTHONNAL!

A Pythonban számos olyan modul létezik, amelyeket használhatsz a programjaidban. A modul lényegében egy használati utasítás, amely elmagyarázza, hogyan kell valamit csinálni. Ezeket a modulokat a Python egy könyvtárban tárolja. A tevékenység során megtanulod, hogyan használhatod a Python „Turtle modulját” geometriai alakzatok rajzolására, miközben még több olyan Python programnyelvet tanulsz, ami hasznos lehet a küldetésed során.

Gyakorlat

1. Nyiss egy új Python 3 ablakot, és írd be az alábbi kódot. Mentsd el és futtasd le.

```
File Edit Format Run Options Windows Help
# to begin using Turtle you need to import the Turtle from the Python library
from turtle import Turtle, Screen

screen = Screen() # creates the window wich will display the turtle drawings
turtle = Turtle() # creates a turtle. The variable (ou noun) turtle is made to refer to it.
turtle.shape("turtle") # gives the turtle shape
```

2. Létrehoztad a teknőcöt, amely segít neked szép rajzokat és alakzatokat készíteni! Most adj neki néhány utasítást.

- 2.1. Add hozzá a következő sort a kódodhoz:

```
turtle.forward(100)
```

Mi történt a teknőccel?

- 2.2. Adj hozzá egy másik utasítást:

```
turtle.right(90)
```

Változtasd meg a zárójelben lévő számot. Próbáld meg a „right” (jobb) szó helyett beírni azt, hogy „left” (bal). Mit csinál a teknőc ezekkel az utasításokkal?

Megjegyzés: a turtle.right(90) utasításban a zárójelben lévő érték fokban van megadva.

- 2.3. Írj az előző alá egy másik utasítást, és figyeld meg a teknőc mozgását:

```
turtle.forward(100)
```

Mi történt a teknőccel?

2.4. Mit kell hozzáadnod a kódodhoz ahhoz, hogy négyzetté egészítsd ki a rajzot? Írd az alábbi mezőbe a végleges kódot, ami alapján a teknőc megrajzol egy négyzetet.

3. Sokféle utasítást adhatsz a teknőcödnek. Figyeld meg a következő kódot:

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

turtle.forward(100)
turtle.right(120)
turtle.forward(100)
turtle.right(120)
turtle.forward(100)
```

3.1. Szerinted milyen geometriai alakzatot fog rajzolni a teknőc? Írd le a választ.

3.2. Mentsd el és futtasd le a kódot. Helyes volt a válaszod?

3.3 A fenti kódban néhány kódsort többször is leírtál. Az informatikában ez nem a leghatékonyabb módszer. Ahelyett, hogy több kódsort írnál be, leírhatod őket egyszer, és hozzáadhatsz egy utasítást az ismétlésre. Pythonban kétféleképpen ismételhetsz meg egy utasítást: while és for ciklusokkal.

3.3.1 Nyiss egy új Python 3 ablakot, és írd be az alábbi kódot. A cikluson belüli kód (azaz a behúzott vagy jobbra tolt kód) az az utasításhalmaz, amelyet a program meg fog ismételni. Mentsd el és futtasd le a programot. Mi történt?

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

turtle.forward(100)
turtle.right(120)
turtle.forward(100)
turtle.right(120)
turtle.forward(100)
```

3.3.2 Ebben a példában egy for ciklust használtál. Másold ki az alábbi kódot, futtasd le, és nézd meg, mi történik.

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

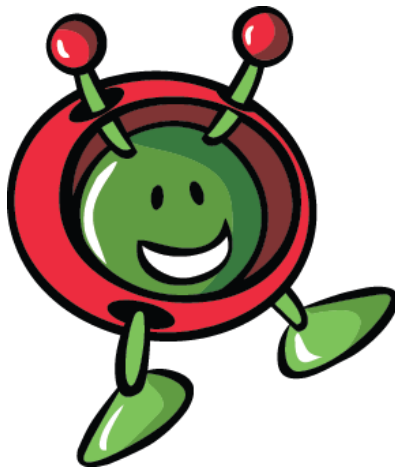
for i in range(3):
    turtle.forward(100)
    turtle.right(120)
```

Megjegyzés: észrevetted, hogy a while/for ciklusok alatti utasítások egymáshoz vannak igazítva? Ezt hívják behúzásnak. A Python behúzással határolja el a kódblokkokat, amikor ciklusokat vagy „if” utasításokat használ. A behúzás minden szintje négy szóközből áll, de a Tab billentyűt megnyomva a program beilleszti a szóközőket. Ne felejtssd használni a behúzást a kód másolásakor és beillesztésekor.

3.3.3 Melyik ciklust választanád a rajz elkészítéséhez? Indokold a választásod, és írd le a két ciklustípus közötti különbségeket.

Kiegészítő feladat

Ciklusok segítségével előre, jobbra és balra is haladhatsz, és ismétléseket is végezhetsz. Most szabadjára engedheted a fantáziád! Tudnál készíteni egy rajzot Paxiról, az Európai Űrügynökség kabalafigurájáról? Milyen geometriai alakzatokat ismersz fel Paxi képén?



HOGYAN TOVÁBB?

Sikerült összeállítanod a Raspberry Pi-t, és most már ismered a Python alapvető programozási struktúráját. A 2. tevékenységből megtudhatod, hogyan kommunikálhatsz az ISS űrhajósaival a Sense HAT LED-mátrix segítségével.

tanítsunk a világűrrel! – ismerd meg az astropi-t! | T05.1
www.esa.int/education

Az Európai Űrügynökség oktatási programja a Raspberry Pi Alapítvánnyal, valamint az Európai Űrügynökség Oktatási Erőforrás Iroda (ESERO) Lengyelországgal és az Egyesült Királysággal együttműködésben

Szerzői jogok 2017 © Európai Űrügynökség