

el?z? fel következ?

Tartalomjegyzék

- 1 Python futtatása
 - ◆ 1.1 Jupyterhub
 - ◆ 1.2 leibniz
 - ◆ 1.3 Saját gépr?!
- 2 Feladatok
 - ◆ 2.1 Köszönés
 - ◆ 2.2 celsiusra
 - ◆ 2.3 prime
 - ◆ 2.4 szobahomerseklet_1
 - ◆ 2.5 szobahomerseklet_2
 - ◆ 2.6 factorial
 - ◆ 2.7 haromszog
 - ◆ 2.8 without_2
 - ◆ 2.9 Tökéletes számok
- 3 Trükkösebb feladatok
 - ◆ 3.1 Fermat-prímteszt
 - ◆ 3.2 Carmichael-számok

Python futtatása

Jupyterhub

- Jelentkezzetek be a jupyter.math.bme.hu-ra a **leibniz**-es felhasználónévvel és jelszóval
- **Python 3**-at fogunk használni!
- Ez a notebook hasonlít ahhoz, mint amikor saját gépr?! ezt futtatod:

```
jupyter notebook
```

leibniz

- A konzol-ba ezt írjuk be:

```
python3
```

- kilépni az így lehet:

```
exit()
```

Saját gépr?!

Installáljuk az Anaconda-t, 3.7-es verzió!

- hogyan Installáljuk az Anacondat Windows-on
- Más disztribúciót is lehet használni, úgymint:
 - ◆ python.org
 - ◆ WinPython

Ha ezt megtettük, akkor több parancs segítségével is interakcióba léphetünk a Python-nal:

- parancssor: `python` vagy `ipython`
- Spyder
- idle
- jupyter notebook

Feladatok

Köszönés

Írjunk egy programot, ami bekért két inputot, majd kiprinteli a következő mondatot:

```
"http://wiki.math.bme.huSzia input1! Én input2 vagyok."http://wiki.math.bme.hu
```

Próbáljuk megírni kétféleképpen is. Összefűzéssel (concatenate) és a tanult behelyettesítéssel (%) is.

celsiusra

Írjunk python függvényt, ami egy Fahrenheitben megkapott hőmérsékletet átvált Celsius fokra. A függvény neve legyen **celsiusra**, és paraméterként egy **fahrenheit** nevű számot kapjon. Úgy lehet kiszámolni ezt az értéket, hogy a Fahrenheit-ben mért hőmérsékletből kivonunk 32-t, majd az így kapott számot megszorozzuk 5/9-el.

- <https://hu.wikipedia.org/wiki/Fahrenheit>
- [példák itt](#)

prime

Írjunk python függvényt, ami megmondja, hogy egy pozitív egész szám prím-e.

A függvény neve legyen **prime**, egy paramétere legyen:

- **x**, a vizsgálandó szám

A függvény **True**-val vagy **False**-al térjen vissza attól függően hogy a szám prím vagy sem.

A biztonság kedvéért érdemes leellenőrizni, hogy az x változó helyes-e egyáltalán. Ha nem egy integer-t vagy nem pozitív integer-t kapunk, akkor térjen vissza a függvény a **None** értékkel.

szobahomerseklet_1

A [Wikipedia](#) szerint a *szobahőmérséklet* 18°C és 25°C között van. Írjunk egy függvényt, ami leellenőrzi ezt.

A függvény

- neve legyen `szobahomerseklet`
- egyetlen változója `fok`, ami a szoba hőmérséklete Celsiusban
- térjen vissza az alábbi string-ek egyikével:
 - ◆ `"http://wiki.math.bme.huTúl hideg!"http://wiki.math.bme.hu`
 - ◆ `"http://wiki.math.bme.huTúl meleg!"http://wiki.math.bme.hu`
 - ◆ `"http://wiki.math.bme.huOK"http://wiki.math.bme.hu`

Figyeljük meg a különbséget a **print** és **return** között!

szobahomerseklet_2

Írjuk meg az előző függvényt úgy is, hogy a bemenetet Fahrenheitben adjuk meg. (Használjuk ehhez a celsiusra függvényünket.)

factorial

Írjunk egy függvényt, ami kiszámolja n faktoriális értékét.

haromszog

Írjunk egy függvényt, melynek bemenete egy n szám, és egy olyan háromszöget printtel ki, melynek n sora van és az n -dik sorban pontosan n -szer szerepel az n szám.

Pl.: $n=4$ esetén:

```
1
22
333
4444
```

Próbáljuk megírni minél többféleképpen.

without_2

Írjunk egy függvényt, ami egy adott n számra visszaadja azt a számot, amelyet úgy kapunk, hogy leosztjuk azzal a legnagyobb kettőhatvánnyal, amivel még osztható a szám.

Például:

```
1 -> 1
2 -> 1
3 -> 3
4 -> 1
6 -> 3
7 -> 7
9 -> 9
```

prime

10 -> 5
100 -> 25

Tökéletes számok

Írjunk programot, mely bekér egy pozitív egész számot és leellenőrizi, hogy tökéletes szám-e.

Trükkösebb feladatok

A következő feladatok azok számára vannak, akik szeretnének mélyebben elmerülni a programozásban. Az itt található feladatok több gondolkodást igényelhetnek, de hasznosak. Későbbi tanulmányaitok során találkozhattok más tárgyak során az itt felmerülő feladatokkal.

Fermat-prímteszt

Implementáljuk a Fermat-prímtesztet. <https://hu.wikipedia.org/wiki/Fermat-pr%C3%ADmteszt>
A már megírt prime függvény és a Fermat-prímteszt segítségével határozzuk meg az összes 100 és 1000 közötti 2-es alapú Fermat-álprímet. https://en.wikipedia.org/wiki/Fermat_pseudoprime

Carmichael-számok

Keressünk egy Carmichael-számot. <https://hu.wikipedia.org/wiki/Carmichael-sz%C3%A1m> Ha túl lassan fut a függvény, akkor elég, ha keresünk egy számot interneten, és leteszteljük az előbb megírt függvények közül a megfelelőt, hogy valóban Carmichael-szám-e.

előző fel következő