

PYTHON BACKEND FEJLESZTÉS

4. forduló



A kategória támogatója: Cambridge Mobile
Telematics (TrueMotion)

RENDELKEZÉSRE ÁLLÓ IDŐ:

40:00

Ismertető a feladathoz

Fontos információk

A forduló után a megoldások publikálásával együtt iránymutatásként elérhetőek lesznek a **helyezéssel kapcsolatos információk**, látni fogod, hogy a kategóriában a játékosok 20%, 40% vagy 60%-a közé tartozol-e épp.

Felhívjuk figyelmedet, hogy a következő, **5. fordulótól az egyes kategóriák csak a kijelölt napokon lesznek megoldhatóak 7-22 óra között**, érdemes letöltened a naptárat a [Kategóriáim](#) menüpontban.

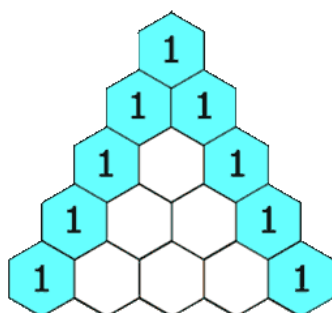
4th round – Generators, Pattern matching

Felhasznált idő: 01:48/40:00

Elért pontszám: 0/13

1. feladat 0/8 pont

Implement a generator function which generates the rows of the Pascal's triangle.



The gen_pascal() generator function should be invocable by the following code snippet:

```
g = gen_pascal(10)
for i in g:
    print(i)
```

So that the code should return the following output:

```
[1]
[1, 1]
[1, 2, 1]
[1, 3, 3, 1]
[1, 4, 6, 4, 1]
[1, 5, 10, 10, 5, 1]
[1, 6, 15, 20, 15, 6, 1]
[1, 7, 21, 35, 35, 21, 7, 1]
[1, 8, 28, 56, 70, 56, 28, 8, 1]
[1, 9, 36, 84, 126, 126, 84, 36, 9, 1]
```

What is the sum of the numbers in the first 30 rows, only counting the odd rows (beginning from the 1st: 1, 3, 5 ... 29)?

Válaszok

A helyes válasz:

357,913,941

357913941

357 913 941

Magyarázat

```
sum([item for sublist in [n for n in gen_pascal(30) if len(n) % 2 == 1] for item in sublist])
```

2. feladat 0/5 pont

Write a pattern matcher for SENSOR_UPLOAD messages where the gps sensor is the first element of the sensor list.

This assignment requires knowledge about pattern matching, introduced in Python3.10 (beta).

Example message:

```
{
    'type': 'SENSOR_UPLOAD',
```

```
'sensors_list': ['gps', 'gyro', 'test_sensor'],  
'sensor_data': [12, 34124, 451251, 515, 5412512],  
'metadata': {}  
}
```

Válasz



```
case {'type': MessageType.SENSOR_UPLOAD.value, 'sensors_list': ['gps', *_], 'sensor_data': sensor_  
    return process_sensor(sensor_data)
```



```
case {'type': MessageType.SENSOR_UPLOAD.value, 'sensors_list': ['gps', _*], 'sensor_data': sensor_  
    return process_sensor(sensor_data)
```



```
case {'type': MessageType.SENSOR_UPLOAD, 'sensors_list': 'gps', 'sensor_data': sensor_data}:  
    return process_sensor(sensor_data)
```



```
case message:  
    t = message['type']  
    if t == 'SENSOR_UPLOAD':  
        sensor_data = message['sensor_data']  
        sensor_list = message['sensor_list']  
        if sensor_list[0] == 'gps':  
            return process_sensor(sensor_data)
```

Magyarázat

Only one answer is syntactically correct.

