



Takács Edit

edit.takacs@gmail.com

GitHub:

https://github.com/gtakacse/CS_class_scripts



Emberi Erőforrások
Minisztériuma

AZ EMBERI
ERŐFORRÁSOK
MINISZTERIUMA ÚJ
NEMZETI KIVÁLÓSÁG
PROGRAMJÁNAK
TÁMOGATÁSÁVAL
KÉSZÜLT



Programozás nyelvészeknek (6. óra)

Methods

For/While ciklusok

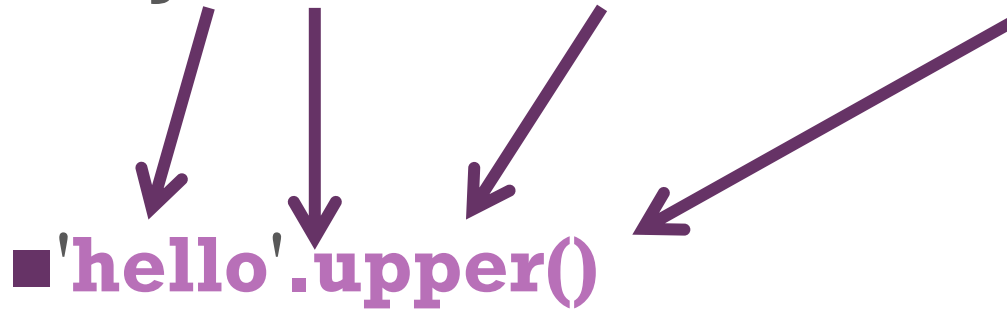


Python method



- Olyan funkciók, amelyek osztályokon (Python objektumokon) belül vannak meghatározva.

■ **object.method_function()**



- methodok listája: **dir(obj)** (e.g. **dir('hello')**)
- **str.upper('hello')** és **'hello'.upper()** ugyanazt jelentik



String methods



- capitalize
- index
- islower
- join
- lower/upper
- replace
- strip
- split
- ...

+ `in` operator

1. `'e' in 'apple'`
True
2. `'x' in 'apple'`
False





A mondat mint Python lista



1. `s = 'Kezdetben volt az ige.'`
2. `sent = s.split()`
3. `print(sent)`

`['Kezdetben', 'volt', 'az', 'ige.']`

- **A mondat szavak listája. A szöveg mondatok listája. Stb.**



Python lista jellemzői, funkciók és method-ok (folyt.)

■ `dir(list)`

- `['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']`

■ az elemek rendezhetők

1. `sorted(list1)`
2. `list1.sort()`

■ bővíthető új elemekkel

1. `list1.append(99)`

■ az elemek törölhetők

1. `del list1[0]`
2. `list1.remove('19')`

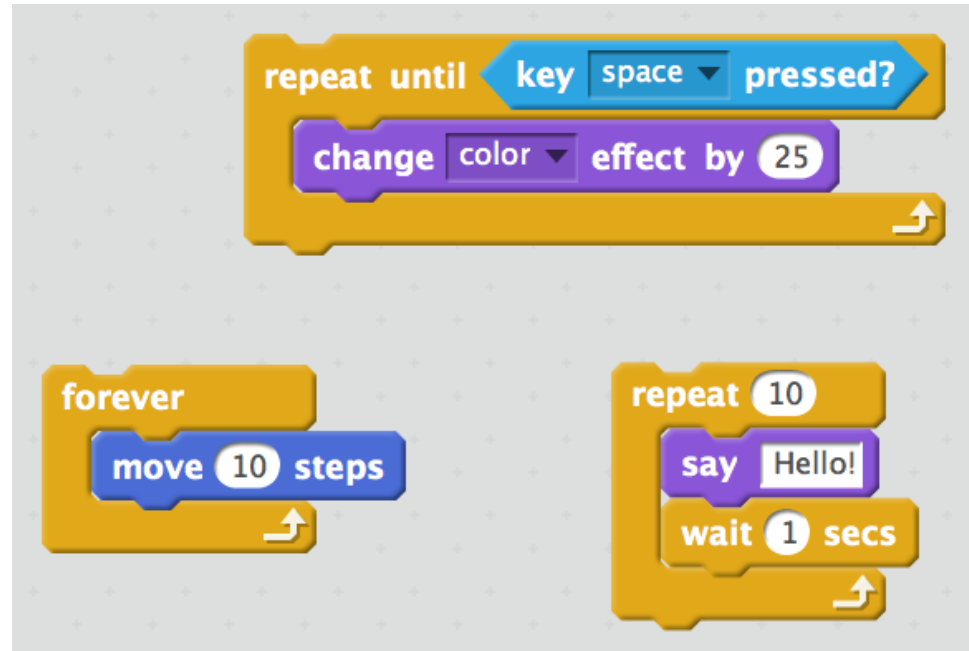
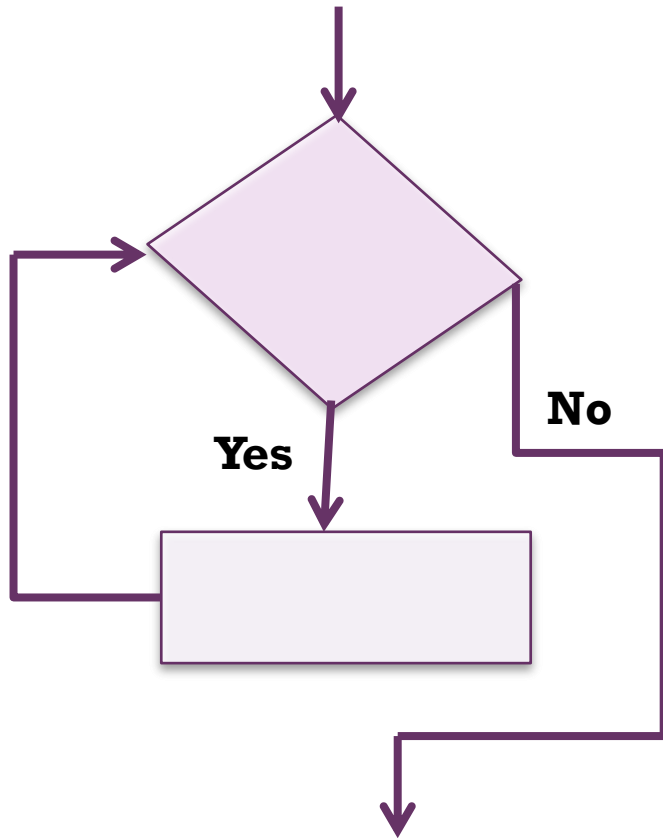
■ listák összevonhatók

1. `list1 + list2`
2. `list1.extend(list2)`

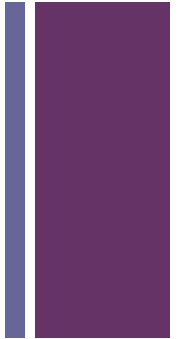
■ iterálható (végig lehet menni az elemeken egyenként (for ciklus később))

■ stb.

+ Ciklusok (Loops) - Ismétléses struktúrák



+ For loop (számlálós ciklus)



1. **for** elem **in** sorozat:
2. csinálj valamit



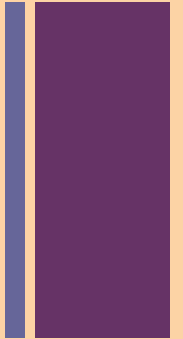
'hajrá' kinyomtatása 5 alkalommal



1. `for i in range(5):`
2. `print(i)`
3. `print('Hajrá!!!')`



Feladat



- Írj egy olyan for ciklust, ami a következő output-ot adja!

10

7

4

1

‘Bye’

‘ByeBye’



While loop (előtesztelő ciklus)



1. **while** a kifejezés igaz:

2. csinálj valamit

1. `count = 5`

2. **while** `count > 0:`

3. `print(count)`

4. `count = count - 1`

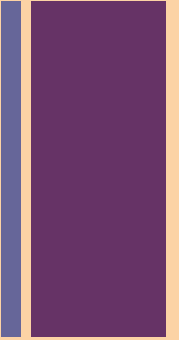
```
count = 0
while count < 500:
    print('I will not throw paper
          airplanes in class')
    count += 1
```

NICE TRY.





Feladat



- Írj egy olyan while ciklust, ami a következő output-ot adja!

10

7

4

1

'Bye'

'ByeBye'

+ Loop control



- break
- continue
- pass



Bennfoglalásos struktúrák



1. `for ind in range(0,15, 3):`
2. `if ind < 10:`
3. `print('Index {} is one digit long. '.format(ind))`
4. `else:`
5. `print('Index {} is at least 2 digits long. '.format(ind))`