

Python programozás az alapvizsgán

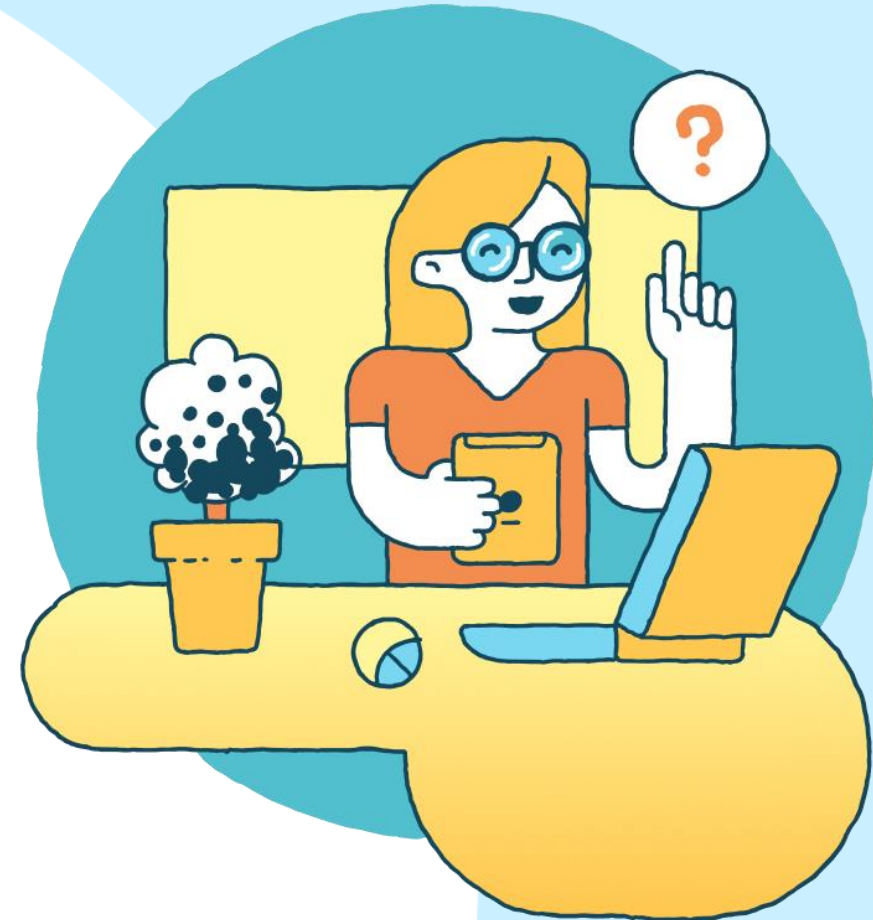
Miért ne féljünk az objektumoktól



Reményi Zoltán, HTTP Alapítvány



Képzési és Kimeneti Követelmények és a Python



6.2 Ágazati alapoktatás szakmai követelményei

Sor-szám	Készségek, képességek	Ismeretek	Elvárt viselkedésmódok, attitűdök	Önállóság és felelősség mértéke
12	Munkája során jelentkező problémák kezelésére vagy folyamatok automatizálására egyszerű alkalmazásokat készít Python programozási nyelv segítségével.	Ismeri a Python nyelv elemeit, azok céljait (vezérlési szerkezetek, adatszerkezetek, változók, aritmetikai és logikai kifejezések, függvények, modulok, csomagok). Ismeri az algoritmus fogalmát, annak szerepét.	Jól átlátható kódszerkezet kialakítására törekszik.	Önállóan készít egyszerű alkalmazásokat.

7.3 Gyakorlati vizsga

A gyakorlati vizsgatevékenység során három feladatból álló feladatsort kell megoldaniuk a tanulóknak.

A tanuló a gyakorlati vizsgatevékenység megkezdésekor mindhárom feladat leírását megkapja. A gyakorlati vizsgatevékenység végrehajtásához rendelkezésre álló idő egybefüggő 180 perc, azon belül az egyes feladatok megoldására fordított idő a tanuló döntése, az egyes feladatok megoldására javasolt időkeret 60-60-60 perc.

A gyakorlati vizsgatevékenység végrehajtásához internetkapcsolat áll a tanulók rendelkezésére. Az internetkapcsolat biztosításának módját és formáját az adott vizsgafeladathoz kiadott útmutató tartalmazza. Ennek megfelelően az internetkapcsolat korlátozódhat meghatározott internetes címekre és/vagy hozzáférési időtartamra, de mindenképpen biztosítani kell, hogy az internetkapcsolatot a tanulók kizárólag általános keresésre használhassák, mással történő kommunikációra vagy a vizsgához célirányosan elkészített anyagok letöltésére ne.

7.3 Gyakorlati vizsga

B) Programozás Pythonban feladat

A feladat során három, egymástól függetlenül is megoldható részfeladatot kell megoldaniuk a tanulóknak Python nyelv segítségével. **A részfeladatok fokozatosan nehezednek**, a legegyszerűbb megoldása pár perc alatt elkészíthető, de a **legnehezebb feladat megoldása sem okozhat különösebb nehézséget egy átlagos képességű, de jól felkészült tanuló számára**. Elvárás lehet teljesen önállóan létrehozott alkalmazás készítése, de **lehet olyan feladat is, amiben egy készen kapott kódot kell a tanulóknak kiegészíteniük**.

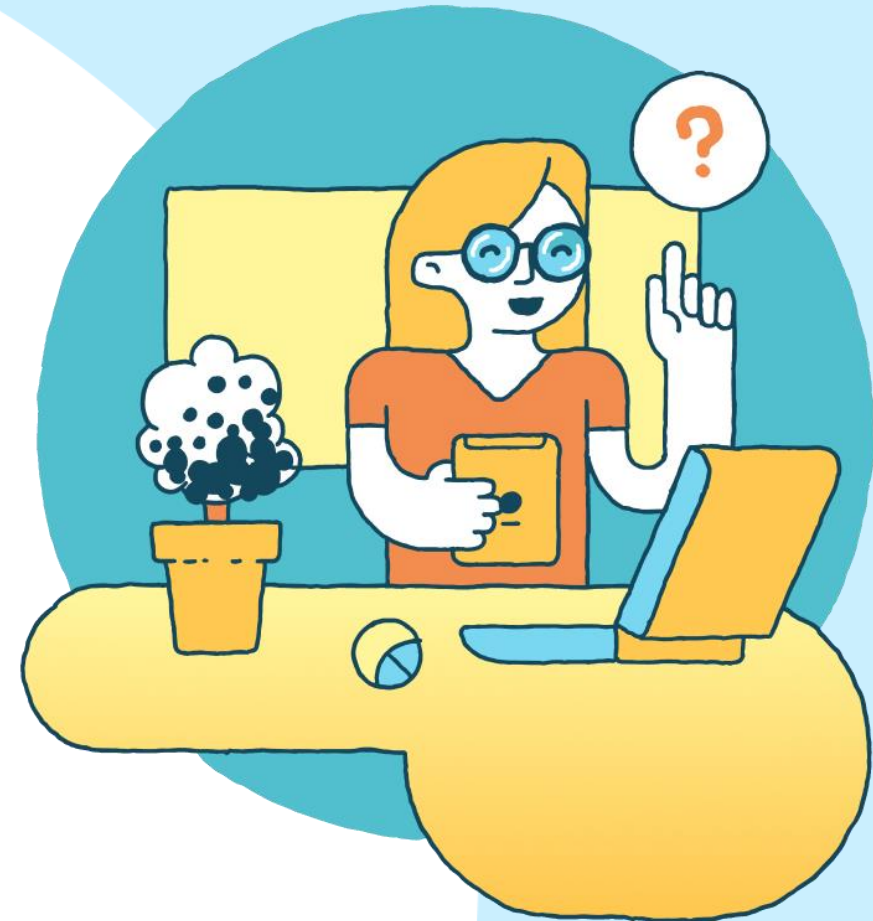
A feladat az alábbi témakörökhöz kapcsolódó gyakorlati készségeket méri:

- önálló alkalmazás készítése, készen kapott alkalmazás kiegészítése, módosítása saját kóddal
- összetett kifejezések készítése aritmetikai, relációs és logikai operátorok segítségével
- saját függvény definiálása (paraméterezés, visszatérési érték meghatározás) és hívása
- modulok felhasználása
- **saját osztály definiálása, saját, vagy készen kapott osztály példányosítása**
- szöveges fájlból adatbeolvasás, a beolvasott adatok tárolása egyszerű vagy összetett adatszerkezetben, adatok kiírása szöveges fájlba
- egyszerűbb problémák megoldására algoritmus készítése és megvalósítása

Programtanterv

(ami már nincs is 😞)

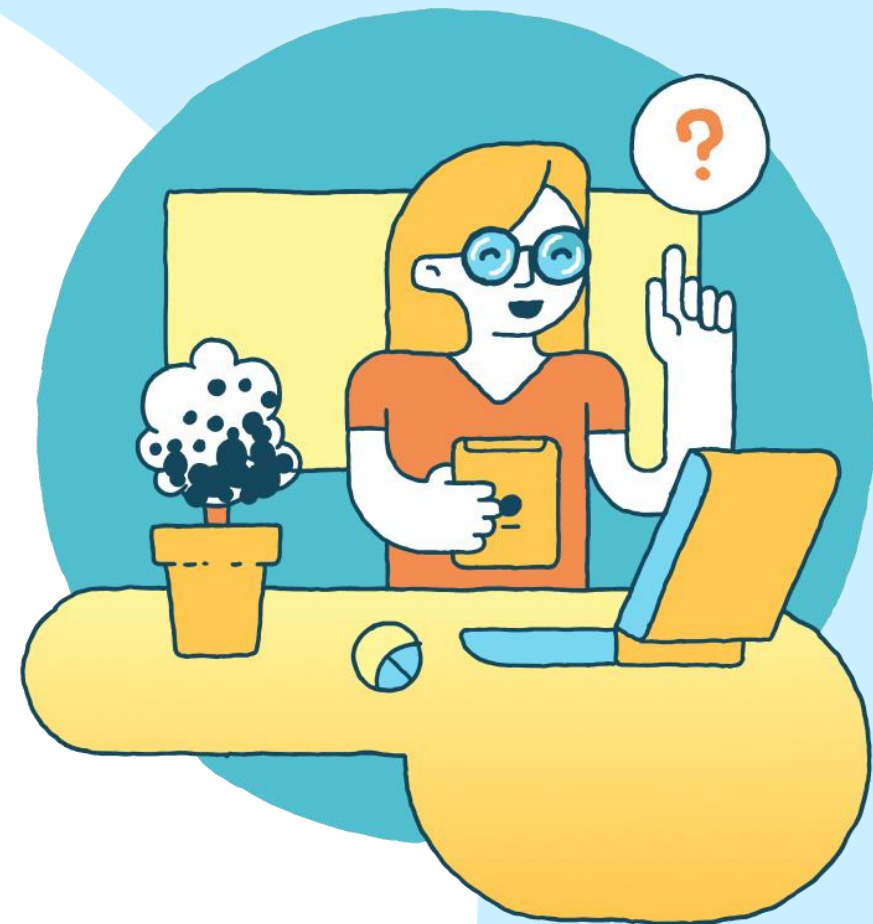
és a Python



3.4.1.6.9 Modulok, objektumok, fájlkezelés Pythonban

- Ismeri az objektumorientált programozás (OOP) koncepcióját.
- Ismeri a procedurális és az objektumorientált megközelítés különbségeit.
- Ismeri az osztály (class) fogalmát.
- Ismeri az osztályhierarchia szerepét.
- Ismeri az objektum fogalmát.
- Képes létrehozni egyszerű saját osztályt és objektumot.

Objektum és a Python



Mi lesz az eredmény?

```
2 Hi! ['Gipsz', 'Jakab', 42]
```

```
<class 'int'>
```

```
<class 'str'>
```

```
<class 'list'>
```

```
<class 'NoneType'>
```

```
<class 'type'>
```

```
a = 2  
b = "Hi!"  
c = ['Gipsz', 'Jakab', 42]
```

```
print(a, b, c)
```

```
print(type(a))
```

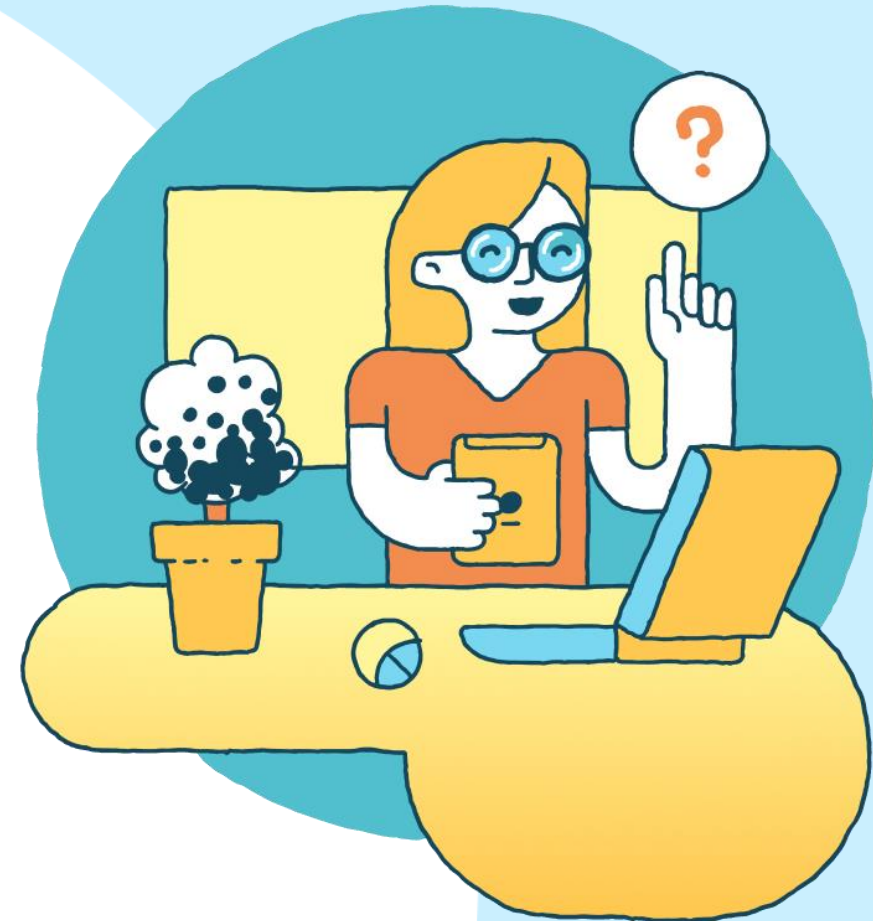
```
print(type(b))
```

```
print(type(c))
```

```
print(type(print()))
```

```
print(type(type))
```

Nézzünk egy-két példát!



Adjuk meg forrásként az alábbi kódot!

```
class derékszögű:
    '''Egy derékszögű háromszöget adhatunk meg a két befogójával.\n
    átfogó() metódus: meghatározza a derékszögű háromszögünk átfogóját'''

    def __init__(self, befogó1: float, befogó2: float):
        self.a = befogó1
        self.b = befogó2

    def átfogó(self) -> float:
        '''Meghatározza a derékszögű háromszög befogóiból az átfogóját'''
        return (self.a*self.a+self.b*self.b)**(1/2)
```

```
háromszög = derékszögű(3, 4)
print(f'A derékszögű háromszög átfogója {háromszög.átfogó(
)} egység, ha a befogók {háromszög.a} és {háromszög.b} egység hosszúak')
```

Feladat

- Egészítsd ki a **derékszögű** osztály definícióját egy új metódussal, ami a derékszögű háromszög területét határozza meg!
(A derékszögű háromszög területe a két befogó szorzatának fele.)

```
def terület(self) -> float:  
    return(self.a*self.b/2)
```

- A program kérje be a felhasználótól a háromszög két befogóját, és a **háromszög** nevű változót azok felhasználásával hozd létre!

```
class derékszögű:  
    '''Egy derékszögű háromszöget adhatunk meg a két befogójával.\n    átfogó() metódus: meghatározza a derékszögű háromszögünk átfogóját'''  
  
    def __init__(self, befogó1: float, befogó2: float):  
        self.a = befogó1  
        self.b = befogó2  
  
    def átfogó(self) -> float:  
        '''Meghatározza a derékszögű háromszög befogóiból az átfogóját'''  
        return (self.a*self.a+self.b*self.b)**(1/2)  
  
háromszög = derékszögű(3, 4)  
print(f'A derékszögű háromszög átfogója {háromszög.átfogó(  
)} egység, ha a befogók {háromszög.a} és {háromszög.b} egység hosszúak')
```

Feladat – Téglalap osztály

Készíts egy **Téglalap** osztályt Python nyelven, amely egy téglalap két oldalát (**szélesség**, **magasság**) tárolja attribútumokban.

Az osztálynak legyenek az alábbi metódusai:

terület(): kiszámítja és visszaadja a téglalap területét.

kerület(): kiszámítja és visszaadja a téglalap kerületét.

Ezután írd meg azt a kódot, amely bekéri a felhasználótól a téglalap oldalainak méreteit, példányosítja a **Téglalap** osztályt, és kiírja a téglalap területét és kerületét.

A megoldást a **derékszögű** osztályban látható módon lásd el a későbbi használatot segítő dokumentációs sztringekkel!

```
class derékszögű:
    """Egy derékszögű háromszöget adhatunk meg a két befogójával.\n
    átfogó() metódus: meghatározza a derékszögű háromszögünk átfogóját"""
    def __init__(self, befogó1: float, befogó2: float):
        self.a = befogó1
        self.b = befogó2
    def átfogó(self) -> float:
        """Meghatározza a derékszögű háromszög befogóiból az átfogóját"""
        return (self.a**2+self.b**2)**(1/2)

háromszög = derékszögű(3, 4)
print(f"A derékszögű háromszög átfogója (háromszög.átfogó(\n
)) egység, ha a befogók (háromszög.a) és (háromszög.b) egység hosszúak")
```

Megoldás - Téglalap osztály

```

class Téglalap:
    '''Egy téglalapot adhatunk meg\n
    a szélesség és a magasság adataival\n
    az objektum metódusai a terület() és a kerület()'''

    def __init__(self, szelesseg: float, magassag: float):
        self.szelesseg = szelesseg
        self.magassag = magassag

    def terület(self) -> float:
        '''Meghatározza a téglalap területét'''
        return self.szelesseg * self.magassag

    def kerület(self) -> float:
        '''Meghatározza a téglalap kerületét'''
        return 2 * (self.szelesseg + self.magassag)

# Felhasználói input és objektum példányosítása
szelesseg = float(input("Kérlek add meg a téglalap szélességét: "))
magassag = float(input("Kérlek add meg a téglalap magasságát: "))
teglalap = Téglalap(szelesseg, magassag)

# Terület és kerület kiírása
print(f"A téglalap területe: {teglalap.terület()} egység")
print(f"A téglalap kerülete: {teglalap.kerület()} egység")

```

KÖSZÖNÖM A FIGYELMET!

zoltan.remenyi@http-foundation.hu

