



#01-----

Feladat: Két szám összege.

Írj egy függvényt "osszead" néven, amely két számot kap és visszatér a két szám összegével.

#assert osszead(12, -8) == 4

#assert osszead(12, 8) == 20



Egy lehetséges megoldás Python nyelven:

python

Copy code

```
def osszead(a, b):  
    return a + b
```

Ezután az assert tesztek sikeresen lefuthatnak, ha az `osszead()` függvényt előtte definiáljuk:

python

Copy code

```
assert osszead(12, -8) == 4  
assert osszead(12, 8) == 20
```

Ha az assert tesztek nem futnak le hibátlanul, akkor valószínűleg hibás az `osszead()` függvény definíciója.

#02-----

Feladat: Melyik a kisebb?

Írj egy függvényt "kisebb" néven, amely két számot kap és visszatér a legkisebbel.

#assert kisebb(12, -8) == -8

#assert kisebb(-8, 12) == -8

Egy lehetséges megoldás Python nyelven:

python

 Copy code

```
def kisebb(a, b):  
    if a < b:  
        return a  
    else:  
        return b
```

Az `if` és `else` kulcsszavak használatával vizsgáljuk meg, hogy melyik szám a kisebb. Ha `a` kisebb, akkor az `a` értékét adja vissza a függvény, egyébként pedig a `b` értékét.

#03-----

Feladat: Melyik a nagyobb?

Írj egy függvényt "nagyobb" néven, amely két számot kap és visszatér a legnagyobbal.

Íme egy lehetséges megoldás Python nyelven:

python

 Copy code

```
def nagyobb(szam1, szam2):  
    if szam1 > szam2:  
        return szam1  
    else:  
        return szam2
```

Ez a függvény két számot kap bemenetként, és megvizsgálja, hogy melyik a nagyobb. Ha az első szám nagyobb, akkor azt adja vissza, különben a második számot adja vissza. Ha a két szám egyenlő, akkor bármelyiket visszaadhatja a függvény.

#04-----

Feladat: Számtani közép

Írj "szamtani_kozep" néven függvényt, amely két számot kap bemenetként és visszatér a számtani középpel.

A számtani közép az a középvérték, amelyet a két szám összege osztva kettővel ad eredményül. Így a megoldás egyszerűen kiszámítható, ahogy az alábbi Python kódban is szerepel:

python  Copy code

```
def szamtani_kozep(szam1, szam2):  
    return (szam1 + szam2) / 2
```

Ez a függvény két számot kap bemenetként, majd a számtani közepüket adja vissza, ami a két szám összege osztva kettővel.

Példa használat:

python  Copy code

```
>>> szamtani_kozep(2, 4)  
3.0  
>>> szamtani_kozep(10, 20)  
15.0
```

Fontos megjegyezni, hogy a függvény jelen megoldása csak az egyszerűség kedvéért egész számokkal dolgozik, de természetesen tetszőleges típusú és tizedespontos számokkal is működik.

#05-----

Feladat: Négyzet kerülete

Írj "negyzet_kerulet" néven függvényt, amely egy négyzet oldalhosszát kapja bemenetként és visszatér a négyzet kerületével.

A négyzet kerülete egyszerűen kiszámítható az oldalhosszak összegének kétszeresével, így a megoldás a következő Python kódban foglalható össze:

python  Copy code

```
def negyzet_kerulet(oldalhossz):  
    return 4 * oldalhossz
```

Ez a függvény egy oldalhosszt kap bemenetként, majd a négyzet kerületét adja vissza, ami az oldalhosszak összegének kétszerese.

Példa használat:

python  Copy code

```
>>> negyzet_kerulet(5)  
20  
>>> negyzet_kerulet(10)  
40
```

Fontos megjegyezni, hogy a függvény csak egyszerűség kedvéért dolgozik egész számokkal, de természetesen tetszőleges típusú és tizedespontos számokkal is működik.

#06-----

Feladat: Négyzet területe

Írj "negyzet_terulet" néven függvényt, amely egy négyzet oldalhosszát kapja bemenetként és visszatér a négyzet területével.


A négyzet területe az oldalhossz négyzetével egyenlő, így a megoldás a következő Python kódban foglalható össze:

python  Copy code

```
def negyzet_terulet(oldalhossz):  
    return oldalhossz ** 2
```

Ez a függvény egy oldalhosszt kap bemenetként, majd a négyzet területét adja vissza, ami az oldalhossz négyzetével egyenlő.

Példa használat:

python  Copy code

```
>>> negyzet_terulet(5)  
25  
>>> negyzet_terulet(10)  
100
```

Fontos megjegyezni, hogy a függvény csak egyszerűség kedvéért dolgozik egész számokkal, de természetesen tetszőleges típusú és tizedespontos számokkal is működik.

#07-----

Feladat: Téglalap kerülete

Írj "teglalap_kerulet" néven függvényt, amely egy téglalap oldalhosszait kapja bemenetként és visszatér a téglalap kerületével.

A téglalap kerülete az oldalhosszak összegének kétszerese, így a megoldás a következő Python kódban foglalható össze:

```
python Copy code  
  
def teglalap_kerulet(a, b):  
    return 2 * (a + b)
```

Ez a függvény két oldalhosszt kap bemenetként, majd a téglalap kerületét adja vissza, ami az oldalhosszak összegének kétszerese.

Példa használat:

```
python Copy code  
  
>>> teglalap_kerulet(5, 8)  
26  
>>> teglalap_kerulet(10, 20)  
60
```

Fontos megjegyezni, hogy a függvény csak egyszerűség kedvéért dolgozik egész számokkal, de természetesen tetszőleges típusú és tizedespontos számokkal is működik.

#08-----

Feladat: Téglalap területe

Írj "teglalap_terulet" néven függvényt, amely egy téglalap oldalhosszait kapja bemenetként és visszatér a téglalap területével.

A téglalap területe az oldalhosszak szorzata, így a megoldás a következő Python kódban foglalható össze:

```
python Copy code  
  
def teglalap_terulet(a, b):  
    return a * b
```

Ez a függvény két oldalhosszt kap bemenetként, majd a téglalap területét adja vissza, ami az oldalhosszak szorzata.

Példa használat:

```
python Copy code  
  
>>> teglalap_terulet(5, 8)  
40  
>>> teglalap_terulet(10, 20)  
200
```

Fontos megjegyezni, hogy a függvény csak egyszerűség kedvéért dolgozik egész számokkal, de természetesen tetszőleges típusú és tizedespontos számokkal is működik.

Feladat: Két szám különbsége

Írj "kulonbseg" néven függvényt, amely két számot kap bemenetként és visszatér a két szám különbségével.

A két szám különbsége egyszerűen a két szám különbsége, így a megoldás a következő Python kódban foglalható össze:

```
python  Copy code
```

```
def kulonbseg(a, b):  
    return a - b
```

Ez a függvény két számot kap bemenetként, majd a két szám különbségét adja vissza.

Példa használat:

```
python  Copy code
```

```
>>> kulonbseg(5, 3)  
2  
>>> kulonbseg(10, 20)  
-10
```

Fontos megjegyezni, hogy a függvény csak egyszerűség kedvéért dolgozik egész számokkal, de természetesen tetszőleges típusú és tizedespontos számokkal is működik.

#10-----

Feladat: Maradékos osztás:

Írj egy "maradek" nevű függvényt, amely két számot kap bemenetként és visszatér a két szám osztásának maradékával.

A maradékos osztás a Python-ban az "%" operátorral végezhető el. A megoldás a következő Python kódban foglalható össze:

```
python  Copy code
```

```
def maradek(a, b):  
    return a % b
```

Ez a függvény két számot kap bemenetként, majd a két szám maradékos osztásának maradékát adja vissza.

Példa használat:

```
python  Copy code
```

```
>>> maradek(10, 3)  
1  
>>> maradek(15, 6)  
3
```

Fontos megjegyezni, hogy ha a második szám, azaz a "b" értéke 0, akkor a függvény ZeroDivisionError hibát fog dobni.

#11-----

Feladat: Páros szám:

#Írj egy "paros" nevű függvényt, amely egy számot kap bemenetként, majd True-val tér vissza, ha a szám páros és False-al ha a szám páratlan.

A megoldás egyszerű: ha egy szám osztva 2-vel maradék nélkül osztható, akkor az páros, különben páratlan. A következő Python kódban foglaljuk össze a megoldást:

python  Copy code

```
def paros(szam):  
    if szam % 2 == 0:  
        return True  
    else:  
        return False
```

Ez a függvény egy számot kap bemenetként, majd True értékkel tér vissza, ha a szám páros, különben False értékkel tér vissza, ha a szám páratlan.

Példa használat:

python  Copy code

```
>>> paros(2)  
True  
>>> paros(5)  
False
```


#12-----

Feladat: Kettővel osztható:

Írj egy "kettovel_oszthato" nevű függvényt, amely egy számot kap bemenetként és True-val tér vissza, ha a szám kettővel osztható és False-al ha nem.

Ezt a feladatot már korábban megoldottuk a #11-es feladatban, az ott írt függvényt használhatjuk:

python

 Copy code

```
def paros(szam):  
    if szam % 2 == 0:  
        return True  
    else:  
        return False
```

Ez a függvény akkor ad vissza True értéket, ha a szám maradék nélkül osztható kettővel, tehát kettővel osztható. Ha a szám nem osztható kettővel, akkor False értéket ad vissza.

#13-----

Feladat: Hárommal osztható:

Írj egy "harommal_oszthato" nevű függvényt, amely egy számot kap bemenetként és True-val tér vissza, ha a szám hárommal osztható és False-al ha nem.

Az alábbi függvénnnyel ellenőrizhetjük, hogy egy szám osztható-e hárommal:

python

 Copy code

```
def harommal_oszthato(szam):  
    if szam % 3 == 0:  
        return True  
    else:  
        return False
```

Ez a függvény akkor ad vissza True értéket, ha a szám maradék nélkül osztható hárommal, tehát hárommal osztható. Ha a szám nem osztható hárommal, akkor False értéket ad vissza.


#14-----

Feladat: Héttel osztható:

Írj egy "hettel_oszthato" nevű függvényt Python programozási nyelvben, amely egy számot kap bemenetként és True-val tér vissza, ha a szám héttel osztható és False-al ha nem.

Az alábbiakban egy lehetséges megoldás található:

python

 Copy code

```
def hetel_oszthato(szam):  
    return szam % 7 == 0
```

A függvény egy egyszerű moduló műveletet használ a szám és 7 osztásának maradékának kiszámítására. Ha a maradék 0, akkor a szám osztható 7-tel, és a függvény True értékkel tér vissza. Ellenkező esetben a függvény False értékkel tér vissza.


#15-----

Feladat: Kocka térfogat:

Írj egy "kocka_terfogat" nevű függvényt, amely bemenetként megkapja a kocka oldal hosszúságát és a kocka térfogatával tér vissza.

Az alábbiakban található a megoldás Python nyelven:

python

 Copy code

```
def kocka_terfogat(a):  
    return a ** 3
```

A függvény neve `kocka_terfogat`, és egyetlen bemeneti értéket vár, ami az `a` oldalhossz. A függvény a kocka térfogatával tér vissza, amelyet az `a` oldalhossz köbénak számításával lehet meghatározni, ami a `a ** 3` kifejezéssel adható meg.

#16-----

Feladat: Téglatest térfogat:

Írj egy "teglatest_terfogat" nevű függvényt, amely bemenetként megkapja a téglatest oldalainak hosszúságát és a téglatest térfogatával tér vissza.

Az alábbiakban megtalálható a kért `teglatest_terfogat()` függvény Python kódja, amely bemenetként megkapja a téglatest oldalainak hosszúságát és a téglatest térfogatával tér vissza:

python

 Copy code

```
def teglatest_terfogat(a, b, c):  
    return a * b * c
```

A függvény egyszerűen kiszámolja a téglatest térfogatát a három oldalhossz szorzatának eredményeként. A bemeneti paramétereknek (a, b, c) a függvényhívás során meg kell adniuk a téglatest oldalainak hosszúságát.


#17-----

Feladat: Derékszögű háromszög területe:

Írj egy "derekszogu_haromszog_terulet" nevű függvényt , amely bemenetként megkapja a befogók hosszát és a háromszög területével tér vissza.

Az alábbiakban található a megoldás Python nyelven:

python

 Copy code

```
def derekszogu_haromszog_terulet(a, b):  
    terület = (a * b) / 2  
    return terület
```

A függvény két oldalt kap bemenetként, majd kiszámolja a derékszögű háromszög területét a befogók hosszának felhasználásával. Végül visszatér a terület értékével.

#18-----

Feladat: Derékszögű háromszög átfogója:

Írj egy "derekszogu_haromszog_atfogo" nevű függvényt , amely bemenetként megkapja a befogók hosszát és az átló hosszával tér vissza.

Egy derékszögű háromszög átfogója a háromszögben található azon pontokat köti össze, amelyek a derékszögű csúcsban indulnak és az átfogóval szemközti oldal talpункon fekszenek. Az átfogó hossza a Pitagorasz-tétel alapján számolható ki: az átfogó négyzetének értéke egyenlő a másik két oldal négyzetösszegével.

Az alábbi függvény számítja ki a derékszögű háromszög átfogóját:

python

 Copy code

```
def derekszogu_haromszog_atfogo(a, b):  
    c = (a ** 2 + b ** 2) ** 0.5 # Pitagorasz-tétel  
    return c
```

A függvény a `a` és `b` befogók hosszát kapja bemenetként, majd kiszámítja az átfogó hosszát a Pitagorasz-tétel alapján és visszatér az eredménnyel.


#19-----

Feladat: Négyzet átlója:

Írj egy "negyzet_atloja" nevű függvényt , amely bemenetként megkapja a négyzet oldalának hosszát és az átló hosszával tér vissza.

Az alábbiakban egy lehetséges megvalósítás látható Python nyelven:

python

 Copy code

```
def negyzet_atloja(a):  
    atlo = a * (2 ** 0.5)  
    return atlo
```

A függvény egy oldalhosszt kap bemenetként, majd kiszámolja az átló hosszát az ``a * (2 ** 0.5)`` képlettel, és visszatér az eredménnyel.

#20-----

Feladat: Téglalap átlója:

Írj egy "teglalap_atloja" nevű függvényt, amely bemenetként megkapja az oldalak hosszát és az átló hosszával tér vissza.

Természetesen, íme a kód a megoldáshoz:

python

 Copy code

```
def teglalap_atloja(a, b):  
    return ((a ** 2) + (b ** 2)) ** 0.5
```

A függvény a téglalap oldalainak hosszát várja bemenetként, majd az átló hosszával tér vissza.

A függvény használja a Pitagorasz-tételt, amely szerint a derékszögű háromszög átfogójának hossza megegyezik a befogók négyzetösszegének négyzetgyökével.


#21-----

Feladat: Abszolútérték:

Írj egy "abszolut" nevű függvényt , amely a bemenő paraméterként kapott szám abszolút értékével tér vissza.

Egy lehetséges megoldás a következőképpen nézne ki Python nyelven:

python

 Copy code

```
def abszolut(szam):  
    if szam < 0:  
        return -szam  
    else:  
        return szam
```

Ez a függvény az alábbi módon működik:

- Bemenő paraméterként egy számot vár.
- Megvizsgálja, hogy a szám negatív-e.
- Ha igen, akkor a szám abszolút értékét azzal a számmal tér vissza, amelynek előjele pozitív és amelynek értéke megegyezik a bemenő számmal, csak előjele eltérő.
- Ha a szám nem negatív, akkor a bemenő számot téríti vissza, mivel annak abszolút értéke megegyezik az eredeti szám értékével.