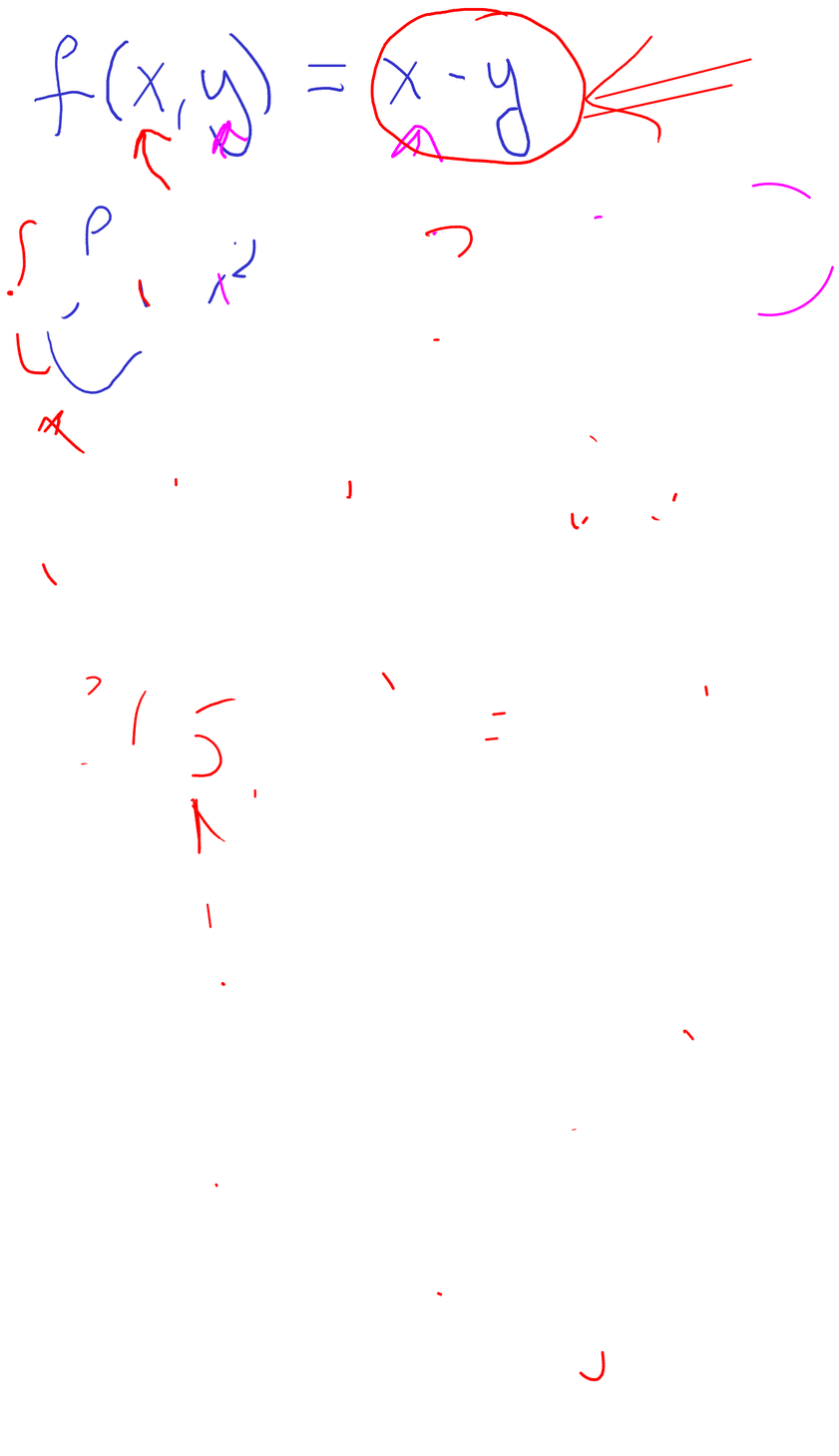
Rascunhos



Resumo

Toda função em Python retora um valor.
Esse valor pode ser None.
None é o único valor do tipo NoneType.

Qualquer função pode chamar qualquer função.

Cada função tem suas próprias variáveis.
Costumamos dizer que essas são suas variáveis locais.

Parâmetros são variáveis locais que são inicializadas com o valor dos argumentos.

Matemática:

parâmetros

f(x, y) = x - y

Python:

```
def f(x, y):
    return x - y
```

Exercício

(a) Escreva uma função que recebe n e retorna n!

```
def fatorial(n):
    '''(int) -> int
    Recebe um inteiro n e retorna
    n!
    '''
    nfat = 1
    # calcule n!
    i = 2
    while i <= n:
        nfat *= i # nfat = nfat*i
        i += 1
    return nfat
```

1 x 2 x 3 x 4 x ... x n

(b) Escreva uma função que recebe inteiros m e n e retorna

$$\binom{m}{n} = \frac{m!}{n!\,(m-n)!}$$

```
def binomial(m, n):
    '''(int, int) -> int
    Retorna m, n-a-n
    '''
    mfat = fatorial(m)
    nfat = fatorial(n)
    mnfat = fatorial(m-n)
    bin = mfat//(nfat*mnfat)
    return bin
```

(c) Escreva uma função main() que lê um número n > 0 e imprime

$$\binom{n}{0}\binom{n}{1}\binom{n}{2}\cdots\binom{n}{n}$$

Para n = 3 imprime

```
binomial(3, 0) = 1
binomial(3, 1) = 3
binomial(3, 2) = 3
binomial(3, 3) = 1
```

```
def main():
    n = int(input("Digite n:"))
    i = 0
    while i <= n:
        print(f"{binomial(n, i)}")
        i += 1 # i = i + 1
```

main()

n	i
3	0

binomial

m	n	mfat
3	0	6

fatorial

n	nfat	i
3	1	2
	2	3
	6	4

retorna 6

Solução

```
def main():
    n = int(input("Digite n: "))
    i = 0
    while i <= n:
        print(binomial(n, i))
        i += 1

#----------------------------------------
def binomial(m, n):
    '''(int) -> int
      Retorna o valor de m n-a-n
    '''
    mfat = fatorial(m)
    nfat  = fatorial(n)
    mnfat = fatorial(m-n)
    return mfat // (nfat*mnfat)
```

```
#----------------------------------------
def fatorial(n):
    '''(int) -> int
      Recebe um inteiro n e retorna  n!
    '''
    nfat = 1
    i = 2
    while i <= n:
        nfat = nfat * i
        i = i + 1
    return nfat


# chamada da função main()
if __name__ == "__main__":
    main()
```