

Introdução ao Processamento de Dados

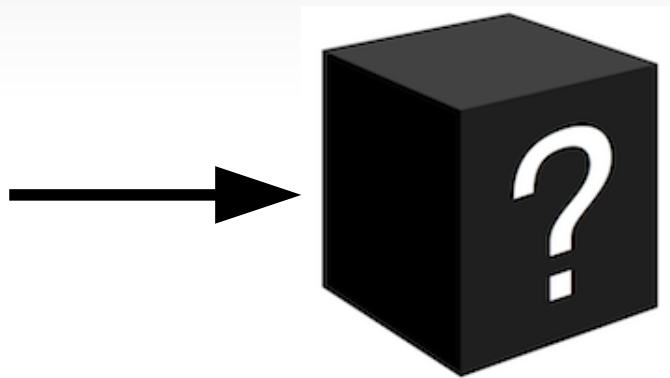
Funções: Abstrações de Código

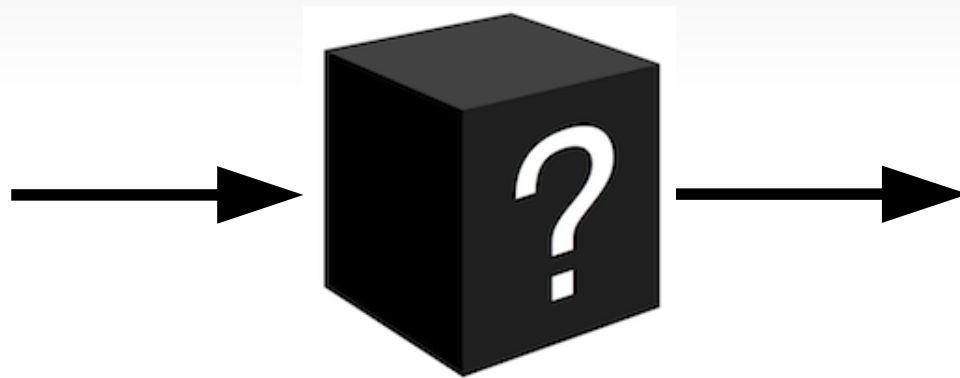
Francisco Sant'Anna

`francisco@ime.uerj.br`

`http://github.com/fsantanna/IPD`

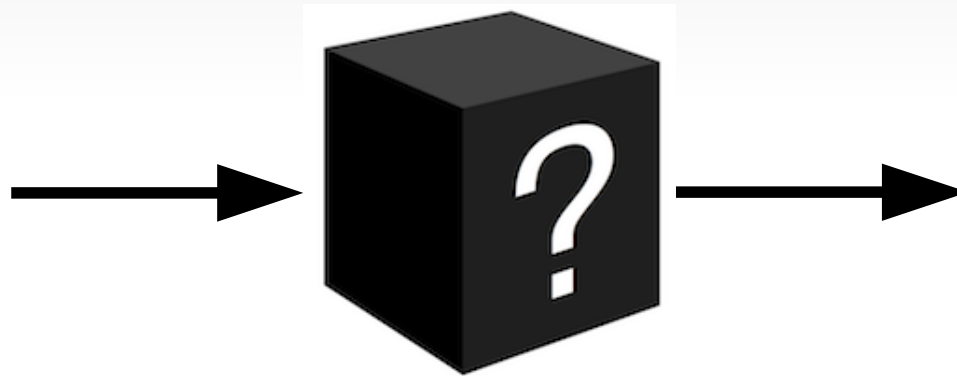






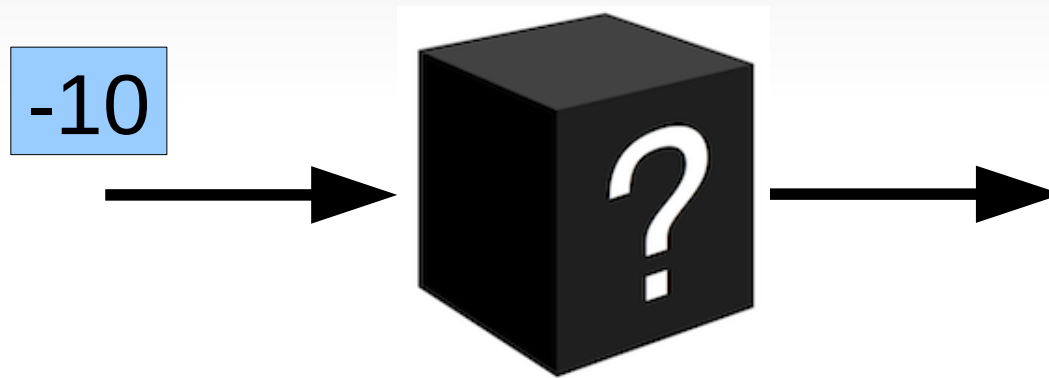
função **abs**:

- recebe um inteiro
- retorna o seu valor absoluto



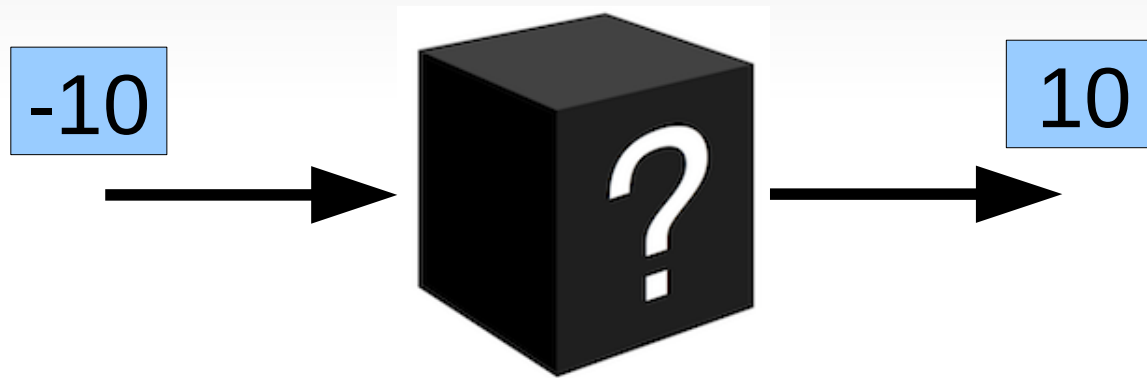
função **abs**:

- recebe um inteiro
- retorna o seu valor absoluto



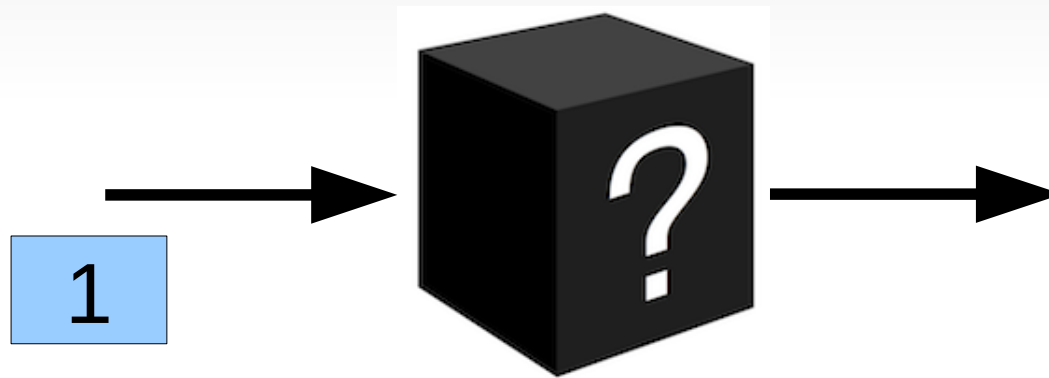
função **abs**:

- recebe um inteiro
- retorna o seu valor absoluto



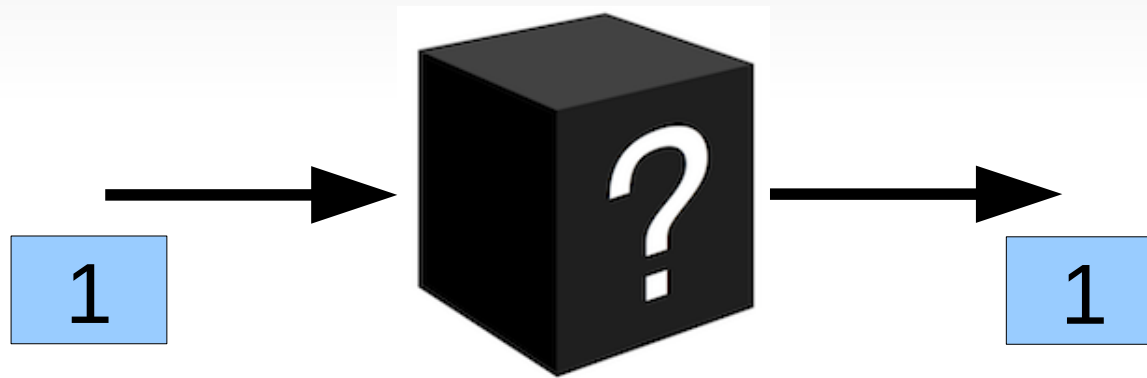
função **abs**:

- recebe um inteiro
- retorna o seu valor absoluto

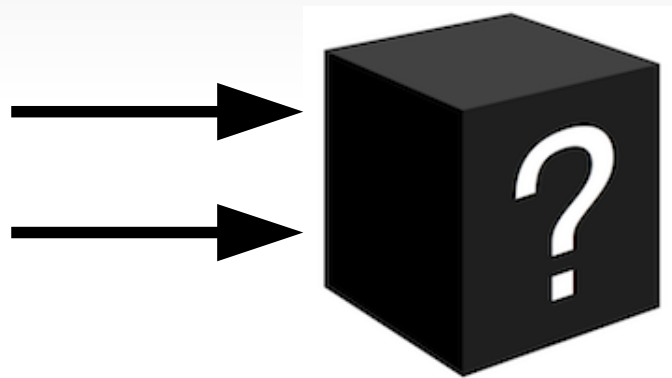


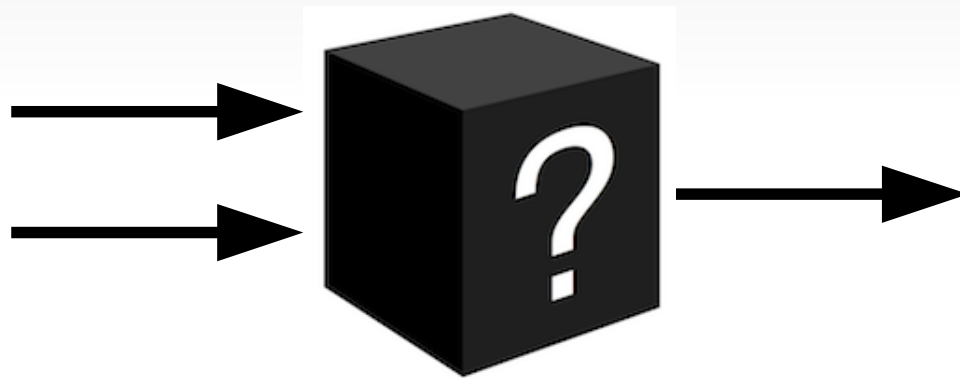
função **abs**:

- recebe um inteiro
- retorna o seu valor absoluto



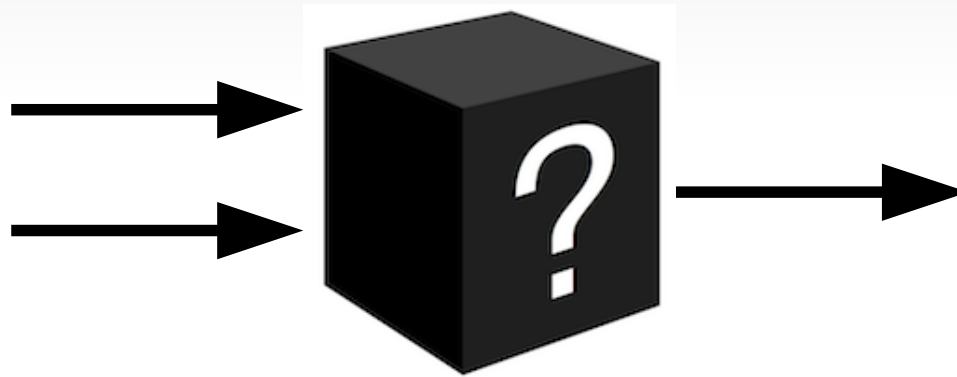






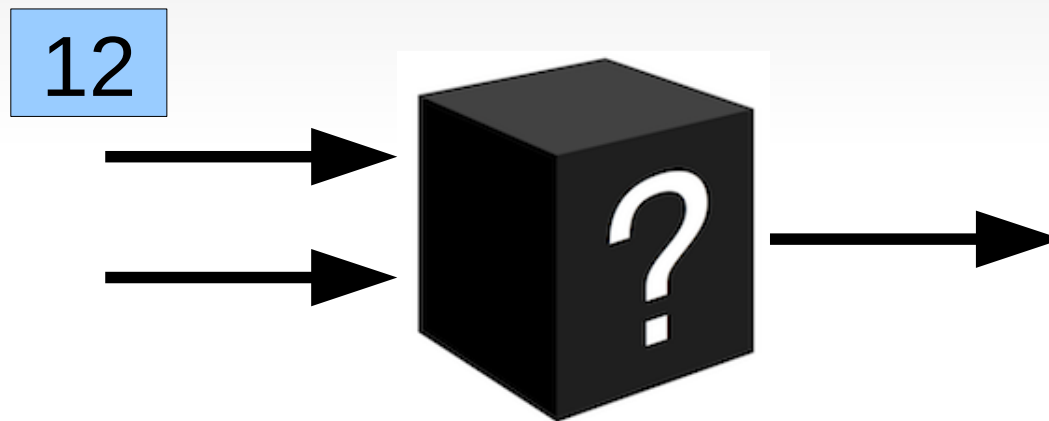
função **mdc**:

- recebe dois inteiros
- retorna o *mdc* entre eles



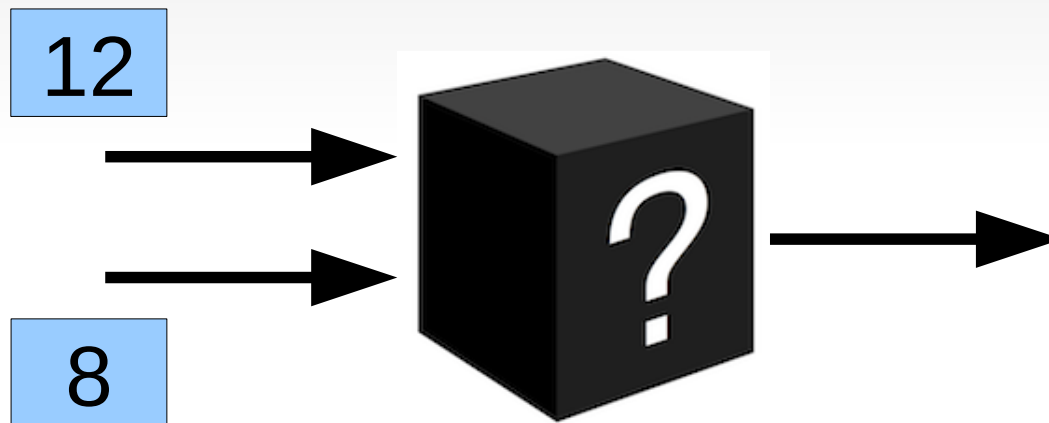
função **mdc**:

- recebe dois inteiros
- retorna o *mdc* entre eles



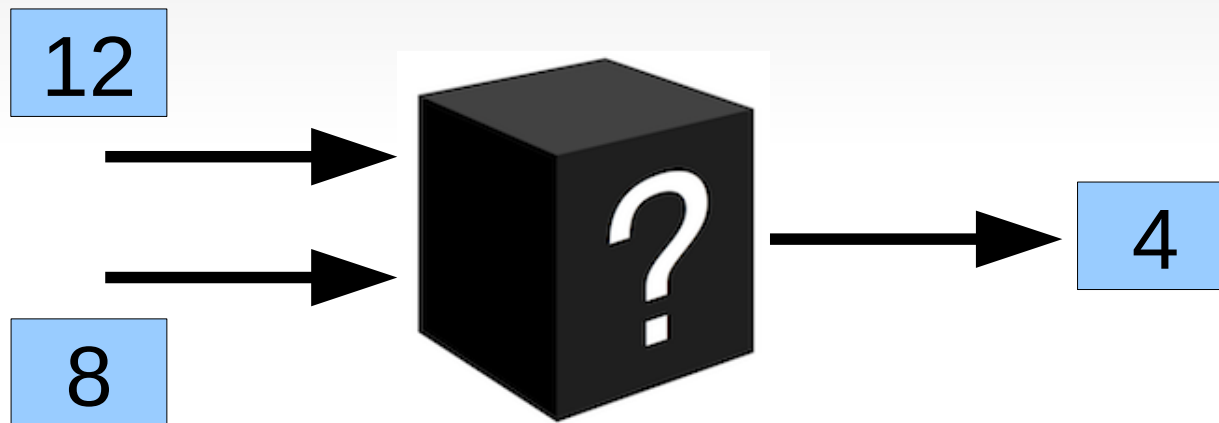
função **mdc**:

- recebe dois inteiros
- retorna o *mdc* entre eles



função **mdc**:

- recebe dois inteiros
- retorna o *mdc* entre eles



Funções

|

Funções

- São abstrações de código

Funções

- São abstrações de código
- Uma função representa uma computação complexa através de um nome

Funções

- São abstrações de código
- Uma função representa uma computação complexa através de um nome
- Vantagens
 - Tornam o código mais legível e organizado
 - Permitem o reuso de código
 - Tornam a localização e correção de erros mais fácil

Funções

- Como usá-las?

Funções

- Como usá-las?

```
v = abs(-10)
```


Funções

- Como usá-las?

```
v = abs(-10)
```

```
v1 = input()  
v2 = abs(v1)  
print "o abs de", v1, "eh", v2
```

Funções

- Como usá-las?

nome

```
v = abs(-10)
```

```
v1 = input()  
v2 = abs(v1)  
print "o abs de", v1, "eh", v2
```

Funções

- Como usá-las?

nome

(

)

```
v = abs(-10)
```

```
v1 = input()  
v2 = abs(v1)  
print "o abs de", v1, "eh", v2
```

Funções

- Como usá-las?

nome (**arg1**)

```
v = abs(-10)
```

```
v1 = input()  
v2 = abs(v1)  
print "o abs de", v1, "eh", v2
```

Funções

- Como usá-las?

nome (**arg1** ,)

```
v = abs(-10)
```

```
v1 = input()  
v2 = abs(v1)  
print "o abs de", v1, "eh", v2
```

Funções

- Como usá-las?

nome (arg1 , arg2)

```
v = abs(-10)
```

```
v1 = input()  
v2 = abs(v1)  
print "o abs de", v1, "eh", v2
```

Funções

- Como usá-las?

nome (**arg1** , **arg2** ,)

```
v = abs(-10)
```

```
v1 = input()  
v2 = abs(v1)  
print "o abs de", v1, "eh", v2
```

Funções

- Como usá-las?

nome (arg1 , arg2 , ...)

```
v = abs(-10)
```

```
v1 = input()  
v2 = abs(v1)  
print "o abs de", v1, "eh", v2
```


Funções

- Como usá-las?

= nome (arg1 , arg2 , ...)

```
v = abs(-10)
```

```
v1 = input()
v2 = abs(v1)
print "o abs de", v1, "eh", v2
```

Funções

- Como usá-las?

`v = nome (arg1 , arg2 , ...)`

```
v = abs(-10)
```

```
v1 = input()
v2 = abs(v1)
print "o abs de", v1, "eh", v2
```

Funções

- Como usá-las?

v = nome (arg1 , arg2 , ...)

```
v = abs(-10)
```

```
v1 = input()  
v2 = abs(v1)  
print "o abs de", v1, "eh", v2
```

```
v = mdc(12,8)
```

Funções

- Como usá-las?

`v = nome (arg1 , arg2 , ...)`

```
v = abs(-10)
```

```
v1 = input()
v2 = abs(v1)
print "o abs de", v1, "eh", v2
```

```
v = mdc(12,8)
```

```
n1 = input()
n2 = input()
res = mdc(n1,n2)
print "mdc eh", res
```

Funções

- Como definí-las?

Funções

- Como defini-las?

```
def abs (v):  
    if v >= 0:  
        return v  
    else  
        return -v
```

Funções

- Como defini-las?

def

```
def abs (v):  
    if v >= 0:  
        return v  
    else  
        return -v
```

Funções

- Como defini-las?

```
def nome
```

```
def abs (v):  
    if v >= 0:  
        return v  
    else  
        return -v
```


Funções

- Como defini-las?

```
def nome ( )
```

```
def abs (v):  
    if v >= 0:  
        return v  
    else  
        return -v
```

Funções

- Como defini-las?

`def nome (` `) :`

```
def abs (v):  
    if v >= 0:  
        return v  
    else  
        return -v
```

Funções

- Como defini-las?

```
def nome ( par1 ) :
```

```
def abs (v):  
    if v >= 0:  
        return v  
    else  
        return -v
```

Funções

- Como defini-las?

```
def nome ( par1 ,           ) :
```

```
def abs (v):  
    if v >= 0:  
        return v  
    else  
        return -v
```

Funções

- Como defini-las?

```
def nome ( par1 , par2 ) :
```

```
def abs (v):  
    if v >= 0:  
        return v  
    else  
        return -v
```

Funções

- Como defini-las?

```
def nome ( par1 , par2 , ) :
```

```
def abs (v):  
    if v >= 0:  
        return v  
    else  
        return -v
```

Funções

- Como defini-las?

```
def nome ( par1 , par2 , ... ) :
```

```
def abs (v):  
    if v >= 0:  
        return v  
    else  
        return -v
```

Funções

- Como defini-las?

```
def nome ( par1 , par2 , ... ) :  
    # codigo python
```

```
def abs (v):  
    if v >= 0:  
        return v  
    else  
        return -v
```


Funções

- Como defini-las?

```
def nome ( par1 , par2 , ... ) :
```

```
# codigo python
```

```
# pode usar par1, par2, ...
```

```
def abs (v):  
    if v >= 0:  
        return v  
    else  
        return -v
```

Funções

- Como defini-las?

```
def nome ( par1 , par2 , ... ) :
```

```
    # codigo python
```

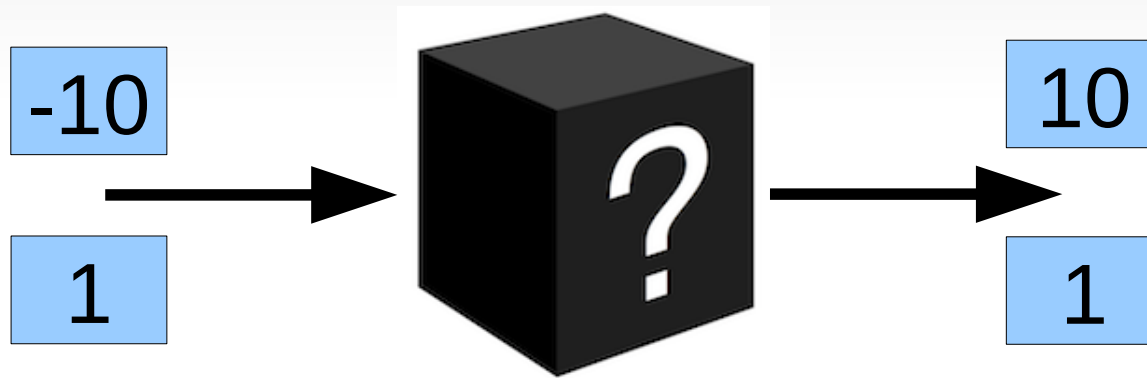
```
    # pode usar par1, par2, ...
```

```
    return <exp>
```

```
def abs (v):  
    if v >= 0:  
        return v  
    else  
        return -v
```

função **abs**:

- recebe um inteiro
- retorna o seu valor absoluto



-10

função **abs**:

- recebe um inteiro
- retorna o seu valor absoluto

-10

função **abs**:

- recebe um inteiro
- retorna o seu valor absoluto

função **abs**:

- **lê** um inteiro
- **escreve** o seu valor absoluto

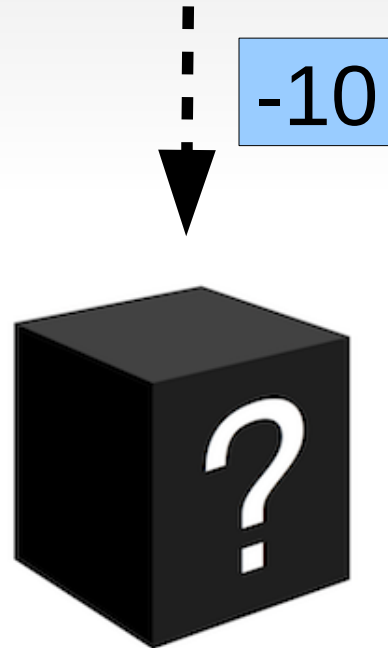
-10

função **abs**:

- recebe um inteiro
- retorna o seu valor absoluto

função **abs**:

- **lê** um inteiro
- **escreve** o seu valor absoluto



função **abs**:

- recebe um inteiro
- retorna o seu valor absoluto

função **abs**:

- **lê** um inteiro
- **escreve** o seu valor absoluto



função **abs**:

- recebe um inteiro
- retorna o seu valor absoluto

função **abs**:

- **lê** um inteiro
- **escreve** o seu valor absoluto



função **abs**:

- recebe um inteiro
- retorna o seu valor absoluto

função **abs**:

- **lê** um inteiro
- **escreve** o seu valor absoluto

Efeitos



Exercício-if-03

- Para doar sangue é necessário ter entre 18 e 67 anos.
- Faça um programa em Python que pergunte a idade de uma pessoa e diga se ela pode doar sangue ou não.

Exercício-if-while

- Para doar sangue é necessário ter entre 18 e 67 anos.
- Faça um programa em Python que pergunte a idade de 100 pessoas e, para cada uma, diga se ela pode doar sangue ou não.

Exercício-funs

- **Faça um programa em Python que pergunte a idade de 100 pessoas e, para cada uma, diga se ela pode doar sangue e se ela pode dirigir.**

Efeitos vs Parâmetros

- **Parâmetros são mais abstratos e gerais**
- **Efeitos são mais concretos e específicos**
- **Evitar uso de efeitos dentro de funções**
- **Usar input/print somente no corpo principal**