

Programação I

Jordana S. Salamon

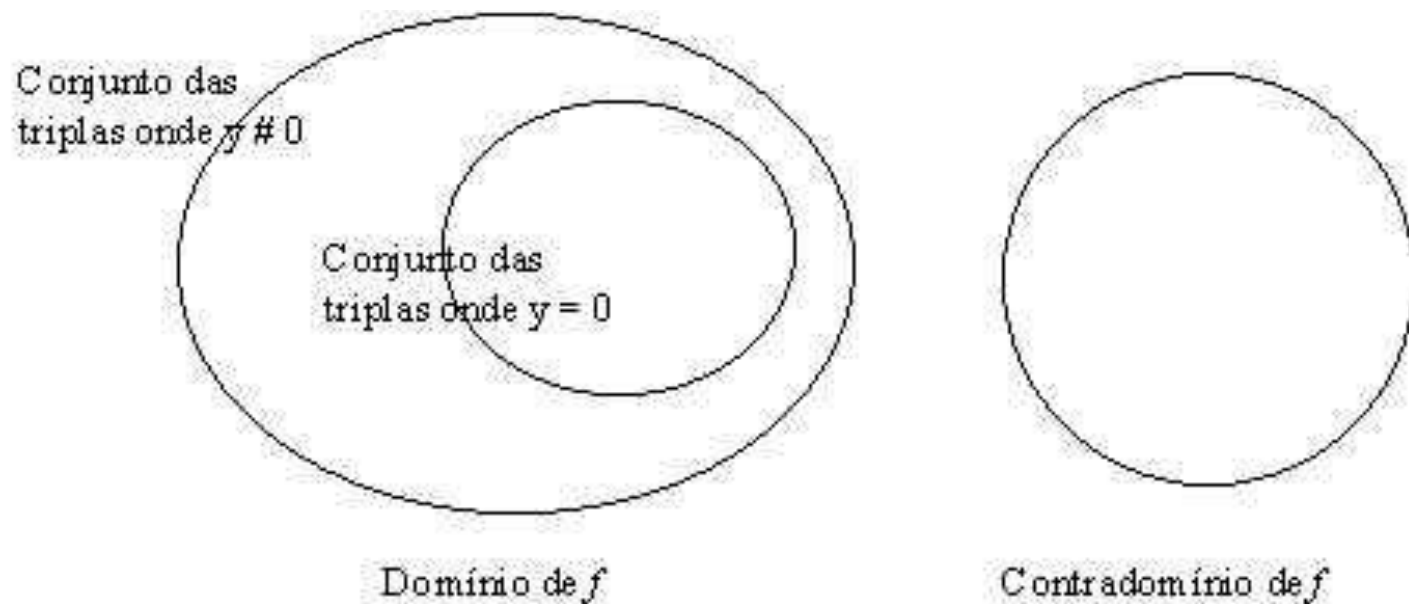
jssalamon@inf.ufes.br

jordanasalamon@gmail.com

DEPARTAMENTO DE INFORMÁTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Validação de Dados

- ▶ Algumas funções não têm definição para uma certa região do domínio.
- ▶ Por exemplo: $f(x, y, z) = x/y + z$, onde $y \neq 0$



Validação de Dados

- ▶ Se codificarmos trivialmente esta função em Python, teremos:

```
>>> def f(x, y, z): return x/y + z
```

```
>>> f(2,0,4)
```

```
Traceback (most recent call last):
```

```
  File "<pyshell#350>", line 1, in <module>
```

```
    f(2,0,4)
```

```
  File "<pyshell#349>", line 1, in f
```

```
    def f(x, y, z): return x/y + z
```

```
ZeroDivisionError: integer division or modulo by zero
```

Validação de Dados

- ▶ Desejamos reconstruir a função original incluindo uma **condição de proteção** para o(s) caso(s) para o(s) qual(is) a função não está definida:

```
>>> def f(x, y, z):  
    if y != 0: return x/y + z  
>>> f(2,0,4)  
>>>
```

Validação de Dados

- ▶ Podemos retornar uma mensagem para explicar o motivo pelo qual a função não pôde ser aplicada:

```
>>> def f(x, y, z):  
    if y == 0:  
        return "Divisao por zero nao permitida"  
    else: return x/y + z  
>>> f(2,0,4)  
'Divisao por zero nao permitida'
```

Validação de Dados

- ▶ Exibir uma mensagem parece ser uma solução adequada para alertar ao usuário o problema ocorrido.
- ▶ Contudo, se o objetivo da sua função é ser reutilizada em outra função, uma mensagem do tipo *string* não será devidamente entendida

```
>>> from math import sqrt
>>> def f(x, y, z):
    if y == 0:
        return "Divisao por zero
            nao permitida"
    else: return x/y + z
>>> def g(x,y,z): return sqrt(f(x,y,z))
>>> g(2, 0, 4)
Traceback (most recent call last):
  File "<pyshell#85>", line 1, in <module>
    g(2,0,4)
  File "<pyshell#84>", line 1, in g
    def g(x,y,z): return sqrt(f(x,y,z))
TypeError: a float is required
```

Validação de Dados

- ▶ E se, ao invés de retornar uma mensagem como resultado da função, apenas imprimíssemos a mensagem na tela?
- ▶ Teríamos resolvido o problema?

```
>>> from math import sqrt
>>> def f(x, y, z):
    if y == 0:
        print("Divisao por zero
              nao permitida")
    else: return x/y + z
```

```
>>> def g(x, y, z):
    return sqrt(f(x,y,z))
```

```
>>> g(2,0,4)
Divisao por zero nao permitida
Traceback (most recent call last):
  File "<pyshell#85>", line 1, in <module>
    g(2,0,4)
  File "<pyshell#84>", line 1, in g
    def g(x,y,z): return sqrt(f(x,y,z))
TypeError: a float is required
```

Validação de Dados

- ▶ Há uma maneira de se tratar esse problema em Python chama-se tratamento de exceção.
- ▶ A idéia é de que a execução da função tenha um curso normal e outro curso para casos excepcionais.

```
>>> def f(x, y, z):  
    try:  
        if y == 0:  
            print("Divisao por zero nao permitida")  
            raise exception()  
        else: return x/y + z  
    except: pass
```


Validação de Dados

```
>>> from math import sqrt
```

```
>>> def g(x, y, z):
```

```
    try:
```

```
        if f(x,y,z) < 0:
```

```
            print("Operacao de raiz quadrada não permitida")
```

```
            raise exception()
```

```
        else: return sqrt(f(x,y,z))
```

```
    except: pass
```

```
>>> g(2,0,4)
```

```
Divisao por zero nao permitida
```

```
>>> g(2,1,-4)
```

```
Operacao de raiz quadrada não permitida
```

Exercícios

- ▶ Descreva funções para calcular o que se pede, validando para quais valores do domínio a função é definida
 - ▶ Cálculo das raízes da equação de 2º grau (uma função para cada raíz)
 - ▶ Cálculo da função $f(x, y) = 1/(x+y)$



That's all Folks!



nemo