



{ LET'S
{ CODE }

Lógica de Programação Orientada a Objetos

Bem-vindos!



Objetivos

- ✓ Recursão
- ✓ Árvore

O que é Recursão?

- ✓ Recursão é um método de resolução de problemas que envolve quebrar um problema em subproblemas menores e menores até chegar a um problema pequeno o suficiente para que ele possa ser resolvido trivialmente. Normalmente recursão envolve uma função que chama a si mesma. Embora possa não parecer muito, a recursão nos permite escrever soluções elegantes para problemas que, de outra forma, podem ser muito difíceis de programar.

Calculando a soma de uma lista de números



```
def listsum(numList):  
    theSum = 0  
    for i in numList:  
        theSum = theSum + i  
    return theSum  
  
print(listsum([1,3,5,7,9]))
```

Calculando a soma de uma lista de números usando recursão



```
def listsum(numList):  
    if len(numList) == 1:  
        return numList[0]  
    else:  
        return numList[0] + listsum(numList[1:])  
  
print(listsum([1,3,5,7,9]))
```

As três leis da recursão

- ✓ Um algoritmo recursivo deve ter um caso básico;
- ✓ Um algoritmo recursivo deve mudar o seu estado e se aproximar do caso básico;
- ✓ Um algoritmo recursivo deve chamar a si mesmo.

Convertendo um inteiro para um string em qualquer base

- ✓ Reduzir o número original para uma série de números de um dígito.
- ✓ Converter o número de um dígito para um string.
- ✓ Concatenar os strings de um dígito para formar o resultado final.

```
def toStr(n, base):  
    convertString = "0123456789ABCDEF"  
    if n < base:  
        return convertString[n]  
    else:  
        return toStr(n // base, base) + convertString[n % base]  
  
print(toStr(1453, 16))
```

Exercício 1

- ✓ Defina a função `soma_nat` que recebe como argumento um número natural `n` e devolve a soma de todos os números naturais até `n`.

```
soma_nat(5)  
15
```


Exercício 2

- ✓ Defina a função `media_digitos` que recebe como argumento um número natural e devolve a média dos seus dígitos.

```
media_digitos(1234)  
2.5
```

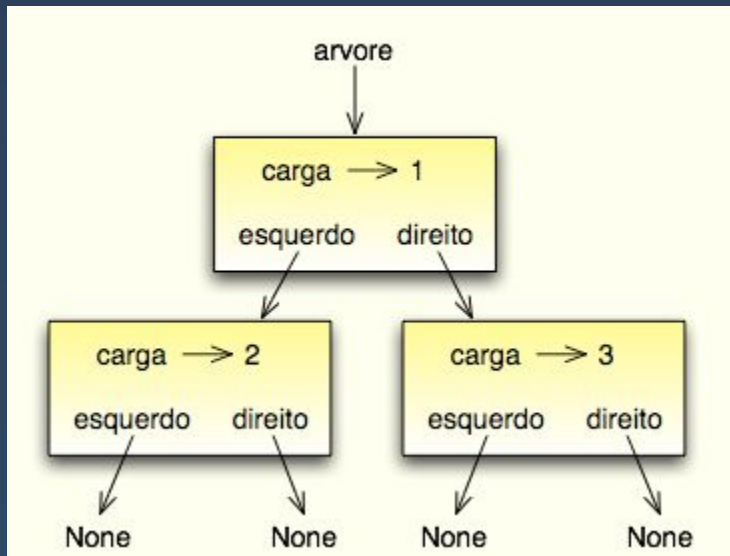
Exercício 3

- ✓ Defina a função `quadrados` que recebe como argumento um número natural `n` e devolve a lista dos `n` primeiros quadrados perfeitos.

```
quadrados(6)  
[1, 4, 9, 16, 25, 36]
```

O que é uma Árvore?

- ✓ Como listas ligadas, árvores são constituídas de células. Uma espécie comum de árvores é a árvore binária, em que cada célula contém referências a duas outras células. Tais referências são chamadas de subárvore esquerda e direita.



Construindo árvores



```
class Tree :  
    def __init__(self, cargo, left=None, right=None) :  
        self.cargo = cargo  
        self.left = left  
        self.right = right  
  
    def __str__(self) :  
        return str(self.cargo)  
  
left = Tree(2)  
right = Tree(3)  
  
tree = Tree(1, left, right)
```

Percorrendo árvores



```
def total(tree):  
    if tree == None: return 0  
    return total(tree.left) + total(tree.right) + tree.cargo
```

Percurso de árvores

```
def print_tree(tree):  
    if tree == None: return  
    print(tree.cargo)  
    print_tree(tree.left)  
    print_tree(tree.right)
```

EXERCÍCIO 1

A startup ABC co. possui 4 funcionários, em que 2 desenvolvedores se reportam a um gerente que se reporta a um diretor. Sabendo que o colaborador possui uma ficha contendo um número de matrícula, um nome, um salário mensal e o seu cargo, construa uma árvore que apresente a hierarquia da empresa e uma função que retorne um dicionário em que a chave são os nomes dos funcionários e o valor são seus salários, organizados em ordem decrescente de salário.

EXERCÍCIO 2

Um jovem estudante de computação tenta criar o motor de inteligência do seu primeiro game, baseado em um RPG em que existem classes de guerreiros que podem vencer outras classes e perder para outras. Dito isto, cada guerreiro possui 3 atributos, sendo o primeiro a sua classe, que pode ser espadachim, arqueiro, escudeiro e mago. O segundo atributo são os pontos de vida, em que todas as classes possuem 100 PVs. Por último, o atributo de dano, em que todos possuem 5, entretanto, o espadachim só causa dano no mago. O arqueiro só causa dano no espadachim, o escudeiro só causa dano no arqueiro e o mago só causa dano no escudeiro. Dito isto, 4 personagens (1 de cada classe) são soltos em uma arena e 1000 lutas randômicas de 1 contra 1 acontecem. Nestas condições, quem seria o primeiro e o segundo colocados, considerando o que tiver mais pontos de vida como o vencedor?