

## Coleta de Lixo – Python

A linguagem de programação Python utiliza um gerenciamento **automático** da memória. Esse mecanismo é baseado em dois componentes: o módulo “**Garbage Collector**” (coletor de lixo) dado como “**gc module**” e o “**Reference Counting**” que é um algoritmo genérico para o tratamento de lixo da memória, baseado em um contador de objetos a serem deletados.

Com isso, o “Garbage Collector” se baseia nos chamados “ciclos de limpeza”, ou seja, em determinados intervalos de tempo esse módulo verifica uma lista de contador de objetos (realizada integralmente pelo “Reference Counting”) e, quando uma referência se torna zero, o **gc** desaloca o objeto da memória.

Em outras palavras, ocorre o seguinte funcionamento: quando um objeto é referenciado na memória, o **Reference Counting** verifica e acrescenta em um contador para aquele objeto. Esse contador depende diretamente da quantidade de referências do objeto no programa. Contudo, caso esse contador torne a obter o valor zero, o **gc** desaloca o objeto da memória.

Por exemplo (segundo o **Python Brasil**):

```
var1 = Objeto() # o objeto Objeto agora tem uma referência chamada var1
var2 = var1

        #o objeto Objeto agora tem duas referências para ele: var1 e var2

var1 = OutroObjeto() # a referência de var1 agora pertence a
        # OutroObjeto, logo, Objeto agora tem só uma referência

var2 = var1 # OutroObjeto agora tem 2 referências e o nosso bom e
        # velho Objeto não tem mais referência nenhuma para ele.
        # Desta forma fica impossível acessá-lo. Ele se perdeu
        # para sempre no 'limbo' da memória do computador.
```

Contudo, esse processo pode gerar alguns problemas como: quando os objetos são referenciados entre si em um ciclo (por exemplo: o objeto A referencia o objeto B e o objeto B referencia o objeto A) ou quando o tempo automático de atuação do **gc** é insuficiente para o programa.

Para solucionar esses problemas, o **Garbage Collector** possui um método de “Coleta de Lixo Geracional” que é baseado em dois conceitos: **geração** e **limite**. O primeiro está relacionado com a geração do coletor de lixo (garbage collector) em que o Python, ao executar um processo de coleta de lixo, verifica se um objeto não pode ser removido. Então esse objeto é passado para uma “segunda geração”. Por padrão, o Python tem três gerações, sendo que o objeto é completamente eliminado após a terceira geração. Já o conceito de limite está associado com o número máximo de objetos permitido no “garbage collector” em que, caso seja extrapolado, o gc acionará um processo de coleta.

Além disso, o processo de funcionamento do “**Garbage Collector**”, bem como o seu limite, pode ser feito de maneira manual. Ou seja, o próprio programador pode acionar manualmente ou desabilitar por completo o processo de coleta de lixo.

### Referências

<https://stackify.com/python-garbage-collection/>

<https://rushter.com/blog/python-garbage-collector/>

<https://www.geeksforgeeks.org/garbage-collection-python/>

<https://wiki.python.org.br/FuncionamentoGarbageCollector>

<https://wiki.python.org.br/ReferenciasCruzadas>

<https://www.tutorialspoint.com/python-garbage-collector-interface-gc>