Computação I - Python
Aula 10 - Entrada, Saída e Programa Principal
Introdução à função principal (main)

Apresentado por: Anamaria Martins Moreira

Produção DCC-UFRJ

Metodologia de referência https://doi.org/10.5753/wei.2016.9683

Usando o computador para fazer cálculos em um laboratório experimental

- Atividade de cinemática:
  - fazer uma série de medidas para verificar a variação de velocidade de um veículo com aceleração constante.
  - fazer um relatório contendo os valores esperados, os valores observados e quanto (porcentagem) o valor observado está diferente do esperado.
  - opcionalmente, fazer um gráfico mostrando algumas dessas informações.

O que precisa ser feito para o relatório

1. Calcular e registrar as velocidades esperadas em cada instante de uma lista de instantes fornecida, dadas também a velocidade inicial do veículo e a aceleração (para cada instante, calcular e registrar a velocidade esperada naquele instante).
2. Calcular e registrar o quanto (percentualmente) as velocidades observadas estavam diferentes das esperadas (para cada instante, calcular e registrar o quanto a velocidade observada estava diferente da esperada naquele instante).
3. Apresentar os resultados sob forma de tabelas ou gráficos.

Cálculos a serem realizados

Podemos identificar nas ações dos itens 1 e 2 acima que serão necessárias funções para calcular:

1.
   - a velocidade esperada num dado instante, dadas também a velocidade inicial do veículo e a aceleração; e
   - as velocidades esperadas em cada instante da lista de instantes fornecida, dadas também a velocidade inicial do veículo e a aceleração.
2.
   - a porcentagem de erro da observação, dadas a velocidade esperada e a observada; e
   - a porcentagem de erro entre os valores observados e os esperados em cada instante da lista fornecida.

Velocidades Esperadas

Funções para calcular a velocidade esperada num dado instante, dadas também a velocidade inicial do veículo e a aceleração; e as velocidades esperadas em cada instante de uma lista de instantes fornecida, dadas também a velocidade inicial do veículo e a aceleração.

```
def velocidade_esperada (v0, a, t):
    '''função que, dadas a velocidade inicial do veículo, a aceleração e o instante,
    devolve a velocidade esperada naquele instante, com até 2 casas decimais.
    float, float, float --> float'''
    return round(v0 + a*t,2)

def calcula_esperadas (v0, a, tempos):
    '''função que, dadas a velocidade inicial do veículo, a aceleração e uma lista de instantes,
    devolve uma lista com as velocidades esperadas em cada instante da lista.
    float, float, list --> list'''
    esperadas = []
    for i in range(len(tempos)):
        list.append(esperadas, velocidade_esperada (v0, a, tempos[i]))
    return esperadas
```

Erros

Funções para calcular a porcentagem de erro da observação, dadas a velocidade esperada e a observada; e os erros entre os valores observados e os esperados em cada instante de uma lista fornecida, juntamente com o instante correspondente, dadas também duas listas contendo respectivamente as velocidades esperadas em cada instante e as velocidades observadas em cada instante.

```
def erro_do_experimento (v_esperada, v_observada):
    '''função que, dadas a velocidade esperada e a observada, devolve a porcentagem de
    erro da observação, com até 2 casas decimais.
    float, float --> float'''
    return round((v_esperada - v_observada)*100/v_esperada, 2)

def erro_ao_logo_do_tempo(v_esperadas, v_observadas, tempos):
    ''' função que, dadas três listas contendo respectivamente as velocidades esperadas em cada instante,
    as velocidades observadas em cada instante e  os instantes considerados, devolve uma lista contendo
    os erros entre os valores observados e os esperados, pareados com os instantes correspondentes.
    list, list --> list'''
    erros = []
    for i in range(len(v_esperadas)):
        erro = (erro_do_experimento (v_esperadas[i], v_observadas[i]), tempos[i])
        list.append (erros, erro)
    return erros
```

Juntando os cálculos - função principal

- Precisamos agora de uma função para completar o algoritmo, chamando as funções anteriores para fazer os cálculos intermediários.
- Como o objetivo dela é especial, e já com vistas a como ela será usada no futuro, não daremos um nome qualquer a ela. A chamada *função principal* do programa é chamada de **main**.
- Apesar de sintaticamente parecida com as outras funções, ela tem este objetivo específico de ser a função que será chamada para a execução de toda a tarefa a ser realizada.

Juntando os cálculos

A função **main** para os cálculos do laboratório de cinemática:

```
def main():
    ''' programa principal para realização dos cálculos do laboratório de cinemática'''

    # dados do experimento
    velocidade_inicial = 10.00
    aceleracao = 0.50
    tempos = [0.00, 1.00, 2.00, 5.00, 10.00, 15.00, 30.00]
    velocidades_esperadas = calcula_esperadas(velocidade_inicial, aceleracao, tempos)

    # resultados observados
    velocidades_observadas = [10.00, 11.10, 12.00, 13.05, 14.00, 15.76, 16.00]

    # cálculo dos erros
    erros = erro_ao_logo_do_tempo(velocidades_esperadas, velocidades_observadas, tempos)

    # retorno(?)
    return erros
```

Usando a função **main**

- Podemos chamar a função main normalmente, no shell do Python.
- Podemos fazer a chamada da main já no próprio arquivo onde está a sua definição, já tentando fazer algo mais automático.

Usando a função **main**

```
def main():
    ''' programa principal para realização dos cálculos do laboratório de cinemática'''

    # dados do experimento
    velocidade_inicial = 10.00
    aceleracao = 0.50
    tempos = [0.00, 1.00, 2.00, 5.00, 10.00, 15.00, 30.00]
    velocidades_esperadas = calcula_esperadas(velocidade_inicial, aceleracao, tempos)

    # resultados observados
    velocidades_observadas = [10.00, 11.10, 12.00, 13.05, 14.00, 15.76, 16.00]

    # cálculo dos erros
    erros = erro_ao_logo_do_tempo(velocidades_esperadas, velocidades_observadas, tempos)

    # retorno(?)
    return erros

main()
```

Isso fará com que a função main seja executada automaticamente quando damos o *Run* do módulo...

Usando a função **main**

- ... mas não veremos o seu resultado no shell do Python
- Além disso, a função main possui um comportamento bastante especial quanto ao recebimento de argumentos.
- No nosso esquema de uso, dentro o IDLE, a main não possui parâmetros.
- Foi necessário definir todos os dados do experimento diretamente na função main.
- Para mudar qualquer informação, o código precisará ser editado, alterando os valores das variáveis desejadas.

Entrada e saída de dados

Essas limitações nos levam ao próximo assunto, a necessidade de prover interação com o usuário, mesmo em programas simples como o do laboratório de Física. O que queremos é algo como:

```
def main():
    ''' programa principal para realização dos cálculos do laboratório de cinemática'''

    # dados do experimento
    # pedir para o usuário fornecer os dados do experimento
    # velocidade_inicial = ?
    # aceleracao = ?
    # tempos = ?
    velocidades_esperadas = calcula_esperadas(velocidade_inicial, aceleracao, tempos)

    # resultados observados
    # pedir para o usuário fornecer os resultados observados
    # velocidades_observadas = ?

    # cálculo dos erros
    erros = erro_ao_logo_do_tempo(velocidades_esperadas, velocidades_observadas, tempos)

    # mostrar os resultados para o usuário
    # sob forma de tabelas, gráficos, etc.

main()
```

Entrada e saída de dados

Veremos como completar esse código com ações de comunicação com o usuário na continuação desta aula.

Boa aula!