

# Aula 15

## Exercício Prático – integração



# Algoritmos e Estrutura de Dados I

1º Semestre – CDN



Prof. Dr. Dilermando Piva Jr.

# Conteúdo Programático - Planejamento

Conteúdo Programático		
Semana	Data	Temas/Atividades
1	12/08	Acolhimento e Boas-vindas! Introdução a Disciplina. Formas de Avaliação e Percorso Pedagógico.
2	19/08	Pensamento Computacional. O que é qual sua importância para Ciência de Dados
3	26/08	Primeiro Programa – Variáveis, Tipos de Dados e Saída em Python
4	30/08	<i>/reposição/ Introdução a Computação e representação da informação – História e evolução da computação. A informação e sua representação. Conversão entre bases.</i>
5	02/09	Operadores, Cálculo Simples e Entrada de Dados. Estruturas sequenciais.
6	09/09	Tomando Decisões: Estrutura Condicional (if/else)
7	16/09	Repetição de Ações: Estrutura de Repetição – Introdução aos loops (for e while)
8	23/09	Prática Integrada – Construindo um Jogo Simples (Adivinhação) – Pj1
9	30/09	Estruturas de Dados Parte 1 – listas e sequências (e strings)
10	07/10	Estruturas de Dados Parte 2 – Dicionários e Dados Estruturados
11	14/10	Integração – Projeto de Análise de Dados Simples com Listas e Dicionários – Pj2
12	21/10	<b>Primeira Avaliação Formal. (P1).</b> Correção da Avaliação após o intervalo.
13	28/10	Modularização do código – Introdução a Funções
14	04/11	Parâmetros, Retornos e Boas Práticas de Funções // Módulos e Pacotes em Python
15	11/11	Exercício de Aplicação – Pj3
16	18/11	<del>Introdução a Ciência de Dados – partes 1 e 2 – Pandas, Numpy, Pipeline e exemplos – Pj4</del>
17	25/11	<b>Semana de Apresentação PI de CDN</b>
18	02/12	<i>Aula destinada ao desenvolvimento do Projeto (p/ recuperação nota) → não haverá aula presencial</i>
19	09/12	<b>Segunda Avaliação Formal (P2).</b> Correção da Avaliação após o intervalo
20	16/12	<b>Exame / Avaliação Substitutiva.</b> Divulgação do Resultado Final.

# Exercício Prático

Funções e Processo de Integração de Equipes

# Função `valida_cpf()`

Faça uma função em python chamada "**`valida_cpf()`**" que recebe como parâmetro uma string no formato "999.999.999-99". Essa função faz a verificação se esse cpf é válido, utilizando as regras da receita federal brasileira, e retorna True se for válido e False se não for válido. A função deve ser salva dentro de um arquivo (script) com o nome `valida_cpf.py` no diretório corrente.

Vamos usar como exemplo, um CPF fictício "529.982.247-25".

Validação do Primeiro Dígito:

$$5 * 10 + 2 * 9 + 9 * 8 + 9 * 7 + 8 * 6 + 2 * 5 + 2 * 4 + 4 * 3 + 7 * 2$$

***Soma % 11 = resto → se resto < 2 → dig1=0 caso contrário dig1=11-resto***

Validação do Segundo Dígito:

$$5 * 11 + 2 * 10 + 9 * 9 + 9 * 8 + 8 * 7 + 2 * 6 + 2 * 5 + 4 * 4 + 7 * 3 + 2 * 2$$

***Soma % 11 = resto → se resto < 2 → dig2=0 caso contrário dig2=11-resto***



# Função `valida_data()`

Faça uma função em linguagem python chamada "**`valida_data()`**" que recebe como parâmetro uma string no formato "dd/mm/aaaa". Essa data refere-se a data de nascimento de uma pessoa em um cadastro. Pessoas com menos de 18 anos não podem se cadastrar. Além disso, a função deve verificar se a data digitada é válida. O retorno da função será `True` se a data for válida e `False` se a data não for válida.

Recomenda-se a utilização da função `datetime` do pacote `datetime`:

```
# Verificar a idade mínima de 18 anos
from datetime import datetime
hoje = datetime.now().date()
data_nasc = datetime(ano, mes, dia).date()
idade = hoje.year - data_nasc.year - ((hoje.month, hoje.day) < (data_nasc.month, data_nasc.day))
```

# Função `exibir_menu()`

Faça uma função em linguagem python que exibe um menu de opções.

- 1 – `Cadastrar`
- 2 – `Exibir Frase`
- 3 – `Sair`

Essa função deve verificar se a opção digitada está dentro do limite entre 1 e 3. Se for um valor diferente, reexibir o menu. Se o usuário escolheu uma das opções válidas, a função deve retornar o valor inteiro correspondente a opção do usuário.

# Script Principal

Supondo que cada uma das funções anteriores (`exibir_menu()`, `valida_data()` e `valida_cpf()`) estejam em um arquivo com o mesmo nome da função com a extensão `.py`, no mesmo diretório corrente.

Crie um script em python que vai utilizar essas funções para realizar uma das opções de `exibir_menu()`:

- Caso a opção seja **1 - Cadastrar**, o cadastro deve conter: Nome, Sobrenome, cpf, data de nascimento e renda bruta. Deve-se utilizar as funções para fazer a validação dos campos correspondentes (cpf e data de nascimento).
- Caso a opção seja **2 - Exibir Mensagem**, o script deve exibir uma das 4 mensagens motivacionais abaixo, de forma aleatória:
  - "A persistência realiza o impossível"
  - "Seus sonhos não precisam de plateia, eles só precisam de você"
  - "A persistência é o caminho do êxito"
  - "No meio da dificuldade encontra-se a oportunidade"
- Caso a opção seja **3 - Sair**, o script deve exibir a mensagem "Bye bye!" e encerrar sua execução.

## Passo 1 – Calcular DV1:

Vamos primeiramente calcular o dígito verificador #1.  
Precisamos usar os nove primeiros dígitos do CPF para isso:

6	6	7	5	5	6	3	1	7
---	---	---	---	---	---	---	---	---

Multiplicamos os dígitos pela sequência decremental de 10 a 2

6	6	7	5	5	6	3	1	7	x
10	9	8	7	6	5	4	3	2	=
60	54	56	35	30	30	12	3	14	

Efetuamos o somatório dos valores obtidos para cada dígito multiplicado:

$$60 + 54 + 56 + 35 + 30 + 30 + 12 + 3 + 14 = 294$$

Calculamos o módulo de 11 do resultado (resto da divisão inteira do resultado por 11):

$$294 \% 11 = 8$$

Finalmente, subtraímos esse módulo de 11:  $11 - 8 = 3$

Assim, **3** é o dígito verificador 1 (DV1).



# Próxima Aula



- Ler os conteúdos da semana 16 e ver os vídeos indicados.
- No curso online, ver a videoaula correspondente!

***Boa semana e bons estudos!!***