**Bruna Olivieri**

**SENAI JANDIRA**

**PROGRAMAÇÃO EM PYTHON**

**31/01/24 - quarta-feira (Aula 1)**

Variável - reserva um espaço na memória para armazenar um dado.

**05/02/2024 - segunda-feira (Aula 2)**

entrada - processamento - saída

entrada - teclado, pendrive, internet, microfone, câmera, leitor

biométrico (fingerprint), etc.

saída - monitor, alto-falante, impressora, etc.

Uma variável pode ter até 256 caratecteres.

O nome de uma variável no python deve começar com letra maiúscula

ou minúscula ou pelo símbolo underscore "\_".

Exemplo: salario\_do\_funcionario

SalarioDoFuncionario (camel case)

**Tipagem de variáveis**

**Numéricos:** inteiros ou ponto flutuante (decimais) - 234, 1345,

2... (inteiro) | 2.45, 56.9987... (ponto flutuante - decimal)

**Texto:** qualquer tipo de caractere

**Lógicos:** True ou False (booleanos)

Linguagem de programação brasileira: Lua (mais utilizada no

desenvolvimento de jogos). PUC-Rio.

Exercícios: cálculo de IMC e cálculo de média.

ctrl + espaço = preenchimento automático.

**07/02/2024 – quarta-feira (Aula 3)**

Exercício - consumo de combustível

**Antes de começar programar é necessário entender o problema.**

**Clean Code –** código limpo.

Baixar o PyCharm Community Edition.

Fazer o cálculo na mão para conferir se o programa está certo.

**Concatenação**

**%s - coringa / pode ser qualquer string**

print("Meu nome é %s, tenho %s anos." % (nome, idade))

Tem que ser na ordem das variáveis. Se for só uma variável não precisa dos parênteses.

print(**f**"Meu nome é **{**nome**}**, tenho **{**idade**}** anos. Já disse, meu nome é **{**nome**}**.")

**É a opção mais legível.**

f = formato

f.string

**Arredondamento**

print("O valor de PI é %s." % pi)

print("O valor de PI é %.2f" % pi)

%.2f = arredondamento das casas decimais.

Não remove casas decimais em operações financeiras.

Só faz arredondamento para exibição para o usuário.

print(f"O valor de PI é {pi:.2f}.")

**:** = separador de argumento.

**Criação de conta no GitHub**

**GitHub** - Repositório remoto de arquivos.

**Git** – ferramenta que roda no computador.

**Entrega do dia** – commit (guarda um histórico de tudo o que foi feito hoje.) Dessa forma eu garanto as versões do meu arquivo.

Plataforma colaborativa.

**git -v**

**git -version**

Comandos para verificar se o Git está instalado no computador.

**Valores lógicos**

- True

- False

**Operadores relacionais**

**==** igual

!= diferente

**>** maior

**<** menor

**>=** maior igual

**<=** menor igual

**19/02/2024 – segunda-feira (Aula 4)**

**Blocos Condicionais (If / Else)**

**If –** se

**Else –** senão

True

False

**Identação**

Serve para identificar o que está dentro e o que está fora de um bloco de instruções.

**Fluxograma**

INÍCIO

IDADE = 16

F

V

IDADE < 18

VOCÊ É  
MENOR!

VOCÊ É  
MAIOR!

**Exercício:** Calcular a média final de um aluno. Exibir se foi aprovado ou reprovado. Condição para ser aprovado: média final ser >= 7.

**Exercício:** Calcular IMC e exibir a classificação.

**Operador OR**

Se uma instrução for verdadeira, o resultado será True.

Se todas forem falsas, o resultado será False.

1 > 3 **or** 5 < 3

False

10 > 3 **or** 5 < 2

True

**Operadores lógicos**

**or** – ou

Se apenas uma instrução for verdadeira, o resultado é verdadeiro.

**and** – e

Só é verdadeiro quando todas instruções forem verdadeiras.

**not** – não

Negação. Inverte o resultado.

**Exemplo 1**

(4 > 3 or 5 < 10) and (10 > 1 or 5 < 3)

**V**  and **V**

**V**

**Exemplo 2**

(4 > 3 or 5 < 10) and (10 > 1 and 5 < 3)

**V**  and **F**

**F**

**Exemplo 2**

not ((4 > 3 or 5 < 10) and (10 > 1 and 5 < 3))

not (**V**  and **F)**

**V**

**Exercício:** Dia da semana. Ao digitar um número de 0 a 6, vai aparecer o dia da semana correspondente.

**0 –** domingo

**1 –** segunda

**2 –** terça

**3 –** quarta

**4 –** quinta

**5 –** sexta

**6 –** sábado

**Nome do usuário no Windows:** não utilizar espaço.

**elif –** mistura do else com o if. (senão, se...)

Comparar outras condições.

**Exercício:** Acrescentar no exercício Média Final a possibilidade de recuperação. Média < 5 = reprovado. Média >= 7 = aprovado.

**21/02/2024 – quarta-feira (Aula 5)**

**Git**

VCS (Version Control System). Sistema de controle de versão.

Criador: Linus Torwalds

Instalar o Git na máquina.

Repositório: lugar destinado ao armazenamento de algo.

**Commit:** tirar uma foto do arquivo naquele exato momento. Em cada commit se faz uma descrição do que foi feito.

**Branch:** linha do tempo onde se tem a versão final do projeto. Main/Master. Ramificação. Cada um trabalha em sua branch sem impactar na branch principal (main).

**Nerge:** Juntar.

c:  
dir  
cd Users

dir

cd celso.alunos

cd python

cls

dir

python

exit()

git status

Se estiver instalado aparecerá a seguinte mensagem:

fatal: not a git repository (or any of the parent directories): .git

git init

dir

git status

git add <arquivo>

git status

git add .

notepad teste.py

git commit -m “Versão do projeto 1.”

git status

git log (histórico de commit)

git checkout 715c2 (iniciais do commit / voltar na linha do tempo)

git log

git checkout master (voltar para o presente)

git remote -v (configuração do repositório remoto / caminho do repositório remoto)

git remote add origin <https://github.com/brunaolivieri/curso_python.git> (criação da variável origin com a URL do repositório)

fetch – obter

push – empurrar

git push origin master

. – pasta onde está

.. – pasta anterior

cd .. – vai para a pasta anterior

md projeto-python (criação de pasta)

cd projeto-python (entrar na pasta)

dir

git clone https://github.com/brunaolivieri/curso\_python.git

dir

**Github**

Criar repositório.

Copia URL.

Colar a URL no prompt.