Problemática

Problemática

Em uma fábrica de produtos químicos, é essencial manter a temperatura de um reator dentro de limites específicos para garantir a qualidade e a segurança dos produtos fabricados. No entanto, variações inesperadas na temperatura podem ocorrer devido a mudanças na demanda, falhas no equipamento ou condições ambientais. Para evitar danos ao equipamento e garantir a consistência do produto, é necessário implementar um sistema de controle de temperatura automatizado.

Riscos

Riscos esperados

- Erros Manuais e Inconsistências nos Dados:
  - Sem um sistema automatizado, os dados podem ser inseridos manualmente, aumentando o risco de erros de digitação e inconsistências nos dados, o que pode levar a análises imprecisas e tomadas de decisão equivocadas.
- Demora e Ineficiência no Processamento de Dados:
  - A ausência de um sistema automatizado pode resultar em processos lentos e ineficientes de entrada, processamento e análise de dados, o que pode impactar negativamente a produtividade e eficácia das operações de engenharia.
- Falta de Rastreabilidade e Auditoria:
  - Sem um sistema centralizado para gerenciar e registrar dados, pode haver uma falta de rastreabilidade e capacidade de auditoria, dificultando a identificação de quem inseriu ou modificou os dados, o que pode comprometer a integridade e segurança dos dados.
- Dificuldade na Tomada de Decisão:
  - A falta de acesso rápido e preciso a dados relevantes pode dificultar a tomada de decisões informadas e oportunas, levando a atrasos ou decisões inadequadas que podem afetar o desempenho e a segurança dos projetos de engenharia.

Riscos esperados

- Vulnerabilidades de Segurança:
  - Sistemas manuais podem estar sujeitos a vulnerabilidades de segurança, como acesso não autorizado, vazamento de dados sensíveis e perda de informações críticas, o que pode representar um risco significativo para a confidencialidade e integridade dos dados.
- Custos Operacionais Elevados:
  - A dependência de processos manuais pode resultar em custos operacionais mais elevados devido ao aumento do tempo e dos recursos necessários para realizar tarefas de entrada, processamento e análise de dados, além dos custos associados à correção de erros e retrabalho.
- Conformidade e Responsabilidade Legal:
  - A falta de um sistema adequado para gerenciar e proteger dados pode resultar em questões de conformidade e responsabilidade legal, especialmente em setores regulamentados, onde a precisão, integridade e segurança dos dados são requisitos críticos.

Benefícios Esperados

Benefícios esperados

- Melhoria na Eficiência Operacional:
  - Com um sistema automatizado, os processos de gerenciamento de dados e tomada de decisão podem ser simplificados e agilizados, resultando em uma maior eficiência operacional.
- Redução de Erros Humanos:
  - A automação de tarefas repetitivas e propensas a erros pode reduzir significativamente a ocorrência de erros humanos, melhorando assim a precisão e confiabilidade dos dados e das operações.
- Aumento da Produtividade:
  - Com menos tempo gasto em tarefas manuais e rotineiras, os profissionais podem se concentrar em atividades mais estratégicas e de maior valor agregado, aumentando assim a produtividade geral da equipe.

Benefícios esperados

- Melhoria na Qualidade dos Serviços:
  - Ao garantir uma gestão mais eficaz dos dados e processos, o sistema pode contribuir para a melhoria da qualidade dos serviços prestados pela empresa, resultando em maior satisfação do cliente.
- Acesso Mais Rápido às Informações:
  - Com um sistema centralizado e organizado, os usuários podem acessar rapidamente as informações necessárias, facilitando a tomada de decisões e a realização de análises.
- Maior Conformidade Regulatória:
  - A implementação de um sistema pode ajudar a garantir que a empresa esteja em conformidade com regulamentações e padrões específicos da indústria, reduzindo assim o risco de multas e penalidades..

Analisando o Problema

Requisitos e Restrições

Requisitos

Facilidade de Uso: O sistema deve ser intuitivo e fácil de usar para os operadores da fábrica, com interfaces claras e simples.

Resposta Rápida: O sistema deve responder rapidamente a variações na temperatura, ajustando os parâmetros conforme necessário para evitar grandes flutuações.

Integração com Sistemas Existentes: O sistema de controle de temperatura deve ser integrado de forma eficiente com outros sistemas de automação e controle existentes na fábrica.

Monitoramento Remoto: Deve ser possível monitorar e controlar o sistema de controle de temperatura remotamente, permitindo uma intervenção rápida em caso de problemas.

Restrições

- Consumo de Energia: Deve-se considerar o consumo de energia do sistema de controle de temperatura e procurar maneiras de utilizá-lo para reduzir os custos operacionais.
- Treinamento de Operadores: Restrições de tempo e recursos podem limitar o treinamento dos operadores para usar efetivamente o sistema de controle de temperatura.
- Disponibilidade de Tecnologia: As opções de tecnologia disponíveis podem ser limitadas por fatores como disponibilidade no mercado e compatibilidade com sistemas existentes.
- Segurança: A segurança dos funcionários e do ambiente de trabalho deve ser uma prioridade, com medidas para mitigar riscos relacionados ao manuseio de equipamentos e substâncias químicas..

Solução Proposta

Solução Proposta

Identificar as necessidades específicas de controle de temperatura na fábrica de produtos químicos.

Realizar um levantamento das variáveis que afetam a temperatura do compressor (corrente do motor, pressão do compressor, etc.).

Selecionar os sensores de temperatura adequados e determinar sua localização estratégica no ambiente de produção.

Escolher o sistema de automação industrial mais adequado para o controle de temperatura, levando em consideração a integração com outros sistemas da fábrica.

Desenvolver algoritmos de controle de temperatura para garantir a estabilidade e uniformidade do processo.

Especificar as entrada de dados, as saída de dados, os tipos de dados, os identificadores e variáveis, além dos operadores aritméticos e expressões.

Monitorar continuamente os dados de temperatura e realizar ajustes conforme necessário para manter os parâmetros dentro dos limites desejados.

Plano de Ação

Desenvolver um algoritmo

- Realizar a definição das variáveis
- Identificar os tipos das variáveis
- Identificar os valores limites
- Identificar os intervalos por variáveis
- Definir os indicadores necessários para o monitoramento

Benefícios esperados

- Melhoria na Eficiência Operacional:
  - Com um sistema automatizado, os processos de gerenciamento de dados e tomada de decisão podem ser simplificados e agilizados, resultando em uma maior eficiência operacional.
- Redução de Erros Humanos:
  - A automação de tarefas repetitivas e propensas a erros pode reduzir significativamente a ocorrência de erros humanos, melhorando assim a precisão e confiabilidade dos dados e das operações.
- Aumento da Produtividade:
  - Com menos tempo gasto em tarefas manuais e rotineiras, os profissionais podem se concentrar em atividades mais estratégicas e de maior valor agregado, aumentando assim a produtividade geral da equipe.

Benefícios esperados

- Melhoria na Qualidade dos Serviços:
  - Ao garantir uma gestão mais eficaz dos dados e processos, o sistema pode contribuir para a melhoria da qualidade dos serviços prestados pela empresa, resultando em maior satisfação do cliente.
- Acesso Mais Rápido às Informações:
  - Com um sistema centralizado e organizado, os usuários podem acessar rapidamente as informações necessárias, facilitando a tomada de decisões e a realização de análises.
- Maior Conformidade Regulatória:
  - A implementação de um sistema pode ajudar a garantir que a empresa esteja em conformidade com regulamentações e padrões específicos da indústria, reduzindo assim o risco de multas e penalidades..

Aula 01

Primeiros passos em programação

Tópicos

- Introdução a Programação com Python
- Instalando e usando o interpretador
- Entrada de dados
- Saída de dados
- Tipos de dados
- Identificadores e variáveis
- Operadores aritméticos e expressões

Introdução a Programação com Python

Programa em Python

- Um programa Python consiste em um ou mais módulos. Um módulo é apenas um arquivo de código, que pode incluir instruções, definições de função e definições de classe.
- Um pequeno programa, também ser chamado de script, pode estar contido em um módulo. Programas mais longos e complexos geralmente incluem um módulo principal e um ou mais módulos de suporte.
- O principal contém o ponto de partida da execução do programa. Módulos de suporte contêm definições de funções e classes.

Instalando e usando o interpretador

Instalando o Python

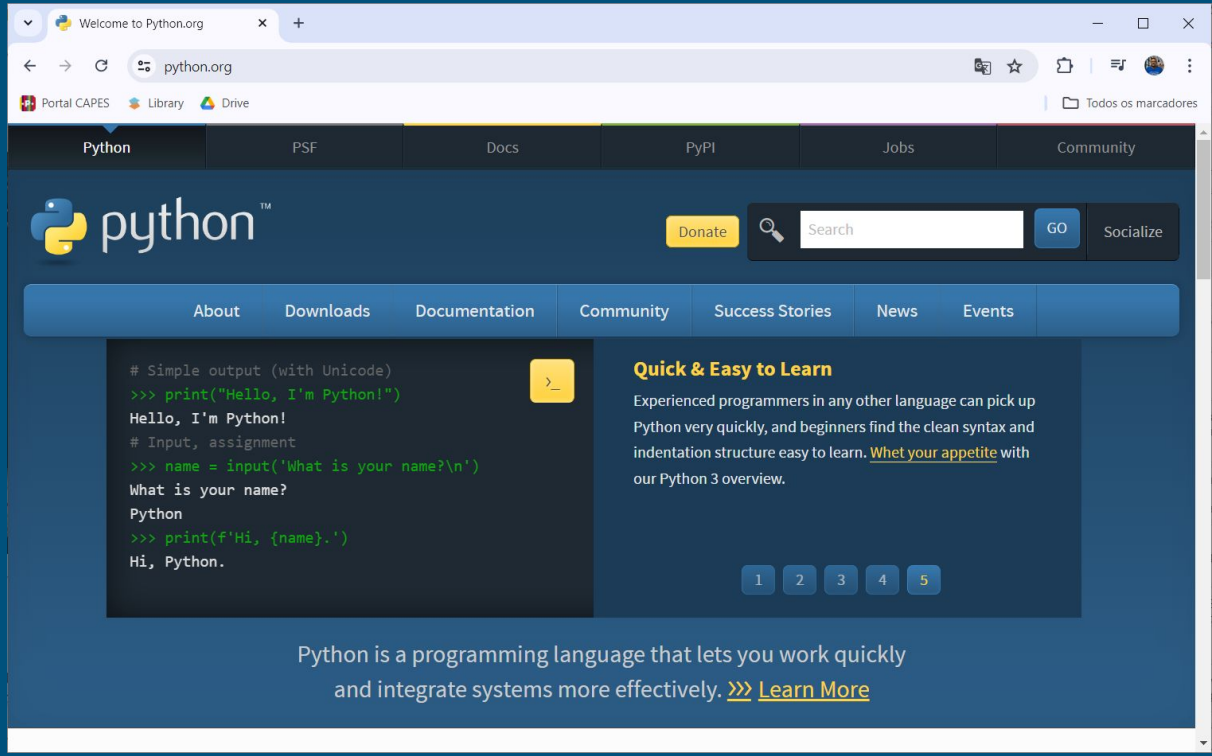
- Python é uma linguagem extremamente versátil com uma vasta gama de ferramentas e plataformas disponíveis para diversos tipos de desenvolvimento e pesquisa. A escolha da plataforma ou ferramenta depende das necessidades específicas do projeto e das preferências pessoais do desenvolvedor.

Instalando o Python

- Ambientes de Desenvolvimento Integrado (IDEs)
  - PyCharm
  - Visual Studio Code (VS Code)
  - Spyder
- Editores de Texto
  - Sublime Text.
  - Atom.
  - Notepad++
- Notebooks Interativos
  - Jupyter Notebook
  - Google Colab
- Ambientes de Linha de Comando
  - Python Shell
  - IPython

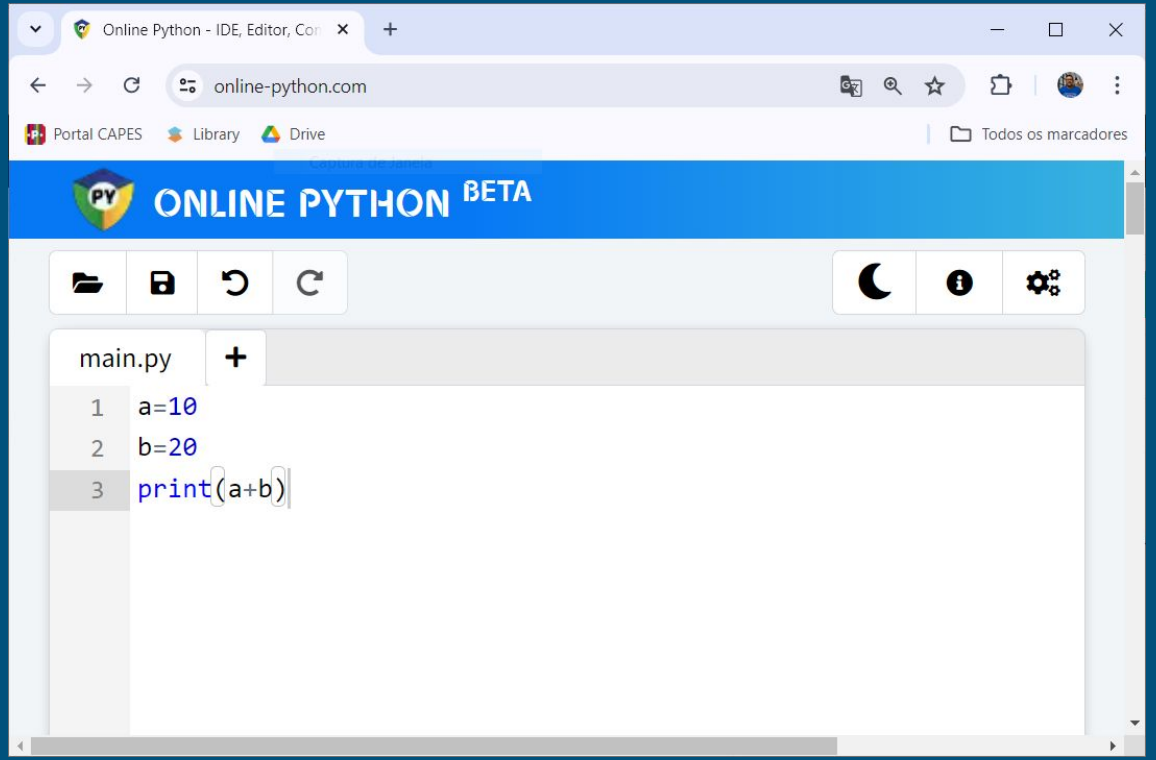
Instalando o Python

- Link de acesso:
  - https://www.python.org/



Executando um código em python

- Ambientes on-line
  - https://www.online-python.com/
- Ambientes desktop
  - Instalação de ferramentas na sua máquina local

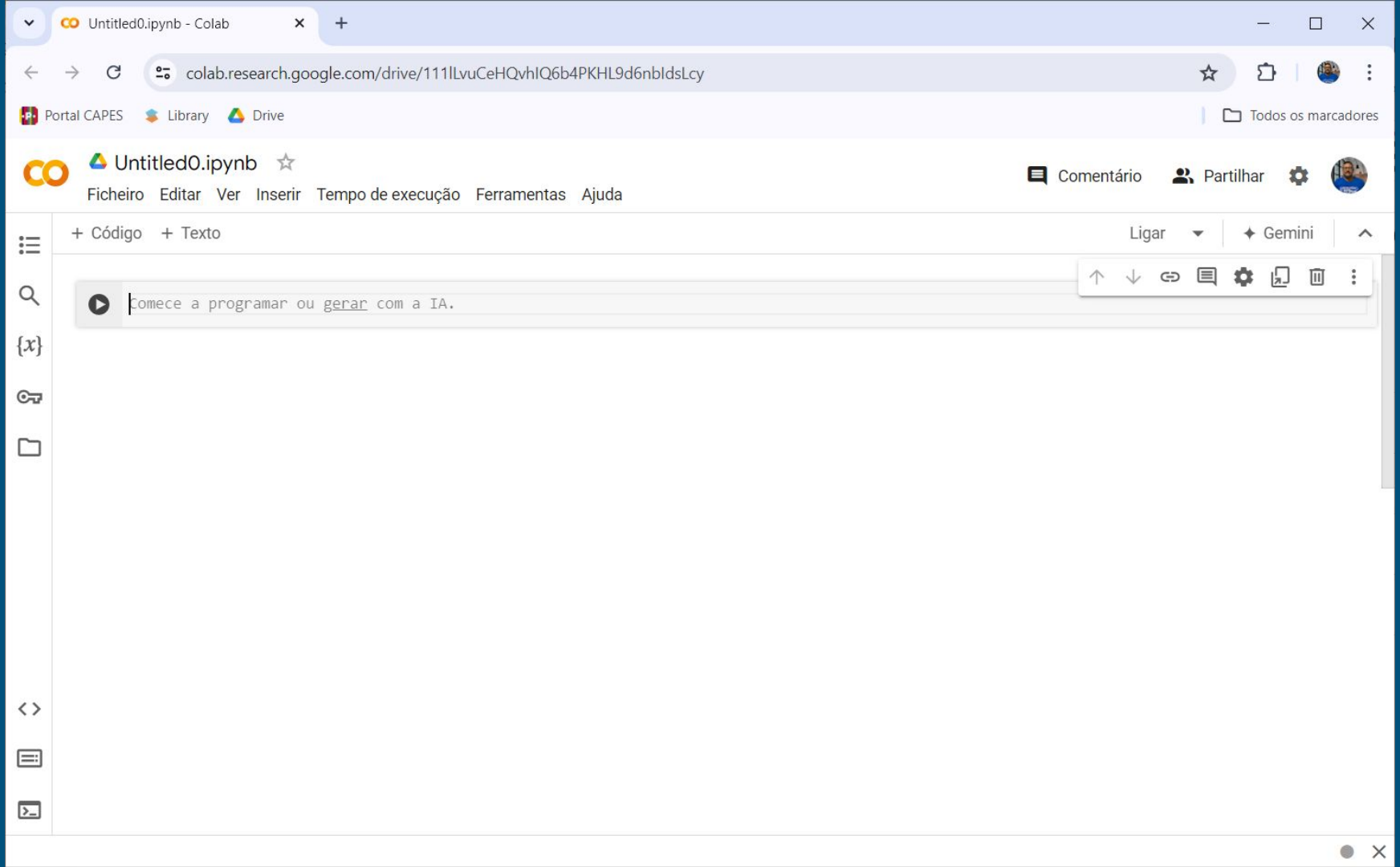


Notebooks Interativos - Google Colab

- O Google Colab, abreviação de Google Colaboratory, é uma plataforma gratuita oferecida pelo Google que permite a execução de código Python diretamente no navegador.
- Com o Colab, você pode escrever e executar código Python em notebooks Jupyter, acessar GPUs gratuitamente para acelerar tarefas intensivas em computação, compartilhar e colaborar em notebooks com outras pessoas em tempo real, e facilmente importar e exportar notebooks do Google Drive ou do GitHub.
- Além disso, o Google Colab suporta bibliotecas populares de aprendizado de máquina e ciência de dados, como TensorFlow, Keras, PyTorch, entre outras, tornando-o uma ferramenta poderosa para prototipagem rápida, aprendizado e experimentação.

O Google Colab

1. Acesse o Site do Google Colab pelo link colab.research.google.com.
2. Login com a Conta Google. Use suas credenciais do Google (Gmail) para entrar.
3. Criar um Novo Notebook: Na página inicial do Google Colab. Clique nela para criar um novo notebook Jupyter.
4. Nomear o Notebook: Na parte superior do novo notebook, você verá o nome padrão "Untitled".
5. Escrever e Executar Código, você pode começar a escrever seu código Python nas células do notebook. Para adicionar uma nova célula de código, clique em "+ Código". Escreva o seu código na célula e pressione "Shift + Enter" para executar.


Untitled0.ipynb - Colab
colab.research.google.com/drive/111lLvuCeHQvhIQ6b4PKHL9d6nbIdsLcy
Portal CAPES
Library
Drive
Todos os marcadores
Untitled0.ipynb
Ficheiro Editar Ver Inserir Tempo de execução Ferramentas Ajuda
Comentário
Partilhar
+ Código + Texto
Ligar
Gemini
Comece a programar ou gerar com a IA.

02_Pratica_Primeiros_Passos - C

colab.research.google.com/drive/1HUUX83Tj8nK-fX609Y3GTovU3Q7x26Bv#scrollTo=fJ3PiP58wNmK

Portal CAPES | Library | Drive | Todos os marcadores

02_Pratica_Primeiros_Passos

Ficheiro Editar Ver Inserir Tempo de execução Ferramentas Ajuda

Comentário | Partilhar

+ Código + Texto

RAM Disco | Gemini

Primeiros Passos na Programação

Aqui vamos apresentar todos os tópicos discutidos em sala de aula pelo professor, com uma ou duas células com exemplos práticos da aplicação do comando ou recurso. A seguir a lista de tópicos trabalhados:

Instalando e usando o interpretador

```
[5]  # Executando um comando
     print('Olá Mundo')
```

```
[4]  # Executando um conjunto de comandos
     valor=6
     for i in range(1,11):
       print(i,' * ',valor,'=',i*valor)
```

```
[10] # Executando com erro de syntaxe
     valor=6
     for i in range(1,11,2,3):
       print(i,' * ',valor,'=',i*valor)
```

Entrada de dados

0 s concluído à(s) 22:25

Interpretador

- Executando com o interpretador (Shift+Enter) ou (Ctrl+ Enter)
  - sem erro
  - com erro

```
# Executando com erro de syntaxe
valor=6
for i in range(1,11,2,2):
  print(i,' * ',valor,'=',i*valor)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-9-e8719e5508bd> in <cell line: 3>()
      1 # Executando com erro de syntaxe
      2 valor=6
----> 3 for i in range(1,11,2,2):
      4   print(i,' * ',valor,'=',i*valor)

TypeError: range expected at most 3 arguments, got 4
```

```
[5] # Executando um comando
    print('Olá Mundo')
```

```
Olá Mundo
```

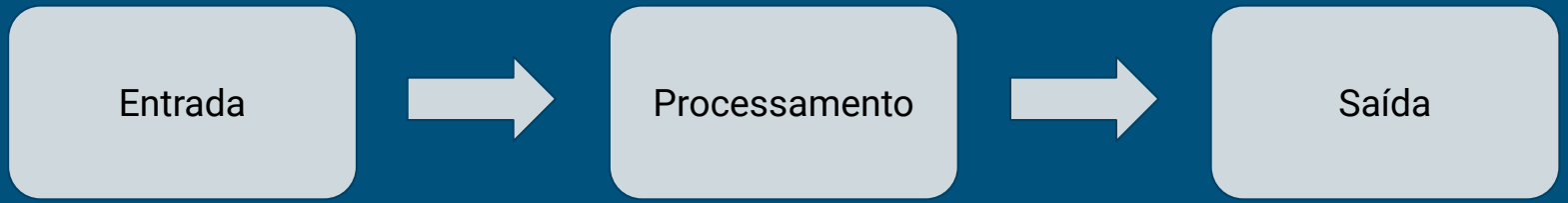
```
[4] # Executando um conjunto de comandos
    valor=6
    for i in range(1,11):
      print(i,' * ',valor,'=',i*valor)
```

```
1  *  6 = 6
2  *  6 = 12
3  *  6 = 18
4  *  6 = 24
5  *  6 = 30
6  *  6 = 36
7  *  6 = 42
8  *  6 = 48
9  *  6 = 54
10  *  6 = 60
```

O processamento de dados

Programa

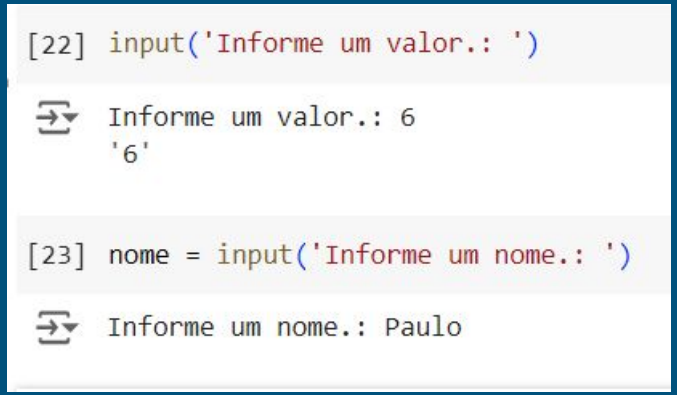
- O processamento de dados consiste, basicamente, em um conjunto de três partes: fundamentais (Entrada, Processamento e Saída) que são articuladas para se atingir as expectativas desejadas.



Entrada de dados

Entrada de dados

- A função de entrada padrão do Python é o comando “input” espera que o usuário insira um texto pelo teclado. Quando o usuário pressiona a tecla “Enter”, a função retorna uma string contendo os caracteres inseridos.
- Esta função recebe uma string opcional como argumento a imprime, sem uma quebra de linha, para solicitar a entrada do usuário.



Saída de dados

Saída de dados

- A função de saída padrão é o comando “print(conteúdo)” exibe seus argumentos no console.
- Ela permite um número variável de argumentos.
- Por padrão, print termina a saída com uma nova linha.
- O “\n” é utilizado para adicionar uma nova linha
  para melhorar a legibilidade da saída.

```
valor = 200
print('Informe um valor')
print(valor)
print('O valor informado foi', valor)
```

```
Informe um valor
200
O valor informado foi 200
```

```
[3] valor = 200
    print('O valor informado foi', valor)
```

```
O valor informado foi 200
```

```
[2] valor = 200
    print('O valor informado foi\n', valor)
```

```
O valor informado foi
 200
```

Tipos de dados

Tipos

- Um valor é uma das coisas básicas com as quais um programa trabalha, como uma letra ou um número. Os valores aqui apresentado no exemplo ao lado, representam os seguinte tipos:
  - Inteiros
  - Strings
  - Float
  - Boolean

```
print(type(A))
print(type(B))
print(type(C))
print(type(D))
```

```
<class 'int'>
<class 'str'>
<class 'float'>
<class 'bool'>
```

```
[18] A = 23
     B = 'Universidade'
     C = 2.5
     D = True
```

```
[19] print(A)
     print(B)
     print(C)
     print(D)
```

```
23
Universidade
2.5
True
```

Identificadores e variáveis

Identificadores

- Identificadores são nomes usados para identificar variáveis, funções, classes, módulos ou outros objetos.
- Variáveis são usadas para armazenar dados que podem ser referenciados e manipulados pelo nome.

Identificadores

- Regras para Identificadores:
  - Devem começar com uma letra (a-z, A-Z) ou um sublinhado (_).
  - Podem conter letras, dígitos (0-9) e sublinhados (_).
  - São sensíveis a maiúsculas e minúsculas (case-sensitive).
  - Não podem ser palavras-chave reservadas da linguagem Python.
- Palavras-chave Reservadas:
  - Palavras que têm um significado especial em Python e não podem ser usadas como identificadores.
  - Exemplos incluem “and”, “or”, “if”, “else”, “for”, “while”, “def”, “return”, etc.

```
[16] # Atribuição de variáveis
     nome = "João"
     idade = 25
     altura = 1.75
     estudante = True

     # Imprimindo os valores
     print(nome)
     print(idade)
     print(altura)
     print(estudante)
```

```
João
25
1.75
True
```

Variáveis e palavras-chave

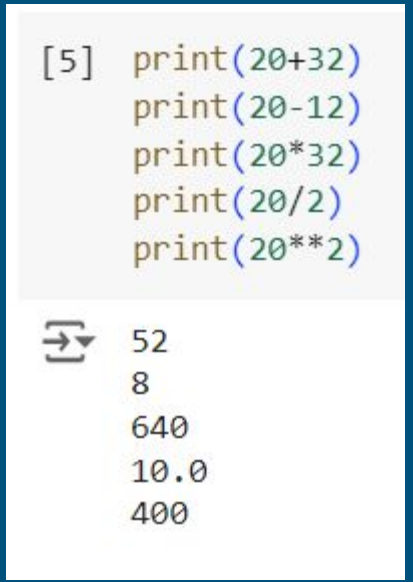
- Nomes de variáveis podem ser arbitrariamente longos. Eles podem conter letras e números, mas precisam começar com uma letra. É legal usar letras maiúsculas, mas é uma boa ideia começar nomes de variáveis com uma letra minúscula (você verá o porquê mais tarde).
- O caractere sublinhado (_) pode aparecer em um nome. Ele é frequentemente usado em nomes com várias palavras, como nome_completo ou endereco_de_atendimento.

```
[15] nome_completo = input('Informe o seu nome completo: ')
     endereco_de_atendimento = input('Qual o endereço de entrega')

     Informe o seu nome completo: Nisston Moraes
     Qual o endereço de entregaRua Manoel Cavalcante
```

Operadores aritméticos e expressões

Operadores e operandos

- Operadores são símbolos especiais que representam cálculos como adição e multiplicação. Os valores aos quais o operador é aplicado são chamados operandos.
- Os operadores são: + , - , * , / e ** e realizam adição, subtração, multiplicação, divisão e exponenciação, respectivamentes, como no exemplo ao lado.



Expressões

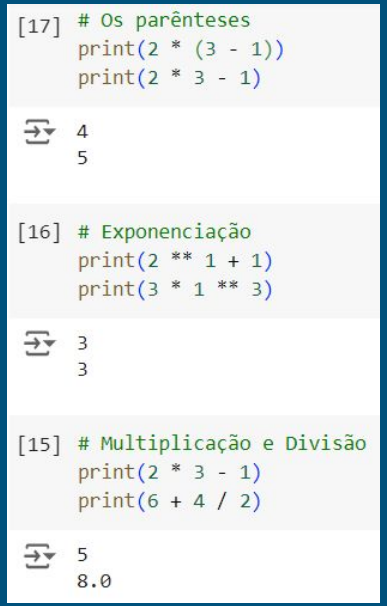
- Uma expressão é uma combinação de valores, variáveis e operadores. Um valor por si só é considerado uma expressão, e também uma variável, então as seguintes são todas expressões legais (assumindo que a variável x recebeu um valor):
- Mas em um script, uma expressão por si só não faz nada! Esta é uma fonte comum de confusão para iniciantes.

```python
# Valores
valor1 = 10
valor2 = 20

# Variáveis
x = valor1
y = valor2

# Operadores
soma = x + y
subtracao = x - y
produto = x * y
divisao = y / x
modulo = y % x

# Expressão completa
# Combinação de valores, variáveis e operadores
expressao = (soma * produto) / (divisao + modulo)

# Imprimindo os resultados
print("Valor de x:", x)
print("Valor de y:", y)
print("Soma:", soma)
print("Produto:", produto)
print("Divisão:", divisao)
print("Módulo:", modulo)
print("Resultado da expressão:", expressao)
```

```
Valor de x: 10
Valor de y: 20
Soma: 30
Produto: 200
Divisão: 2.0
Módulo: 0
Resultado da expressão: 3000.0
```

Ordem das operações

- Quando mais de um operador aparece em uma expressão, a ordem de avaliação depende das regras de precedência . Para operadores matemáticos, Python segue a convenção matemática. O acrônimo PEMDAS é uma maneira útil de lembrar das regras.
  - P - Os parênteses têm prioridades
  - E - Seguida da exponenciação
  - MD - Depois vem a Multiplicação e Divisão tem a mesma precedência
  - AS - Por fim a Adição e Subtração tem a mesma presidência.



Vamos praticar!

Link

Vamos exercitar!

Link