sãojudas universidade

Machine Learning com Python Básico

Prof. Marcelo Duduchi

1

Machine Learning com Python

EMENTA:

Aprendizagem, Agrupamento e Classificação, Phyton, Numpy, Pandas, MatPlotLib, Scikit-Learn para Machine Learning, Agrupamentos e Classificação, Algoritmos K-means, Árvores de Decisão, KNN, SVM e Naive Bayes

OBJETIVO GERAL:

Desenvolver competências relacionadas à aplicação de algoritmos básicos de Machine Learning para agrupamento e classificação em Python.

OBJETIVOS ESPECÍFICOS:

- Desenvolver conceitos sobre aprendizagem supervisionada e não supervisionada
- Desenvolver competências no uso de Python básico para Machine Learning
- Desenvolver competências no uso de bibliotecas básicas para Machine Learning
- Desenvolver competências no uso de algoritmo para agrupamento
- Desenvolver competências no uso de algoritmos para classificação

2

Machine Learning com Python

CONTEÚDOS PROGRAMÁTICOS:

- Aprendizagem supervisionada e não supervisionada
- Agrupamento e classificação
- Python básico para Machine Leanrning
- Bibliotecas Numpy, Pandas, MatPlotLib, Seaborn e Scikit-Learn
- Aplicação do algoritmo K-means
- Aplicação de algoritmos de Árvore de Decisão
- Aplicação do algoritmo KNN
- Aplicação do algoritmo SVM
- Aplicação do algoritmo Naive Bayes

3

Machine Learning com Python

CRONOGRAMA DE AULA:

- 1ª aula (4 horas): Conceitos básicos de IA e aprendizagem de máquina, uso do ambiente Colab e recursos de Python para Machine Learning.
- 2ª aula (4 horas): Bibliotecas de apoio a Machine Learning para computação científica e a recuperação, armazenamento, tratamento, armazenamento e visualização de dados.
- 3ª aula (4 horas): Agrupamento e classificação: uso dos algoritmos K-means e Árvore de Decisão.
- 4ª aula (4 horas): Algoritmos para classificação: uso dos algoritmos KNN, SVM e Naive Bayes.
- 5ª aula (4 horas): Algoritmos para classificação: uso prático dos algoritmos e comitês.

4

Bibliografia sobre IA

RUSSEL, S.; NORVIG, P. Inteligência Artificial. 3ed. Rio de Janeiro: Elsevier, 2013
https://integrada.minhabiblioteca.com.br/#/books/9788521215479/cfi/4!/4/4@0.00:52.7
FACIT, K.; LORENA, A. C.; GAMA, J.; ALMEIDA, T. A.; A. C. P. L. F., CARVALHO. Inteligência Artificial: Uma abordagem de aprendizagem de máquina. 2ed. Rio de Janeiro: LTC, 2021
https://integrada.minhabiblioteca.com.br/#/books/9788521637509/cfi/6/2!/4/2/2@0:66.9
LUGER, George F. Inteligência artificial. 6ed. São Paulo: Pearson, 2013
https://plataforma.bvirtual.com.br/Acervo/Publicacao/180430
COPPIN, Ben. Inteligência artificial. Rio de Janeiro: LTC. 2013
https://integrada.minhabiblioteca.com.br/books/978-85-216-2936-8
SIMÕES, M. G.; SHAW, I. S. Controle e modelagem fuzzy. São Paulo: Blucher, 2007
https://integrada.minhabiblioteca.com.br/#/books/9788521215479/cfi/4!/4/4@0.00:52.7

5

Recursos

BANIN, S. L. Python 3: Conceitos e Aplicações. São Paulo: Érica, 2018
https://integrada.minhabiblioteca.com.br/#/books/9788521215479/cfi/4!/4/4@0.00:52.7
Google Colaboratory (ou apenas Colab) – Ambiente on-line de Notebooks Jupyter Python para Ciência de Dados
https://colab.research.google.com
Biblioteca Pandas – Ferramenta para Ciência de Dados
https://pandas.pydata.org/
Biblioteca NumPy – Ferramenta de Computação Científica
https://numpy.org/
Biblioteca MatPlotLib – Ferramenta para visualização de dados
https://matplotlib.org/
Biblioteca Seaborn – Ferramenta para estatística e visualização de dados
https://seaborn.pydata.org/
Biblioteca scikit-learn – Ferramenta para Análise de Dados Preditiva
https://scikit-learn.org/

6

Machine Learning com Python
Aula 01

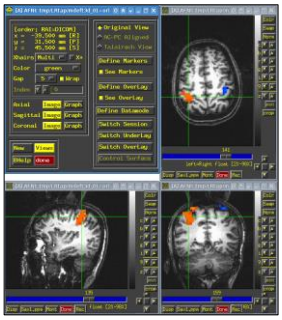
7

Qual é a visão de vocês sobre o que é Inteligência Artificial?

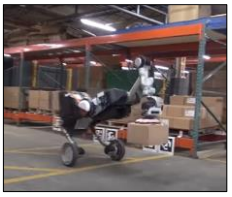
8

Inteligência Artificial








9

O que é inteligência artificial?

10

IA não é coisa de hoje...

Trailer do filme "Metropolis" no século XXI, de Fritz Lang (1927)
https://youtu.be/gdtZv3XROnc

11

IA não é coisa de hoje...

Trailer do filme "2001 uma Odisséia no Espaço", de Stanley Kubrick (1969)
https://youtu.be/GYs2hwsn96s

12

“Inteligência Artificial”

O termo “Artificial Intelligence” foi cunhado por John McCarthy 1 em 1956 durante o seminário de Darthmouth, onde também participaram Marvin Minsky 2, Claude Shannon 3, Allen Newell 4, Herbert Simon 5, etc...


3


1


5

2


4


13

“Inteligência Artificial”

Lá John McCarthy 1, Marvin Minsky 2, Claude Shannon 3, Allen Newell 4 e Herbert Simon 5, afirmaram que: “Cada aspecto do aprendizado ou outra forma de inteligência pode ser descrita de forma tão precisa que uma máquina pode ser usada para simular isso"


3


1


5

2


4


14

História

- Em meados de 1956, foi criado o campo de pesquisa em inteligência artificial.
- Inicialmente nomeado como: "A ciência e engenharia de produzir máquinas inteligentes".
- Um dos maiores nome da computação foi o pioneiro na área de inteligência artificial
- Em 1950 Alan Turing propõe uma definição de sistema inteligente, que ficou conhecida como "Teste de Turing".

Artigo científico em que Turing explica seu teste:
TURING, A.M. *Mind*, Volume LIX, Issue 236, October 1950, Pages 433–460.
https://doi.org/10.1093/mind/LIX.236.433
https://academic.oup.com/mind/article/LIX/236/433/986238

15

IA não é coisa de hoje

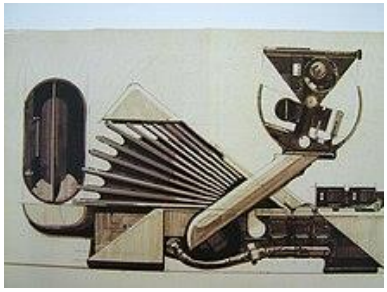
Cenas do filme "Blade Runner" 2019, de Ridley Scott (1982)
https://youtu.be/ajeR8jn_m_s

16

História

Teste Voight-Kampff:

- Teste científico e psicológico fictício citado no livro de ficção científica “Do Androids Dream of Electric Sheep?” d
e Philip K. Dick
- O teste consiste em observar o aumento involuntário da pupila e sua dilatação. Combinando as respostas obtidas com o teste das pupilas, o interrogador obtém a resposta.
- O interrogador faz séries de perguntas.
- No filme "Blade Runner" é dito serem necessárias 30 perguntas para se verificar se é Replicante.
- É uma espécie de Teste de Turing...



17

História

Cenas do filme “Blade Runner”, de Ridley Scott (1982)

https://youtu.be/Umc9ezAyJv0

18

Discussões éticas sobre IA

Cenas do filme "Blade Runner 2049", de Ridley Scott (2017)

https://youtu.be/86XtZMgFzil

19

Discussões éticas sobre IA

Cenas do filme "Eu, robô" 2035, baseado em contos de Isaac Asimov (2004)

https://youtu.be/w-7G0EjyDYQ

20

Evolução da IA

- Agindo humanamente (anos 50-70): Teste de Turing
- Pensando humanamente (anos 50-60): simulação cognitiva com sistemas especialistas *(Simon & Newell)*
- Pensando idealmente (anos 60-70): A escola logicista com formalismos de representação do conhecimento *(McCarthy)*
- Agindo idealmente (anos 80 em diante): Agente inteligente com abordagem abrangente e unificadora *(Newell, Minsky, Russel & Norvig)*

21

21

Stuart Russell & Peter Norvig (2013)

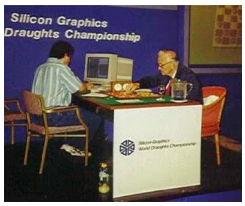
- A Inteligência Artificial tenta não apenas **compreender**, mas também **construir entidades inteligentes**.
- Em linhas gerais os autores ao definirem Inteligência Artificial consideram **duas dimensões:**

	Pensamento ou raciocínio	Comportamento
Fidelidade ao desempenho humano	**Pensando como Humano**	**Agindo como humano**
Racionalidade com ideal da inteligência	**Pensando racionalmente**	**Agindo racionalmente**

22

Evolução da IA

Marion Tinsley x Chinook (1992 -1994)
World Checkers Man-Machine Championship Match



- histórico confronto entre humanos e computadores de 1992.

- Tinsley: imbatível campeão mundial de Damas
- Chinook: programa projetado para jogar pela Universidade de Alberta

- Federação Americana de Damas em 1990 não permitiu a participação da máquina no campeonato mundial.

-Tinsley renunciou ao título de campeão e aceitou jogar contra Chinook.

- Tinsley venceu por 4-2, com 33 empates.
- As 2 derrotas de Tinsley foram apenas as suas 6a e 7a derrotas desde 1950.

Equipe de desenvolvimento do Chinook

- Em uma das partidas, depois que o computador tinha jogado seu 10º movimento, Tinsley disse: "Você vai se arrepender disso".
- Chinook foi derrotado 26 movimentos mais tarde.
- Tinsley tinha planejado 64 movimentos à frente para encontrar como vencer.

- Revanche em 1994: Após seis empates em 6 partidas, Tinsley, já com a saúde deveras debilitada, não pôde mais prosseguir com a competição.

23

Evolução da IA

Garry Kasparov x Deep Blue (1996 – 1997)
Desafio de Xadrez

- 2 matches de 6 partidas

- Garry Kasparov: campeão mundial de xadrez
- Deep Blue::computador da IBM

- 1o match em 1996 Kasparov venceu por 4-2.
- Deep Blue analisava 100 milhões de jogadas

- 2o match em 1997 Deep Blue vence por 3½–2½
- Deep Blue analisava 250 milhões de jogadas

Evento chamado de:

"o mais espetacular evento de xadrez na história".

24

Evolução da IA

Watson x Ken Jennings e Brad Rutter (2011)
Jeopardy

- Programa de perguntas e respostas.
- No ar desde 1964.

- Ken Jennings e Brad Rutter: maiores campeões do programa
- Watson: 100 servidores IBM Power 750 com 15 terabytes de RAM

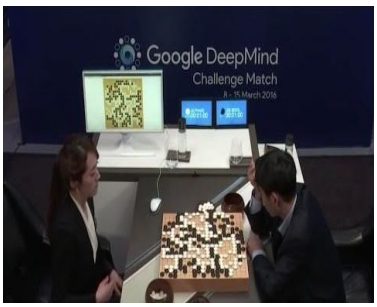
- Disputa de conhecimentos gerais, em linguagem humana.
- Jogo de perguntas e respostas
- Participantes recebem dicas em forma de respostas
- Participantes precisam responder na forma de perguntas.
- A maioria das respostas começa com "what is…".

- Participantes escolhem uma das categorias
- São categorias ou bem específicas ou bem genéricas.
- podem selecionar as de valor e dificuldade que preferirem.

O Watson venceu!

25

Evolução da IA



AlphaGo x Lee Sedol (2016)
Google DeepMind Challenge Match

- Disputa de 5 jogos no milenar jogo de tabuleiro Go
- Jogo de alta complexidade
- Posicionamento de pedras de cores opostas
- Objetivo é dominar o tabuleiro

- Software AlphaGo (DeepMind Tech em Supercomputador da Google).
- Lee Sedol: Sul coreano 18 vezes campeão mundial.

Vários especialistas acreditavam que os programas de inteligência artificial demoraria décadas até vencer o ser humano.

Para que fosse capaz de vencer um campeão mundial de Go, algoritmos de aprendizado com redes neurais profundas foram treinados por meio de base de dados com milhões de jogadas já realizadas por seres humanos.

- O resultado final foi 4-1 para o AlphaGo.
- O jogo Go era a última barreira a ser quebrada em jogos de tabuleiro.

26

Evolução da IA

OpenIA x Danil “Dendi” Ishutin (2017)

Em 2017 um bot que usa IA venceu os melhores jogadores de Dota 2 em partidas individuais.

A OpenAI, cofundada por Elon Musk, desenvolveu um mecanismo de inteligência artificial que desafiou o jogador profissional Danil “Dendi” Ishutin no The International, um dos maiores eventos globais de e-sports.

Dendi foi derrotado duas vezes, e desistiu de futuras partidas com o bot.

Em 2021, a IA também venceu:

- Syed “Suma1L” Hassan, melhor jogador mundial de partidas 1v1.

- Artour “Arteezy” Babaev, melhor jogador mundial de Dota 2.

27

Evolução da IA

- A evolução da Inteligência Artificial depende diretamente da evolução tecnológica (processadores, memória, armazenamento, comunicação e novas tecnologias)
- A evolução tecnológica permitiu que a Inteligência Artificial atuasse em diversas áreas, como por exemplo:
  - Área médica
  - Automação Residencial
  - Segurança
  - Setor Agrícola
  - Setor Financeiro
  - Setor Comercial
  - Atendimento

28

Evolução da IA

- O que possibilita essa "invasão" da IA em todas as áreas?
- Evolução tecnológica
- Mais pessoas têm acesso a tecnologia.
  - Tecnologia se torna mais barata.
  - As empresas tendem a ter mais clientes.
  - Há mais concorrência.
- Há cada vez mais dados no mundo.
- Novos modelos de IA

29

Evolução da IA

- Até anos 1970 IA era vista como uma área teórica com aplicações em pequenos problemas curiosos
- A partir da década de 1970 IA começou-se a pensar em IA para a solução de problemas reais
- Muitas vezes esses problemas eram tratados na computação por aquisição de conhecimento de especialistas de um domínio
- A aquisição de conhecimento era feita modelando informações que esses especialistas traziam por meio de entrevistas
- Com crescente complexidade de problemas e volume de dados surgiu a necessidade de ferramentas mais autônomas
- Foram criadas técnicas capazes de criar por si próprias hipóteses a partir de experiências passadas (Aprendizagem de Máquina)

30

Aplicações

31

Aprendizagem de Máquina

Hoje falaremos sobre:

- Aprendizagem Supervisionada
- Aprendizagem Não Supervisionada
- Aprendizagem por Reforço

32

Aprendizagem de Máquina

- Vimos em aulas anteriores o conceito de **cognição** que está relacionado à **aquisição de conhecimento**;
- A partir de agora estaremos trabalhando o conceito de como podemos fazer **algoritmos** que trabalhem a **aquisição de conhecimento** que chamamos de Aprendizagem de Máquina (**Machine Learning**);
- Para isso precisaremos antes compreender quais os **tipos de algoritmos de aprendizagem** que podemos usar para realizarmos esta aquisição de conhecimento...

33

Aprendizagem de Máquina

- Estudaremos **três categorias** de aprendizagem:
  - **Aprendizagem supervisionada;**
  - **Aprendizagem não supervisionada;**
  - **Aprendizagem por reforço;**
- Apesar de usarem **estratégias diferentes** todas as três categorias trabalham a ideia de que:
  - O sistema aprende a **inferir com base em experiências passadas**;

34

Aprendizagem de Máquina

- Vamos começar com um exemplo:

- Vou apresentar a vocês alguns **equipamentos** para **aquecer ambientes** em tempo de frio.
- Ao apresentar o **objeto** vou falar o **tipo** que ele é.
- Depois de um **período de treino** vou para o **teste** pedindo para **vocês me indicarem** de que **tipo** que ele é, ok?

35

Aprendizagem de Máquina

- Fase de treino...

36

Aprendizagem de Máquina

- Salamandra

37

Aprendizagem de Máquina

- Lareira

38

Aprendizagem de Máquina

- Salamandra

39

Aprendizagem de Máquina

- Salamandra

40

Aprendizagem de Máquina

- Lareira

41

Aprendizagem de Máquina

- Lareira

42

Aprendizagem de Máquina

- Salamandra

43

Aprendizagem de Máquina

- Lareira

44

Aprendizagem de Máquina

- Fase de teste...

45

Aprendizagem de Máquina

- O que é isso?

46

Aprendizagem de Máquina

- O que é isso?
- Salamandra!

47

Aprendizagem de Máquina

- O que é isso?

48

Aprendizagem de Máquina

- O que é isso?
- Lareira!

49

Aprendizagem de Máquina

- O que é isso?

50

Aprendizagem de Máquina

- O que é isso?
- Salamandra!

51

Aprendizagem de Máquina

- O que é isso?

52

Aprendizagem

- O que é isso?
- Salamandra!

53

Aprendizagem

- O que é isso?

54

Aprendizagem de Máquina

- O que é isso?
- Lareira!

55

Aprendizagem Supervisionada

- O que ocorreu no exemplo:
  - Aprendizagem Supervisionada!
- Na aprendizagem supervisionada:
  - Informamos o que é cada entrada (com a resposta)
  - O sistema aprende as características das entradas de acordo com a resposta desejada.
- Dentro da Aprendizagem Supervisionada temos alguns tipos de algoritmos:
  - algoritmo de classificação
  - algoritmo de regressão

56

Aprendizagem Supervisionada

- Classificação
  - O exemplo que trabalhamos é um típico exemplo de classificação.
- O intuito era identificar se o objeto é uma lareira ou uma salamandra.
- Acredito que parte de vocês não sabia o que era uma salamandra.
- Agora conseguem diferenciar:
  - Salamandra
  - Lareira

57

Aprendizagem Supervisionada

- Regressão
  - Na regressão teremos um resultado numérico ao invés de classificatório.
- Um determinado sistema realiza o cálculo de valor de um veículo de acordo com determinadas informações (tipo, marca, ano, motor, kms rodados, etc.)
- De acordo com uma base já existente de veículos com essas informações e o valor de venda de cada um, aprendemos a “encaixar” o novo veículo e precificá-lo.

58

Aprendizagem Supervisionada

- Neste exemplo dos veículos também temos um Aprendizado Supervisionado
- Tentamos prever uma variável (valor), e temos uma base que usamos para inferir este conhecimento
- Temos neste exemplos dois tipos de variáveis:
  - Dependentes: Variável objeto
  - Independentes: Variáveis observadas
- A Aprendizagem supervisionada cria modelos preditivos

59

Aprendizagem Supervisionada

- Para resolver este tipo de Aprendizagem Supervisionada, há algumas técnicas, como:
  - Árvores de decisão
  - SVM (Support Vector Machine)
  - KNN (K Nearest Neighbor)
  - Naive Bayes
  - Regressão linear
  - Regressão logística
  - Redes neurais artificiais

60

Aprendizagem de Máquina

- Vamos agora a outro exemplo:

- Vou apresentar a vocês algumas flores.
- Ao apresentar as flores não vou falar o tipo que cada uma é, mas quero que assimilem a característica de cada uma e tentem agrupá-las em tipos (3 grupos).
- Depois de um período de treino vou pedir para vocês me indicarem de que tipo que a flor apresentada é, ok?

61

Aprendizagem de Máquina

- Fase de treino...

62

Aprendizagem de Máquina

- Observe a Flor 1

63

Aprendizagem de Máquina

- Observe a Flor 2

64

Aprendizagem de Máquina

- Observe a Flor 3

65

Aprendizagem de Máquina

- Observe a Flor 4

66

Aprendizagem de Máquina

- Observe a Flor 5

67

Aprendizagem de Máquina

- Observe a Flor 6

68

Aprendizagem de Máquina

- Observe a Flor 7

69

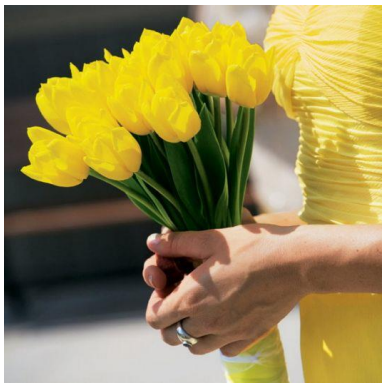
Aprendizagem de Máquina

- Observe a Flor 8

70

Aprendizagem de Máquina

- Observe a Flor 9



71

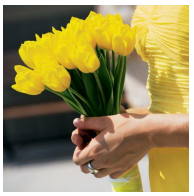
Aprendizagem de Máquina

- Como vocês agruparam em três grupos?







72

Aprendizagem de Máquina

- Fase de teste...

73

Aprendizagem de Máquina

- Qual o grupo?

74

Aprendizagem de Máquina

- Qual o grupo?

75

Aprendizagem de Máquina

- Qual o grupo?

76

Aprendizagem de Máquina

- Qual o grupo?

77

Aprendizagem de Máquina

- Qual o grupo?

78

Aprendizagem de Máquina

- Qual o grupo?

79

Aprendizagem de Máquina

- Qual o grupo?

80

Aprendizagem Não Supervisionada

- Acabamos de proceder a uma aprendizagem não supervisionada.
- O que muda na aprendizagem não supervisionada?
  - Não há rótulos
  - Não informamos os nomes que damos às flores
  - No máximo informamos em quantos grupos queremos que se faça a divisão
  -

81

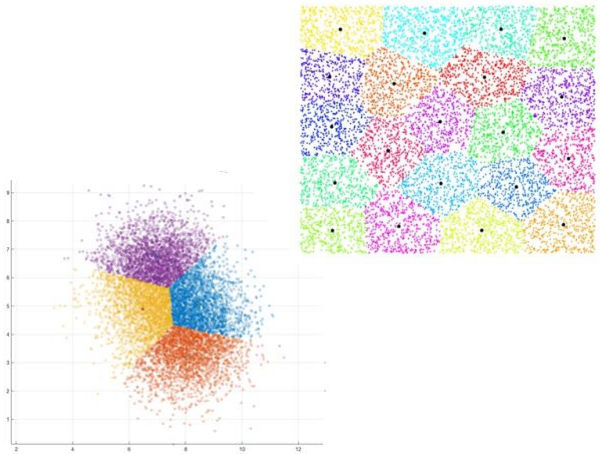
Aprendizagem Não Supervisionada

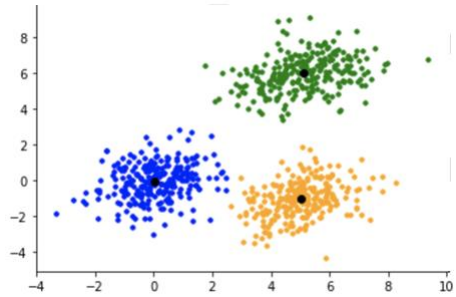
- Vamos para mais um exemplo:
- Imagine dados de uma Universidade.
- Essa Universidade quer separar seus alunos em algumas categorias mas não sabe como fazer.
- Ao usar a Aprendizagem Não Supervisionada, serão entradas as informações, sendo que não se sabe quais serão as classes criadas.
- Na verdade a Aprendizagem Não Supervisionada cria o que chamamos modelos descritivos.

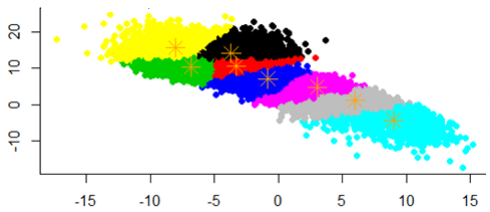
82

Aprendizagem Não Supervisionada

- Exemplos de resultados:







83

Aprendizagem Não Supervisionada

- Além do exemplo de agrupamento comentado, o aprendizado não supervisionado também pode ser utilizado para:
  - Sistemas de recomendação
  - Detecção de anomalias (doenças, fraudes, ..)
- Técnicas para Aprendizagem Não Supervisionada:
  - K-means
  - Redes Neurais
  - SVM (Support Vector Machine)

84

Aprendizagem Não Supervisionada

- Não ter resultados de cada item, não é um problema?
  - Sim e não.
- Não ter um resultado esperado para cada item acarreta:
  - Complicação na avaliação

- Muitas vezes é necessário trabalhar com resultados que muitas vezes podem não ser compreendido pelos humanos.

85

Aprendizagem por Reforço

- Esta categoria de aprendizagem ocorre a partir do oferecimento de recompensa.
- Exemplo:
  - Quando queremos ensinar um cachorro a fazer uma determinada ação pode-se aplicar a aprendizado por reforço.
  - Ação desejada: o cachorro deitar.
  - Cada vez que o cachorro deitar recebe recompensa.
  - Desta forma ele aprende o que ele deve fazer.

86

Aprendizagem por Reforço

- A aprendizagem por reforço é uma maneira de ensinar o que se deve realizar dada uma determinada situação/comando.
- Como é feito isso?
- Vinculando recompensas e/ou punições as ações
- Essas recompensas e punições tem que estar junto com a prioridade de cada ação/objetivo

87

Aprendizagem por Reforço

- Essa ideia vem do conceito de “aprendizagem por reforço” da psicologia, que na verdade é conhecido como behaviorismo
- Skinner um dos principais psicólogos behavioristas:
- Construiu um experimento denominado a caixa de Skinner
- realizou um experimento onde aplicou a ideia de recompensas e punições para treinar pombos para conduzir mísseis na Segunda Guerra Mundial.

88

Aprendizagem por Reforço

Exemplo:

The Big Bang Theory

Reforço e Punição

https://www.youtube.com/watch?v=C1i0Os8WEG4

89

Aprendizagem por Reforço

Exemplo:

Algoritmo Genético

Aprendendo a andar

https://www.youtube.com/watch?v=uwz8JzrEwWY

90

Python para ML

- Python é uma linguagem de propósito geral usada para as mais diversas aplicações.
- Atualmente Python é uma das linguagens mais utilizadas no mundo dada a sua diversidade de uso e o número enorme de bibliotecas disponíveis para as mais diversas áreas.
- Uma das áreas em que Python é muito utilizado é a Inteligência Artificial (IA) principalmente no contexto da Aprendizagem de Máquina e Ciência de Dados.
- Nosso objetivo aqui nesse e no próximo roteiro não será destrinchar todos os conhecimentos sobre Python, mas sim apresentar os conceitos importantes de Python como ferramental de IA.

91

Google Colab

- Como foi dito usaremos o Google Colab como ambiente de trabalho
- O Google Colab é um serviço de nuvem gratuito hospedado pelo Google para incentivar a pesquisa de Inteligência Artificial
- Permite que você escreva e execute Python em seu navegador sem necessidade alguma de configuração e com acesso gratuito à aceleração de processamento e recursos de colaboração.
- Além disso, tem pré-instalado uma série de bibliotecas de nosso interesse.
- Para entrar no Google Colab acesse o link a seguir pelo browse:
- https://colab.research.google.com/

92

Google Colab

- Ao acionar o link https://colab.research.google.com/ você verá a tela a seguir:

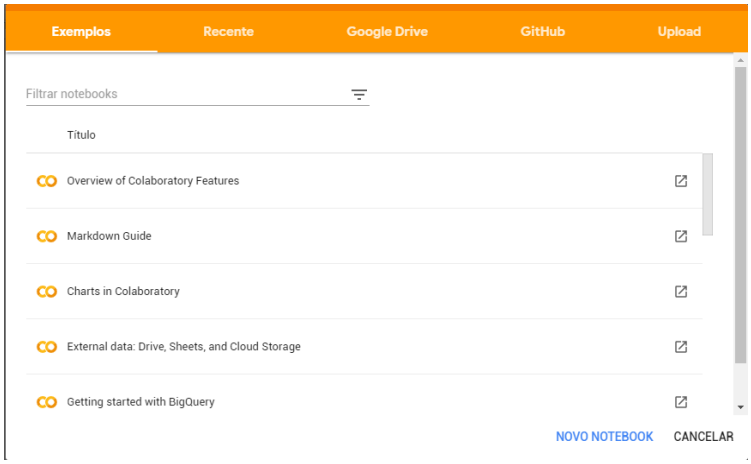
a) Exemplos: Exemplos sobre recursos;

b) Recente: Acesso aos Notebooks mais recentemente;

c) Google Drive: Acesso direto ao Notebooks do seu Google Drive;

d) GitHub: Onde você pode autentificar sua conta do GitHub para trazer Notebooks armazenados lá;

e) Upload: Permite fazer uploads de arquivos Python armazenados no seu computador.



93

Google Colab

- Ao acionar o link https://colab.research.google.com/ você verá a tela a seguir:

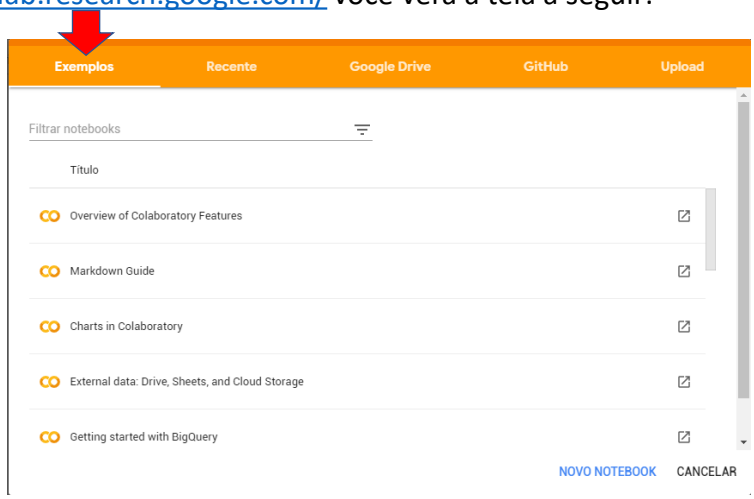
a) Exemplos: Exemplos sobre recursos;

b) Recente: Acesso aos Notebooks mais recentemente;

c) Google Drive: Acesso direto ao Notebooks do seu Google Drive;

d) GitHub: Onde você pode autentificar sua conta do GitHub para trazer Notebooks armazenados lá;

e) Upload: Permite fazer uploads de arquivos Python armazenados no seu computador.



94

Google Colab

- Ao acionar o link https://colab.research.google.com/ você verá a tela a seguir:

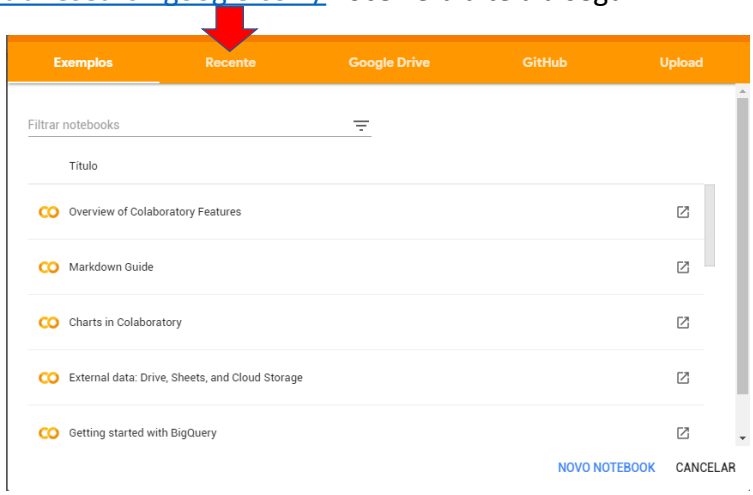
a) Exemplos: Exemplos sobre recursos;

b) Recente: Acesso aos Notebooks mais recentemente;

c) Google Drive: Acesso direto ao Notebooks do seu Google Drive;

d) GitHub: Onde você pode autentificar sua conta do GitHub para trazer Notebooks armazenados lá;

e) Upload: Permite fazer uploads de arquivos Python armazenados no seu computador.



95

Google Colab

- Ao acionar o link https://colab.research.google.com/ você verá a tela a seguir:

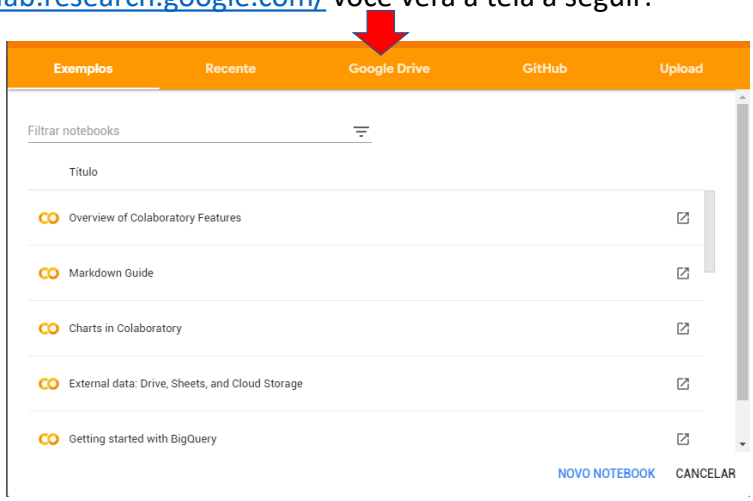
a) Exemplos: Exemplos sobre recursos;

b) Recente: Acesso aos Notebooks mais recentemente;

c) Google Drive: Acesso direto ao Notebooks do seu Google Drive;

d) GitHub: Onde você pode autentificar sua conta do GitHub para trazer Notebooks armazenados lá;

e) Upload: Permite fazer uploads de arquivos Python armazenados no seu computador.



96

Google Colab

- Ao acionar o link https://colab.research.google.com/ você verá a tela a seguir:

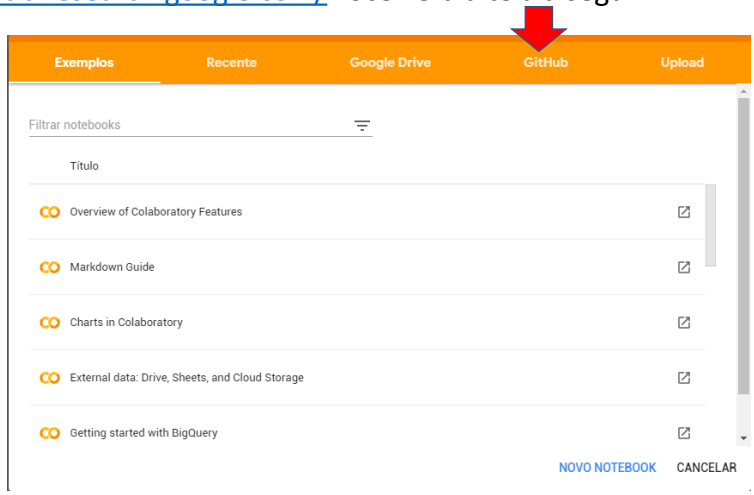
a) Exemplos: Exemplos sobre recursos;

b) Recente: Acesso aos Notebooks mais recentemente;

c) Google Drive: Acesso direto ao Notebooks do seu Google Drive;

d) GitHub: Onde você pode autentificar sua conta do GitHub para trazer Notebooks armazenados lá;

e) Upload: Permite fazer uploads de arquivos Python armazenados no seu computador.



97

Google Colab

- Ao acionar o link https://colab.research.google.com/ você verá a tela a seguir:

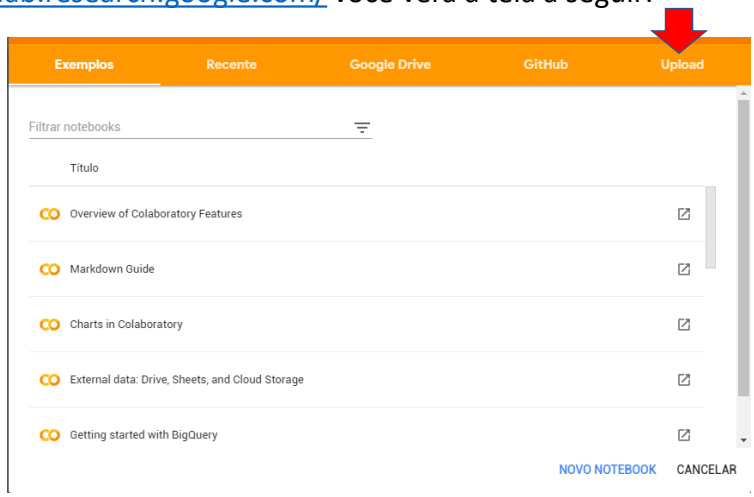
a) Exemplos: Exemplos sobre recursos;

b) Recente: Acesso aos Notebooks mais recentemente;

c) Google Drive: Acesso direto ao Notebooks do seu Google Drive;

d) GitHub: Onde você pode autentificar sua conta do GitHub para trazer Notebooks armazenados lá;

e) Upload: Permite fazer uploads de arquivos Python armazenados no seu computador.



98

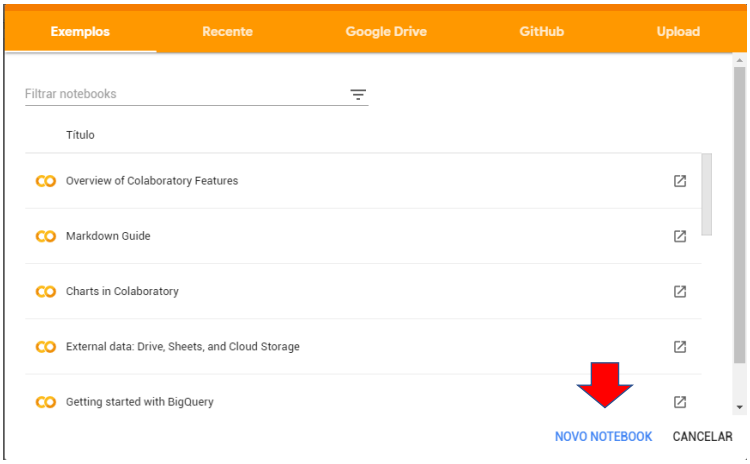
Google Colab

- Criando um novo Notebook:

Os Notebooks Colab são Notebooks Jupyter armazenados no Google Drive que permitem combinar código executável e texto em um único documento

Eles são armazenados em sua conta do Google Drive.

Para criar um novo Notebook basta selecionar a opção "NOVO NOTEBOOK" da parte inferior da tela do Google Colab.



99

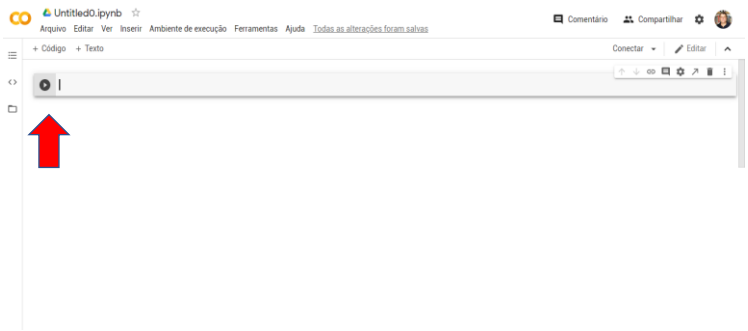
Google Colab

- Incluindo códigos e Textos:

Ao criar um novo Notebook você já pode digitar seus códigos em ambiente responsivo

Pode digitar um código e ver o resultado dele clicando no botão a esquerda do código digitado

Posicionando o mouse abaixo da célula existente aparecerão opções para adicionar novas células de código ou ainda células de texto



100

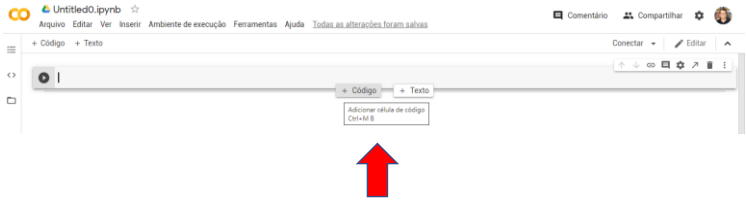
Google Colab

- Incluindo códigos e Textos:

Ao criar um novo Notebook você já pode digitar seus códigos em ambiente responsivo

Pode digitar um código e ver o resultado dele clicando no botão a esquerda do código digitado

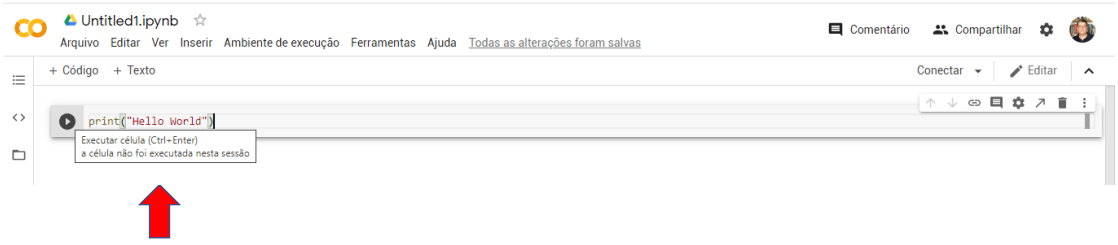
Posicionando o mouse abaixo da célula existente aparecerão opções para adicionar novas células de código ou ainda células de texto



101

Google Colab

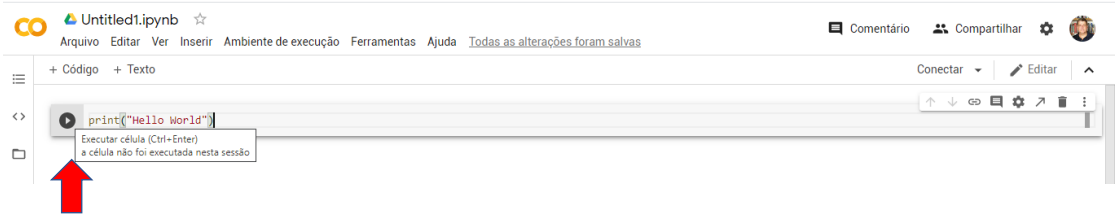
- Para executar seu primeiro programa em Python, o famoso “Hello World” basta digitar o comando print (‘Hello World’) e executar clicando na seta à direita do código



102

Google Colab

- Para executar seu primeiro programa em Python, o famoso “Hello World” basta digitar o comando print (‘Hello World’) e executar clicando na seta à direita:

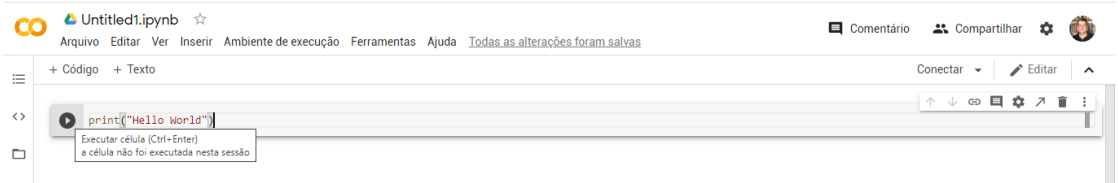


- O resultado da execução aparecerá logo abaixo:

103

Google Colab

- Para executar seu primeiro programa em Python, o famoso “Hello World” basta digitar o comando print (‘Hello World’) e executar clicando na seta à direita:

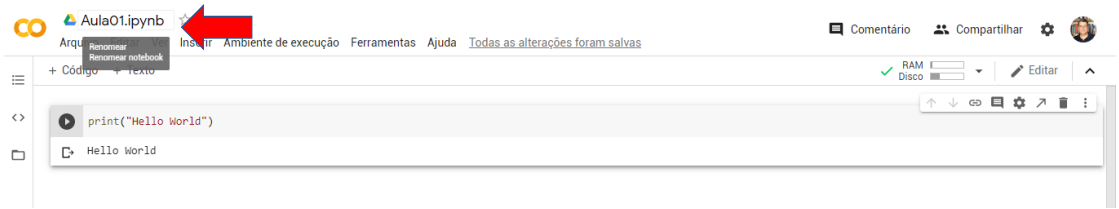


- O resultado da execução aparecerá logo abaixo:

104

Google Colab

- Você pode alterar o nome do Notebook diretamente escrevendo por sobre o nome atual ou usar o menu de opções “Arquivo” que permite a manipulação de arquivos.



- Além do menu “Arquivo” ainda existem os menus “Editar” com opções voltadas para edição dos textos, “Ver” para verificar detalhes do arquivo que está sendo editado, “Inserir” para incluir células diversas, “Ambiente de Execução” para controles diversos relativos à execução e o menu de Ferramentas e Ajudas.

105

Tipos de Dados

Os tipos de dados básicos são o numérico, lógico e conjunto de caracteres (String).

- Numéricos:

Os valores numéricos podem fazer parte de expressões aritméticas com os operadores

- soma (5 + 7 = 12), subtração (4 – 2 = 2), multiplicação (3 * 2 = 6), divisão (6 / 3 = 2),
- resto da divisão (11 % 3 = 2) e potência (5 ** 2 = 25).

Os valores numéricos também podem fazer parte de expressões de comparação com os operadores:

- “menor que” (a < 7), “menor ou igual” (b <= 6), “maior que” (c > 3), “maior ou igual” (d >= 5),
- “igual” (e == 9) e “diferente” (f != 3).

106

Código de Exemplo

```
x = 3
print(type(x))
# Mostra "<class 'int'>"
print(x)
# Mostra "3"
print(x + 1)
# Adiciona e mostra "4"
print(x - 1)
# Subtrai e mostra "2"
print(x * 2)
# Multiplica e mostra "6"
print(x ** 2)
# Faz a exponenciação e mostra "9"
x += 1
print(x)
# Mostra "4"
x *= 2
print(x)
# Mostra "8"
y = 2.5
print(type(y))
# Mostra "<class 'float'>"
print(y, y + 1, y * 2, y ** 2)
# Mostra "2.5 3.5 5.0 6.25"
```

107

Resultado da Execução

```
<class 'int'>
3
4
2
6
9
4
8
<class 'float'>
2.5 3.5 5.0 6.25
```

108

Tipos de Dados

Os tipos de dados básicos são o numérico, lógico e conjunto de caracteres (String).

- Lógicos:

- O tipo lógico considera apenas os valores True e False. Os valores lógicos podem fazer parte de expressões lógicas com os operadores:
- a) “not” onde “not a”, se a é True, vira False e se é False, vira True.
- b) “and” onde a expressão “(a < 3) and (c <= 7)” só será True se as duas expressões são True.
- c) “or” onde a expressão “(a < 3) or (c <= 7)” só será False se as duas expressões são False.

109

Código de Exemplo

```
t = True
f = False
print(type(t))
# Mostra "<class 'bool'>"
print(t and f)
# Trabalha o operador AND e mostra "False"
print(t or f)
# Trabalha o operador OR e mostra "True"
print(not t)
# Trabalha o operador NOT e mostra "False"
print(t != f)
# Trabalha o operador XOR (ou exclusivo) e mostra "True"
```

110

Resultado da Execução

```
t = True
f = False
print(type(t))
# Mostra "<class 'bool'>"
print(t and f)
# Trabalha o operador AND e mostra "False"
print(t or f)
# Trabalha o operador OR e mostra "True"
print(not t)
# Trabalha o operador NOT e mostra "False"
print(t != f)
# Trabalha o operador XOR (ou exclusivo) e mostra "True"
```

```
<class 'bool'>
False
True
False
True
```

111

Tipos de Dados

Os tipos de dados básicos são o numérico, lógico e conjunto de caracteres (String).

- Strings:

- Strings são conjuntos de caracteres. Nele podemos guardar nomes e frases. Para considerar uma variável como string é necessário que esta receba o conjunto de caracteres entre aspas simples ou aspas duplas.
- Exemplo: nome = ‘Jose da Silva’ ou ainda nome = “Jose da Silva”
- Existem várias funções e operações que podem ser realizadas com strings como por exemplo a concatenação que é a junção entre dois strings a partir do operador soma.
- Exemplo: Com n = ‘Joao’ sn = ‘Souza’ criar nome completo nc = n + sn com nome e sobrenome

112

Código de Exemplo

```
nome = 'Jose da Silva'
print(nome)
n = 'Joao'
sn = 'Souza'
nc = n + ' ' + sn
print(nc)
```

113

Resultado da Execução

```
nome = 'Jose da Silva'
print(nome)
n = 'Joao'
sn = 'Souza'
nc = n + ' ' + sn
print(nc)
```

```
Jose da Silva
Joao Souza
```

114

Tipos de Dados

Existem uma série de funções para manipulações de strings. Segue uma relação delas:

- a) len(): Retorna o tamanho da string.
- Exemplo: Em teste = “Apostila de Python” o uso len(teste) retorna 18
- b) capitalize(): Retorna a string com a primeira letra maiúscula.
- Exemplo: Em a = "python" o uso a.capitalize() gera 'Python'
- c) count(): Informa quantas vezes um caractere (ou uma sequência de caracteres) aparece na string.
- Exemplo: Em b = "Linguagem Python" o uso b.count("n") retorna 2
- d) startswith(): Verifica se uma string inicia com uma determinada sequência.
- Exemplo: Em c = "Python" o uso c.startswith("Py") retorna True

115

Tipos de Dados

Existem uma série de funções para manipulações de strings. Segue uma relação delas:

- e) endswith(): Verifica se uma string termina com uma determinada sequência.
- Exemplo: Em d = "Python" o uso d.endswith("Py") retorna False
- f) isalnum(): Verifica se a string possui algum conteúdo alfanumérico (letra ou número).
- Exemplo: Em e = "!@#$%" o uso e.isalnum() retorna False
- g) isalpha(): Verifica se a string possui apenas conteúdo alfabético.
- Exemplo: Em f = "Python" o uso f.isalpha() retorna True
- h) islower(): Verifica se todas as letras de uma string são minúsculas.
- Exemplo: Em g = "pytHon" o uso g.islower() retorna False

116

Tipos de Dados

Existem uma série de funções para manipulações de strings. Segue uma relação delas:

- i) isupper(): Verifica se todas as letras de uma string são maiúsculas.
- Exemplo: Em h = "# PYTHON 12" o uso h.isupper() retorna True
- j) lower(): Retorna uma cópia da string trocando todas as letras para minúsculo.
- Exemplo: Em i = "#PYTHON 3" o uso i.lower() gera '#python 3'
- k) upper(): Retorna uma cópia da string trocando todas as letras para maiúsculo.
- Exemplo: Em j = "Python" o uso j.upper() gera 'PYTHON'
- l) swapcase(): Inverte o conteúdo da string (Minúsculo / Maiúsculo).
- Exemplo: Em k = "Python" o uso k.swapcase() gera 'pYTHON'

117

Tipos de Dados

Existem uma série de funções para manipulações de strings. Segue uma relação delas:

- m) title(): Converte para maiúsculo todas as primeiras letras de cada palavra da string.
- Exemplo: Em l = "apostila de python" o uso l.title() gera 'Apostila De Python'
- n) split(): Transforma a string em uma lista, utilizando os espaços como referência.
- Exemplo: Em m = "cana de açúcar" o uso m.split() gera ['cana', 'de', 'açúcar']
- o) replace(S1, S2): Substitui na string o trecho S1 pelo trecho S2.
- Exemplo: Em n = "Apostila teste" o uso n.replace("teste", "Python") gera 'Apostila Python'
- p) find(): Retorna o índice da primeira ocorrência de um determinado caractere na string. Se o caractere não estiver na string retorna -1.
- Exemplo: Em o = "Python" o uso o.find("h") retorna 3

118

Tipos de Dados

Existem uma série de funções para manipulações de strings. Segue uma relação delas:

- q) ljust(): Ajusta a string para um tamanho mínimo, acrescentando espaços à direita se necessário.
- Exemplo: Em p = " Python" o uso p.ljust(15) gera ' Python '
- r) rjust(): Ajusta a string para um tamanho mínimo, acrescentando espaços à esquerda se necessário.
- Exemplo: Em q = "Python" o uso q.rjust(15) gera ' Python'
- s) center() Ajusta a string para um tamanho mínimo, acrescentando espaços à esquerda e à direita, se necessário.
- Exemplo: Em r = "Python" o uso r.center(10) gera ' Python '
- t) lstrip(): Remove todos os espaços em branco do lado esquerdo da string.
- Exemplo: Em s = " Python " o uso s.lstrip() gera 'Python '

119

Tipos de Dados

Existem uma série de funções para manipulações de strings. Segue uma relação delas:

- u) rstrip(): Remove todos os espaços em branco do lado direito da string.
- Exemplo: Em t = " Python " o uso t.rstrip() gera ' Python'
- v) strip(): Remove todos os espaços em branco da string.
- Exemplo: Em u = " Python " o uso u.strip() gera 'Python'

120

Tipos de Dados

Existem uma série de funções para manipulações de strings. Segue uma relação delas:

- Além dessas funções existe ainda a possibilidade de fatiamento de Strings que permite extrair do string apenas uma parte dos elementos de uma string.
- O fatiamento é uma ferramenta usada para extrair apenas uma parte dos elementos de uma string. O formato usado é:
- NomeDoString [LimiteInferior : LimiteSuperior]
- Ele retorna string com os elementos das posições do limite inferior até o limite superior - 1.
- Para s = "Python" o uso s[1:4] seleciona os elementos das posições 1,2,3 e gera 'yth'
- Para s = "Python" o uso s[2:] seleciona os elementos a partir da posição 2 e gera 'thon'
- Para s = "Python" o uso s[:4] seleciona os elementos até a posição 3 e gera 'Pyth'

121

Estruturas de Dados

As estruturas de dados do tipo “Containers” são listas, dicionários, sets e tuplas.

- **Listas:** Estrutura mutável, ou seja, ela pode ser modificada. A seguir estão algumas funções utilizadas:
- a) len: retorna o tamanho da lista.
- Exemplo: L = [1, 2, 3, 4] e len(L) retorna 4
- b) min: retorna o menor valor da lista.
- Exemplo: L = [10, 40, 30, 20] e min(L) retorna 10
- c) max: retorna o maior valor da lista.
- Exemplo: L = [10, 40, 30, 20] e max(L) retorna 40
- d) sum: retorna soma dos elementos da lista.
- Exemplo: L = [10, 20, 30] e sum(L) retorna 60

122

Estruturas de Dados

As estruturas de dados do tipo “Containers” são listas, dicionários, sets e tuplas.

- **Listas:** Estrutura mutável, ou seja, ela pode ser modificada. A seguir estão algumas funções utilizadas:
- e) append: adiciona um novo valor na no final da lista.
- Exemplo: L = [1, 2, 3] e L.append(100) gera L [1, 2, 3, 100]
- f) extend: insere uma lista no final de outra lista.
- Exemplo: L = [0, 1, 2] e L.extend([3, 4, 5]) gera L [0, 1, 2, 3, 4, 5]
- g) del: remove um elemento da lista, dado seu índice.
- Exemplo: L = [1,2,3,4] e del L[1] gera L [1, 3, 4]
- h) in: verifica se um valor pertence à lista.
- Exemplo: L = [1, 2, 3, 4] e 3 in L retorna True

123

Estruturas de Dados

As estruturas de dados do tipo “Containers” são listas, dicionários, sets e tuplas.

- **Listas:** Estrutura mutável, ou seja, ela pode ser modificada. A seguir estão algumas funções utilizadas:
- i) sort( ): ordena em ordem crescente
- Exemplo: L = [3, 5, 2, 4, 1, 0] e L.sort( ) gera L [0, 1, 2, 3, 4, 5]
- j) reverse( ): inverte os elementos de uma lista.
- Exemplo: L = [0, 1, 2, 3, 4, 5] e L.reverse( ) gera L [5, 4, 3, 2, 1, 0]

124

Estruturas de Dados

É possível também fazermos operações com diversas listas:

- a) Concatenação (+): coloca a 2a lista após a1a. lista formando uma com todos elementos
- Exemplo: se tivermos a = [0,1,2] e b = [3,4,5], então c = a + b gera c=[0, 1, 2, 3, 4, 5]
- b) Repetição (*): cria uma lista com n vezes os elementos da original
- Exemplo: se tivermos L = [1,2] então R = L * 4 gera R= [1, 2, 1, 2, 1, 2, 1, 2]

125

Estruturas de Dados

É possível também fazermos operações com diversas listas:

- c) Fatiamento: O fatiamento de listas é semelhante ao fatiamento de strings.
- Exemplo: se tivermos L = [3 , 'abacate' , 9.7 , [5 , 6 , 3] , "Python" , (3 , 'j')] então
- L[1:4] seleciona os elementos das posições 1,2,3 e gera ['abacate', 9.7, [5, 6, 3]]
- L[2:] seleciona os elementos a partir da posição 2 e gera [9.7, [5, 6, 3], 'Python', (3, 'j')]
- L[:4] seleciona os elementos até a posição 3 e gera [3, 'abacate', 9.7, [5, 6, 3]]

126

Estruturas de Dados

É possível também fazermos operações com diversas listas:

- é possível criar listas com range ( ). A função range() define um intervalo de valores inteiros. Associada a list(), cria uma lista com os valores do intervalo. A função range() pode ter de 1 a 3 parâmetros (compreenderemos melhor funções na próxima aula):
- - range(n) gera um intervalo de 0 a n-1
- - range(i , n) gera um intervalo de i a n-1
- - range(i , n, p) gera um intervalo de i a n-1 com intervalo p entre os números
- Exemplos:
- L1 = list(range(5)) gera [0, 1, 2, 3, 4] L2 = list(range(3,8)) gera [3, 4, 5, 6, 7] L3 = list(range(2,11,3)) gera [2, 5, 8]

127

Estruturas de Dados

As estruturas de dados do tipo “Containers” são listas, dicionários, sets e tuplas.

- **Dicionários:** conjunto de valores, um a um, associados a uma chave de acesso. Um dicionário em Python é declarado dentro de chaves com cada chave e valor separado por “:” e cada conjunto de chave e valor separados por vírgulas. tem para cada valor de produto um nome associado como chave
- Precos = {'lapis': 5.5, 'borracha': 7.0, 'caneta': 6.5}
- Para mostrar todo o dicionário basta indicar o nome do dicionário no print:
- Print(Precos)
- Para acessar um valor específico do dicionário basta usar o nome do dicionário e a chave entre parênteses:
- print("o preco da borracha eh:", Precos ['borracha'])

128

Estruturas de Dados

Algumas funções dos dicionários que podem ser úteis são:

- a) del: exclui um item informando a chave.
- Exemplo: del Precos[’borracha’] retirará ‘borracha’ e deixará Precos com [‘lapis’: 5.5, ‘caneta’: 6.5]
- b) in: verificar se uma chave existe no dicionário.
- Exemplo: ‘caneta’ in Precos retorna True e ‘caderno’ in Precos retorna False
- c) keys ( ) : obtém as chaves de um dicionário.
- Exemplo: Precos.keys ( ) retorna dict_keys([‘lapis’, ‘borracha’, ‘caneta’])
- d) values ( ): obtém os valores de um dicionário.
- Exemplo: Precos.values ( ) retorna dict_values([5.5, 7.0, 6.5])

129

Estruturas de Dados

As estruturas de dados do tipo “Containers” são listas, dicionários, sets e tuplas.

- **Sets:** conjuntos não ordenados de elementos. Para sets usamos chaves.
- animais = {'gato', 'cachorro’}
- a) add: inclui um item no conjunto
- Exemplo: animais.add(‘peixe’)
- b) len: retorna o tamanho do conjunto
- Exemplo: len(animais)
- c) remove: retira um item do conjunto
- Exemplo: animais.remove(‘gato’)
- d) in: para verificar se está no conjunto
- Exemplo: print(‘peixe’ in animais)

130

Estruturas de Dados

As estruturas de dados do tipo “Containers” são listas, dicionários, sets e tuplas.

- **Tuplas:** listas que não podem ser alteradas. Em tuplas usamos parênteses “( )” ao invés de colchetes “[ ]” como em listas.
- tp1 = (1,2,3,4,5,6)
- Uma forma alternativa de fazer a construção de tuplas usando iterações:
- Exemplo: d = {(x, x + 1): x for x in range(10)}
- Cria a tupla: {(0, 1): 0, (1, 2): 1, (2, 3): 2, (3, 4): 3, (4, 5): 4, (5, 6): 5, (6, 7): 6, (7, 8): 7, (8, 9): 8, (9, 10): 9}

131

Funções

O uso de funções permite incluir na análise de dados estruturas de condição e iteração.

A seguir é apresentado o primeiro exemplo de uma função que tem sua definição a partir da instrução “def” com o nome da função logo a seguir e os parâmetros entre parênteses.

Exemplo:

```
def soma(a,b):
    c=a+b
    return c
soma(5,4)
```

132

Entrada de Dados e Formatação

Um dos recursos que as vezes precisamos considerar é o de receber valores de quem está usando a aplicação.

Isto permite que as funções e conjunto de instruções em Python no Colab se tornem mais genéricas sem necessidade de alterá-las a todo o momento para alterar determinados valores.

Para isso é possível utilizar a instrução “input”. A instrução “input” para a execução da célula ou função, solicita ao usuário que digite um determinado valor e após a digitação do valor atribui para uma eventual variável o valor digitado.

Exemplo:

```
nome = input("Digite seu nome:")
```

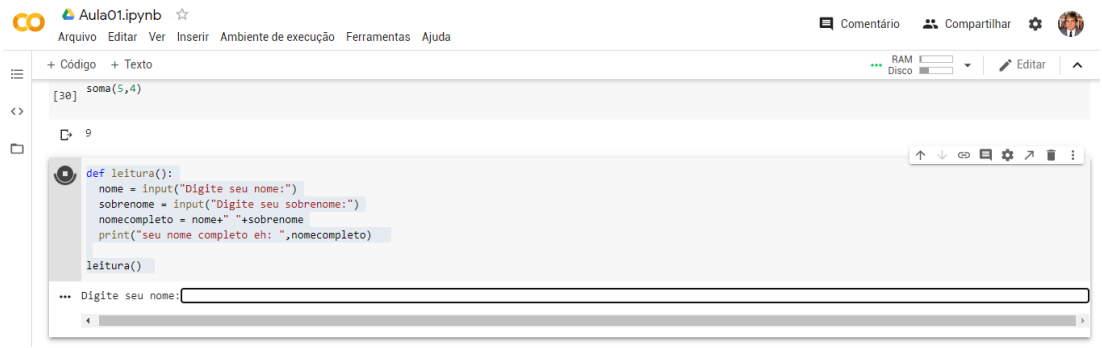
133

Entrada de Dados e Formatação

Veja como a tela de entrada de dados fica nesse exemplo de função de entrada:

Exemplo:

```
nome = input("Digite seu nome:")
```



134

Entrada de Dados e Formatação

Se a entrada de dados necessita ser de um tipo específico é necessário fazer a conversão indicando o tipo para o qual a entrada deve ser transformada. Para isso, utiliza-se o int (string) para converter para o tipo inteiro, ou float (string) para converter para o tipo float. Veja os exemplos:

Exemplos:

```
def potencia(a,b):
    c=a**b
    return c
x = int(input("Digite um valor:"))
y = int(input("Digite o valor de potência:"))
print(potencia(x,y))
```

```
def soma(a,b):
    c=a+b
    return c
x = float(input("Digite um valor:"))
y = float(input("Digite um valor:"))
print(soma(x,y))
```

135

Estruturas de Decisão (Condicional)

As estruturas de decisão permitem alterar o fluxo de execução de um programa, percorrendo um ou outro conjunto de instruções de acordo com o valor (Verdadeiro/Falso) de um teste lógico. Em Python temos as estruturas de decisão “se” (if), “se/senão” (if..else) e “se/senão se/senão” (if..elif..else)

Sintaxes:

```
if <condição>:
    <instruções>
```

```
if <condição1>:
    <instruções1>
else :
    <instruções2>
```

```
if <condição1>:
    <instruções1>
elif <condição2>:
    <instruções2>
 ...
else :
    <instruçõesN>
```

136

Estruturas de Repetição

Laço while

No laço while, o trecho de código da repetição está associado a uma condição. Enquanto a condição tiver valor verdadeiro, o trecho é executado. Quando a condição passa a ter valor falso, a repetição termina.

Sintaxe:

while <condição>:

    <instruções>

137

Código de Exemplo

- Exemplo:

```
senha = "54321"
leitura =" "
while (leitura != senha):
    leitura = input("Digite a senha: ")
    if leitura == senha : print('Acesso liberado ')
    else: print('Senha incorreta. Tente novamente’)
----------------------------------------------------------------------------------
Digite a senha: 12345
Senha incorreta. Tente novamente
Digite a senha: abcde
Senha incorreta. Tente novamente
Digite a senha: 54321
Acesso liberado
```

138

Código de Exemplo

- Exemplo:

```
contador = 0
somador = 0
while contador < 5:
      contador = contador + 1
      valor = float(input('Digite o '+str(contador)+'° v
alor: '))
      somador = somador + valor
print('Soma = ', somador)
----------------------------------------------------------------------------------------------------
Digite o 1° valor: 1
Digite o 2° valor: 2
Digite o 3° valor: 3
Digite o 4° valor: 4
Digite o 5° valor: 5
Soma =  15.0
```

139

Estruturas de Repetição

Laço for

O laço for é a estrutura de repetição mais utilizada em Python. Pode ser utilizado com uma sequência numérica (gerada com o comando range) ou associado a uma lista. O trecho de código da repetição é executado para cada valor da sequência numérica ou da lista.

Sintaxe:

for <variável> in range (<início>, <limite>, <passo>):

    <instruções>

for <variável> in <lista>:

    <instruções>

140

Código de Exemplo

- Exemplo:

- a) Encontrar a soma S = 1+4+7+10+13+16+19

```
S=0
for x in range(1,20,3):
      S = S+x
print('Soma = ',S)
```

```
Soma =  70
```


Código de Exemplo

- Exemplo:

- b) As notas de um aluno estão armazenadas em uma lista. Calcular a média dessas notas.

```
Lista_notas= [3.4,6.6,8,9,10,9.5,8.8,4.3]
soma=0
for nota in Lista_notas:
      soma = soma+nota
média = soma/len(Lista_notas)
print('Média = ', '{:.4f}'.format(média))
```

```
Média =  7.4500
```

Música para terminar a aula

Trilha sonora de Vangelis "Love Theme" (1982)
https://youtu.be/C9KAqhbIZ7o

143