Histórico(1)-Fritz

Foi criada por Guido van Rossum no intuito de criar uma linguagem fácil e intuitiva enquanto que ainda sendo tão poderosa quanto as maiores competidoras

Histórico(2)-Fritz

Guido Van Rossum é Considerado o Ditador Benevolente Vitalício da Linguagem Python, por ser seu criador e também por ser o principal supervisor de desenvolvimento.

COntexto para a criação(1) - Guilherme

Algumas características importantes da linguagem python foi inspirada em outras linguagens como ALGOL 60 , pascal e até ABC linguagem usada pelo grupo CWI. Alguns exemplos são : indentação, tuplas listas e dicionários, a ideia que deveria ter tipos de dados mutáveis e imutáveis, a ideia de não deveria ter limites de caracteres por linha por exemplo, prompt iterativo.
Vendo que a linguagem não poderia aceitar coisas novas a ela. Não podíamos fazer coisas como representar graficamente ou era muito difícil de fazer isso. Outros fatores levaram a ruína do ABC como Letras maiúsculas para designar palavras chave, a criação de um nova terminologia para certos procedimentos(HOW TO ao invés de procedure) e falta de integração com o mundo externo, não podendo criar arquivos e ler arquivos.
Algo entre c que demorava pra programar e é rápido e Shell que era rápido de escrever mais lento e ineficiente para alguns programas. Começou um interpretador python,

Mudanças na comunidade python - Amanda

Mudanças entre as versões - Guilherme

Premissas - Gabriel Miranda

O interpretador Python é facilmente estendido com novas funções e tipos de dados implementados em C ou C ++ (ou outros idiomas que podem ser chamados a partir de C)
Comunidade grande e ativa, o que facilita bastante o processo de desenvolvimento
Facilidade de implementação e leitura.
◦ Python foca na produtividade, sendo muito útil na criação rápida de programas.

Principais Características - Gabriel MIranda

Usuario característico - Gabriel Miranda

◦ Importante dizer que a linguagem é usada por um amplo espectro de usuários dês de programadores iniciantes até profissionais na área cientifica. Guido diz que “não vai compromissar a linguagem com respeito a acessibilidade, pensando sempre nos diferentes tipos de usuários. Aplicações Web,O YouTube foi desenvolvido em Python – Uma das três linguagens predominantes nos servidores do Google (as outras são C++ e Java),O G1 foi construído em Django – Framework mais popular do mundo Python ,Mozilla Firefox Ad-Ons ,Dropbox ,Google App Engine
Aplicações desktop;
Computação gráfica;
Computação Científica;
Desenvolvimento de Games:Civilization IV ,Frets on Fire

Tipos de dados - Fritz

mais importantes a aprender ao fazer a transição para a linguagem de programação Python de outra linguagem é que tudo em Python é um objeto
Uma maneira fácil de visualizar como a determinação de tipo dinâmica funciona é imaginar uma única classe base chamada PyObject a partir da qual todos os outros tipos de objetos em Python são herdados. Neste modelo, qualquer variável criada faz referência a um objeto que foi criado a partir da hierarquia de classe geral. Se você também tiver a classe PyObject , registre o tipo real ou o nome da classe-filha, que é criada e designada à variável, um programa Python pode determinar de forma apropriada as etapas necessárias a serem executadas durante a execução do programa.
Tipos simples
Esses tipos são imutáveis, o que significa que quando um objeto integer é criado, seu valor não pode ser alterado. Em vez disso, um novo objeto de tipo simples é criado e designado à variável.

Tipos primitivos:
◦ Simples: números (int, long, float, complex) e cadeias de caracteres (strings)
◦ Compostos: listas, dicionários, tuplas e conjuntos

Tipos definidos

LIstas,TUplas, Strings e Dicionarios - Guilherme
Uma lista é um conjunto ordenado de valores, onde cada valor é identificado por um índice.
String é um cadeia de caracteres.
◦ strings são imutáveis e listas são mutáveis.
Tuplas é similar a uma lista exceto por ele ser imutável. Sintaticamente, uma tupla é uma lista

de valores separados por vírgulas
Dicionários são seqüências que podem utilizar índices de tipos variados, bastando que estes índices sejam imutáveis

Métodos são simplesmente como funções, com duas diferenças
◦ Métodos são definidos dentro da definição de uma classe para tornar explícita a relação entre a classe e o método.
◦ A sintaxe para a chamada do método é diferente da sintaxe para a chamada de uma função.

O Tipo Complexo
O tipo final, complex, provavelmente não é tão reconhecível para a maioria dos programadores, pois não é um tipo de dados integrado comum em outras linguagens de programação. Para engenheiros e cientistas, números complexos são um conceito familiar. Formalmente, um número complexo tem um componente real e um imaginário, ambos representados por tipos float em Python. Um número imaginário é um múltiplo da raiz quadrada de menos um, que é denotada por i ou j -- dependendo de se você foi treinado como um cientista ou um engenheiro. Em Python, o componente imaginário de um número complexo é indicado por um j:

Classes e Objetos.

Metodos e funcoes

excecoes

Avaliação da linguagem

- Legibilidade (Amanda)
Phyton é uma linguagem bastante legível. Isso se deve muito a herança do ABC. Um exemplo de legibilidade é o fato de a indentação ser usada com delimitador de blocos de código, ou seja, foi extinguida a necessidade de chaves e/outros delimitadores de início e fim de um bloco. Muitas outras linguagens têm padrões de indentação convencionados para melhor a legibilidade, mas em Phyton isso é obrigatório. É importante mencionar que na linguagem existe restrição para a identificação de variáveis, que essa tem palavras reservas e semântica consistente.
Também é importante mencionar o uso de poucos caracteres não alfanuméricos sendo que os que são usados costumam ser bastante usuais como os parênteses, chaves, aspas e dois pontos. Isso garante que o código seja menos intimidador para um programador iniciante, por exemplo, o que é uma das razões de phyton ser usado para ensinar cursos básicos de

programação. Também facilita a manutenção.
As estruturas de controle englobam o tratamento de exceções o que influencia na queda da legibilidade, porém como não são tão extensas isso não se trona um fator decisivo. A interpretação do código também pode ser prejudicada com o fato de não precisar de declaração de variáveis cuja existência depende apenas de uma expressão.

-Capacidade de Escrita (Fritz)
É uma linguagem fácil de usar e tem um conjunto bastante pequeno de tipos primitivos e de estruturas lógicas e de controle, ou seja, é uma linguagem bastante simples. Tem alta capacidade de abstração e é de alto nível tornando essa simplicidade poderosa e expressiva. Alguns exemplos disso são os recursos para compreensão e manipulação de listas, a capacidade de se poder retornar funções como resultado de expressões e usar uma função como parâmetro de outra função. Muitos desses recursos poderosos podem ser encontrados na sua biblioteca padrão. Sua alta capacidade de abstração também facilita o reuso.
Com um conjunto reduzido de estruturas lógicas e de controle e de tipos primitivos e por permitir a ligação entre diferentes tipos de dados, podemos classificar Phyton como tendo uma boa ortogonalidade.
No final, sua capacidade de escrita é alta para domínio de negócios e domínios mistos. Phyton alcança o objetivo de ser uma linguagem de propósito geral.

- Confiabilidade (Gabriel)
A verificação de dados em Phyton não realiza uma conversão implícita de um tipo de dado para outro e dá exceções quando tipos inconsistentes são utilizados. O que acontece é uma vinculação de nome e espaço a cada atribuição de valor a uma variável sendo que o tipo de dados do valor recebido é inferido. Também não existe uma definição dos tipos de parâmetros em funções ou método, ou seja, fica a cargo do programador o tratamento dos tipos recebido e retornados.
Para exceções Phyton se baseia no conceito de herança provindo da orientação a objetos. Todas as exceções são derivadas da classe Exception. Assim, graças a unificação entre classe e tipos – e subclasses sendo subtipos – ao tratar uma exceção de uma determinada classe todas as subclasses dessa exceção também são tratadas. Esse suporte ao tratamento de exceções nativo em Phyton é importante, pois evita que sejam feitas verificações que tornariam o código mais complexo e impactariam direto a legibilidade, o custo de implementação e a manutenção como em outras linguagens. A manipulação de exceções aumento a confiabilidade e contribuem para evitar que o programa seja finalizado de forma inesperada.
Também é importante para sua confiabilidade a facilidade encontrada no modelo de objetos que garante que objetos mutáveis com listas, dicionários e entidades externas como arquivos e janelas funcionem quase como ponteiros. Isso porque quando passados como argumento para uma função e modificados o chamador vê a mudança.
Possui um alto grau de flexibilidade o que impacta negativamente sua confiabilidade. Porém, com um modelo elegante de objetos, com o tratamento nativo de exceções orientado a objetos e com alta capacidade de abstração e recursos funcionais, isso é contrabalanceado.
A verificação de tipos pesa muito na confiabilidade geral da linguagem Phyton, tornando –a

uma linguagem de baixa confiabilidade.

- Custo (Amanda)
O custo de treinamento é baixo, graças a legibilidade e a simplicidade da linguagem Phyton. Também porque graças a comunidade ativa existem uma grande quantidade de tutoriais e programas exemplos disponíveis.
O custo de escrever programas é baixo graças a alta capacidade de escrita e de novo a legibilidade. Isso também barateia a manutenção de programas.
O código fonte é primeiro traduzido para um formato intermediário chamado de byte code que por sua vez é interpretado. Assim, como uma linguagem de interpretação híbrida, Phyton, não tem uma velocidade de execução rápida. O que afeta o custo de compilação e execução aumentando-os. Com a verificação de tipo durante a execução ela acaba dependendo praticamente de quão rápido é o interpretador.
É uma linguagem open source com uma comunidade ampla e espalhada pelo mundo, assim, sua documentação é muito bem definida e de fácil acesso podendo ser encontrada em http://wiki.python.org.br/DocumentacaoPython .
Em geral, Phyton é uma linguagem de baixo custo.

- Portabilidade (Guilherme)
Em sua implementação padrão Phyton é escrito em ANSI C portável que compila e executa em praticamente todas as principais plataformas em uso atualmente. Além do interpretador da linguagem, as bibliotecas padrões são implementadas para serem portáveis o máximo possível entre plataformas. Os programas são compilados automaticamente em código de bute portável que é executado igualmente em qualquer plataforma com uma versão de Phyton instalada. A portabilidade de Phyton tem crescido bastante nos últimos anos o que se deve muito ao incentivo e necessidade de sua enorme comunidade de desenvolvedores. Phytone é uma linguagem de confiabilidade média.

NOTA: O número em frente ao nome do título do slide corresponde ao número do slide do tema

O projeto (1) - Guilherme

A ideia do nosso projeto realizar um programa de mineração do twitter de maneira a integrar a biblioteca de acquisição de dados “Tweepy” juntamente com uma interface gráfica. De maneira a facilitar o processo de configuração e análise dos dados obtidos.

Para isso, realizamos um estudo sobre data mining, principalmente data mining no twitter usando python, e verificamos técnicas interessantes de análise de dados em tutoriais pela internet.

Decidimos, por isso, que analisaríamos as ocorrências de tokens mais frequentes, separando-os em tokens de propósito geral, hashtags e @-mentions, e verificaríamos a densidade geográfica dos tweets por meio de um mapa interativo.

Justificativas(1) - Amanda

Uma das justificativas para o uso de python como linguagem para realizar tal programa reside no fato de que essa linguagem apresenta grande uso na comunidade open-source, quando o assunto é data mining, machine learn, big data, data analysis.

Caso seja pesquisado no site Github, famoso repositório de projetos open-source, por “Data mining”, 1051 projetos são produzidos em python, 764 projetos so codificados em Java, 364 em R e apenas 171 em Javascript. Isso mostra que grande parte da comunidade open-source vinculada com projetos relativos a data mining utilizam python.

Além disso, python possui a biblioteca Tweepy, uma das bibliotecas mais famosas de mineração no twitter. Somado a isso, temos a biblioteca Tkinter, que permite a produção de interfaces gráficas de maneira fácil e intuitiva.

Justificativas(2) - Fritz

Em se tratando de dados não numéricos, comuns em aplicações do gênero, python apresenta-se como uma boa escolha devido as suas estruturas de dados padrão e suas estruturas de controle de sequência intuitivas.

Estruturas como dicionário facilitam muito a vida do programador, por exemplo, quando deseja manipular um arquivo JSON, muito comum em aplicações envolvendo acquisição de dados da internet. Esse pode ser transformado em um dicionário com muita facilidade e, assim, manipulado, dependendo das necessidades do programador.

As estruturas de controle permitem a realizaço de operações de maneira intuitiva e extremamente legível. (slide possui exemplos)

Especificidades (1) - Gabriel Martins

Usamos diversas bibliotecas padão do python:
A biblioteca threading possibilita a criação e manipulação de threads em geral, as quais foram essenciais para o funcionamento do programa.
A biblioteca RE, usada para compilar uma expressão regular, responsável pela tokenização dos textos obtidos pelo twitter.
A biblioteca json, que permite que jsons sejam manipulados facilmente no ambiente do python (principalmente como dicionários).

O Counter da biblioteca collections, que permite a contagem de tokens, a fim de verificar suas frequencias.
A biblioteca webbrowser, que permite o controle e utilização de browsers.

Além das bibliotecas padrão, usamos algumas bibliotecas externas:

A biblioteca tkinter, usada na geração da interface gráfica, bem como o controle de suas utilidades.

A biblioteca tweepy, para a obtenção dos dados a serem analisados
A biblioteca vincent, que permite a conversão de dados em formato dicionário do python em um formato inteligível para a biblioteca D3.js do javascript, muito boa para vizualização de dados.

Além disso, usamos dois arquivos HTML : “chart.html” e “map,html”, obtidos de um tutorial (licença aberta), que foram atualizados e modificados a fim de servir ao propósito da nossa aplicação.

Paradigmas(1) - Amanda

O paradigma orientado a objeto foi utilizado para quase todo o programa, a fim de criar classes que implementam as threads e a classe responsável pelo streaming dos dados do twitter.
O paradigma imperativo foi apenas utilizados a fim de criar a thread principal e executá-la.

Uso de Threads(1) - Fritz

Ao começar a desenvolver o programa, verificamos que o uso de threads não seria necessário para a maioria das operações, uma vez que todas essas eram capazes de serem executadas sem que tal execução interferisse no funcionamento da interface gráfica. Tais operações, quando chamadas no loop principal da interface (loop que roda na interface, captando possíveis pedidos de aplicação, por meio de pressionamento de botões), eram executadas, terminavam a execução e o loop da interface voltava a funcionar normalmente, realizando a captação dos acionamento dos botões.
Uso de Threads(2)- Fritz

Porém, a operação de obtenção de dados do twitter, por meio de um streaming, funciona por meio de um captador de eventos e roda indefinidamente. Com isso, o loop da interface gráfica nunca completava e a captação de pressionamentos parava de acontecer.

Uso de Threads(3) - Fritz

Por isso, foi necessária a implementação de uma thread cuja única função é a captação dos dados do twitter, até que uma determinada flag a mandasse parar. Separando o programa

nessas duas threads permitia que a acquisição de dados e a interface gráfica funcionassem de maneira paralela.

Implemetação(1): Slide não tem nada (Mostraremos o código e explicaremos).

Aqui (Mostrar class uiThread), implementamos a thread principal: - Amanda

Essa thread herda de threading, por isso, devemos chamar o construtor de threading dentro dela. ThreadID corresponde a identificação da thread, name corresponda ao nome da thread e counter corresponde a um contador de threads(que deve ser controlado pelo programador).

Nela desenvolvemos a interface gráfica (mostrar e explicar def run(self)): - Amanda

Nela usamos a variável global raiz (a qual representa o raiz da interface gráfica), para inicializar a interface e colocamos um frame associado à raiz em frame = Frame(raiz).

Após isso, geramos os botões associados aos comandos, que são métodos da classe,dispondo-os com o método grid, e executamos o loop principal da raiz, para captar as requisições do usuário.

A configuração de acesso do usuário (mostrar entrada_acesso, getaccess_secret, getaccess_key, getcons_secret, getcons_key): - Guilherme

O método entrada_acesso cria uma nova janela, com as entradas que o usuário deve digitar e os botões para pegar o que o usuário digitou e colocar na variável correspondente. As funções getaccess_secret, getaccess_key, getcons_secret e getcons_key são responsáveis por colocar valor digitado pelo usuário na variável correspondente, caso a entrada não seja nula ou string vazia.

Configurar a palavra-chave (key e getkey): - Gabriel Martins

O método key cria uma nova janela, com a entrada que o usuário deve digitar e o botão para pegar o que o usuário digitou e colocar na variável correspondente. A função getkey é responsável por colocar valor digitado pelo usuário na variável correspondente, caso a entrada não seja nula ou string vazia.

A interface da acquisição de dados (data_aquire e stop): - Guilherme

No método data_aquire, inicia-se a thread de acquisição de dados e cria-se uma janela com um botão de finalizar a acquisição de dados. Caso esse botão seja apertado, a nova janela é destruida e a flag para parar a acquisiço de dados é ativada.

A abertura dos dados (data_open e leitura_arquivo): - Amanda

No método data_open, cria-se uma nova janela com espaço para printar texto, e, por meio do método leitura_arquivo, recupera-se do arquivo os textos dos usuários.

Definir os tipos de tokens (token_type, tokens_hashtags, tokens_mentions e tokens_all): - Gabriel Martins

No método token_type, cria-se uma janela de botões para o usuário escolher que tipo de tokens são levados em consideração na hora de fazer o gráfico. O método tokens_all, quando acionado pelo botão, define que todos os tokens devem ser levados em consideração, o método tokens_hashtags, quando acionado pelo botão, define que todos somente tokens hashtag devem ser levados em consideração, o método tokens_mentions, quando acionado pelo botão, define que todos somente tokens @mentions devem ser levados em consideração.

Definir o número de tokens a mostrar no gráfico (token_num, getnumtk): - Guilherme

O método token_num abre uma janela com entradas para que se possa digitar o nmero de tokens a serem mostrados no gráfico de frequências. O boto get chama o método getnumtk que coloca o valor digitado pelo usuário na variável correspondente, caso entrada não seja nula ou string vazia, transformando- a em um inteiro.

A abertura dos gráficos dos tokens mais frequentes(tokenizer juntamente com chart.html): - Amanda

Gera a tokenização dos textos e apresentação em forma gráfica. (Fritz irá explicar)

A geolocalização (geoLocation juntamente com map.html): - Fritz

Gera geojsons e apresenta mapa interativo. (Fritz ir explicar)

A finalização do programa (finaliza): - Amanda

Finaliza a execução do programa.

thread secundaria - Fritz

A thread secundária é ativada quando a função de acquisição de dados da thread principal for chamada. Nesse método, realiza-se a criação da thread secundária (mostrar process thread). Ao executar o método run da thread secundária ,inicia-se o streaming (mostrar run). Na classe Leitor_Twitter, implementa-se o streamer (mostrar classe Leitor_Twitter). Nela, o método Leitor_Stream configura as chaves de acesso e inicia o streaming, com a palavra-chave escolhida. O método reescrito on_data é responsável por tratar o dado recebido, no nosso caso, apenas guardamos o dado em um arquivo (mostrar método on_data). Caso o usuário clique no boto sair da interface aberta, a variável global stop vira false e o streamer para de

adquirir dados. O método on_error (mostrar on error), trata os casos de erro de streaming.(Fritz ir explicar).

(Após mostrar código, revisar tudo com os slides, até o slide com título (Classe Leitor_Twitter),slide único).