Domínios da Linguagem python

1. Programação de sistemas

o tornam ideal para escrever ferramentas e utilitários de administração de sistemas portáveis e fáceis de manter (às vezes chamadas de ferramentas de shell).

* GUIs

A simplicidade e o rápido retorno do Python também o tornam bom para programação de GUI (interface gráfica com o usuário). O Python vem com uma interface orientada a obejtos padrão para a API de GUI Tk, chamada Tkinter, que permitem aos programas em Python implementarem GUIs portáveis com aparência e comportamento nativos.

* Scripts de Internet

Python possui módulos para a internet padrão que permitem aos programas executar uma grande variedade de tarefas em rede, tanto no modo cliente como servidor. Os scritps podem comunicar-se por meio de sockets; extrair informações de formulários enviados para um script CGI no lado servidor; transferir arquivos via FTP; processar URLs; analisar o código HTML e XML das paginas buscadas da Web. Além disso, existem um amplo conjunto de ferramentas de outros fornecedores na Web para fazer programação para internet em Python.

* Integração de componentes

Python possui facilidade para integrar componentes em sistemas C e C++. Ferramentas de código SWIG podem automatizar grande parte do trabalho necessário para vincular componentes compilados no Python, para uso em scripts.

* Programação de banco de dados

Existe um módulo denominado Pickle padrão do Python que fornece um sistema de persistência de objetos simples. Dessa forma. Permite que os programas salvem e restaurem facilmente objetos Python inteiros em arquivos e objetos do tipo arquivo. Para demanda de banco de dados tradicionais, existem interfaces Python para Sysbase, Oracle, ODBC, MySql e outros.

* Composição rápida de protótipos

Para os programas em Python, os componentes escritos na linguagem e em C parecem semelhantes. Por isso, é possível fazer o protótipo de sistemas inicialmente em Python e, então, mover componentes para uma linguagem compilada, como C e C++, para distribuição.

2.0 Critérios de Avaliação

* Legibilidade

O Python é uma linguagem bastante legível, o que não é de espantar, dado que descende do ABC, uma linguagem desenhada expressamente com fins didáticos. Usa o nível de indentação no início de cada linha para definir blocos de código, dispensando a utilização de chavetas ou de outros indicadores de início e fim de bloco

A sua legibilidade torna um programa escrito em Python mais num livro aberto do que num código para ser decifrado, facilitando a sua manutenção ou modificação.

* Capacidade de escrita

Python tem por objetivo ser minimalista e possui menos instruções básicas e palavras reservadas, o que contribui para o seu fácil aprendizado. Phyton procura implementar apenas um modo correto de se executar uma determinada operação, isso fortalece a simplicidade e consequentemente a legibilidade da linguagem. É uma linguagem de alto nível, com alta capacidade de abstração o que torna essa simplicidade poderosa e expressiva. A linguagem foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional. Prioriza a legibilidade do código sobre a velocidade ou expressividade.

* Confiabilidade

Tradicionalmente o sistema de verificação de tipos de uma linguagem de programação é classificado em dois grupos principais: estaticamente (fortemente) tipadas ou dinâmicamente (fracamente) tipadas. A capacidade de verificação de tipos de uma linguagem afeta diretamente a sua confiabilidade.

Python não se enquadram nesse modelo, pois é uma linguagem dinâmicamente tipada e ao mesmo tempo fortemente tipada. Python é dinamicamente tipada e também possui a inferência dos tipos. Porém, o sistema de verficação de tipos não realiza a conversão implícita de um tipo de dados para outro, leventando uma exceção quando tipos inconsistentes são utilizados.

Python possui um grau de flexibilidade muito alto e isso tem impacto na sua confiabilidade, mas que é contrabalanceado com um modelo de objetos elegante, tratamento de exceções nativo e orientado a objetos e alta capacidade de abstração e recursos da programação funcional. A sua capacidade de escrita é alta para domínio de negócios e domínios mistos.

* Custo

O custo de treinamento e o custo para escrever programas em Python é baixo, uma vez que a que capacidade de escrita e a legibilidade é alta. O tempo do entendimento de códigos exemplos na aprendizagem e o para a maneabilidade é baixo, otimizando o tempo de desenvolvimento.

Sua velocidade de execução nem sempre é tão rápida quanto linguagens compiladas, como C e C++. As implementações padrão do Python compilam instruções do código-fonte em formato intermediário (código de byte) e depois o interpretam

* Outros

Portabilidade

A implementação padrão do Python é escrita em ANSI C portável, compila e executa em praticamente todas as principais plataformas em uso atualmente. Por exemplo, os programa em Python são executados em tudo, de PDAs até supercomputadores.

3.0 Categorias (paradigmas)

* Python é uma linguagem de propósito geral de alto nível, multiparadigma, suporta o paradigma orientado a objetos, imperativo, funcional e procedural. Possui tipagem dinâmica e uma de suas principais características é permitir a fácil leitura do código e exigir poucas linhas de código se comparado ao mesmo programa em outras linguagens. Devido às suas características, ela é utilizada, principalmente, para processamento de textos, dados científicos e criação de CGIs para páginas dinâmicas para a web

4.0 Metados de Implementação

* CPython

Esta é a implementação original e a é a versão do Python que mais vem sendo desenvolvido e a mesma está escrita com a linguagem C. Novas funcionalidades ou recursos da linguagem aparecerão por aqui primeiro.

* Jython

Versão do Python implementado em Java. Esta implementação pode ser usada como linguagem de Script em aplicações Java, ou pode ser usada para criar aplicativos usando as bibliotecas das classes do Java. Também vem sendo bastante utilizado na construção de testes unitários para as bibliotecas do Java. Mais informações podem ser encontradas no the Jython website.

* Python for .NET

Essa implementação utiliza de fato a implementação CPython, mas é um aplicativo gerenciado .NET e disponibilizado como uma bibliotecas .NET. Foi desenvolvido por Brian Lloyd. Para obter mais informações, consulte Python for .NET home page.

* IronPython

Um versão alternativa do Python para a plataforma .NET. Ao contrário do Python.NET, esta é uma implementação completa do Python que gera IL e compila o código Python diretamente para assemblies .NET. Foi desenvolvida por Jim Hugunin, o criador original do Jython. Para obter mais informações, consulte the IronPython website.

* PyPy

Uma implementação do Python escrita completamente em Python. A mesma suporta vários recursos avançados não encontrados em outras implementações, como suporte sem pilhas e um compilador Just in Time. Um dos objetivos do projeto é incentivar a construção de experimentos com a própria linguagem, facilitando a modificação do interpretador (uma vez que o mesmos está escrito em Python).

Links pesquisados

<https://docs.python.org/pt-br/3.7/reference/introduction.html#alternate-implementations>

<https://lpunb.fandom.com/wiki/Linguagem_Python_-_2/2012_Grupo_2#Crit.C3.A9rios_de_avalia.C3.A7.C3.A3o>

<https://www.python.org/>