Trabalho de Formatura Supervisionado

Identificador de plágio para o Adessowiki

Aluno: Lucas Ikeda

Orientador: Prof. Roberto Hirata Jr.

Plágio

- Definição

Plágio

- Definição
- Preocupação acadêmica

Plágio

- Definição
- Preocupação acadêmica
- Problema: quantidade de tarefas

Plágio

- Definição
- Preocupação acadêmica
- Problema: quantidade de tarefas
- Existem ferramentas para código-fonte (Moss, JPlag) e texto (Sherlock), mas nenhuma está associada ao...

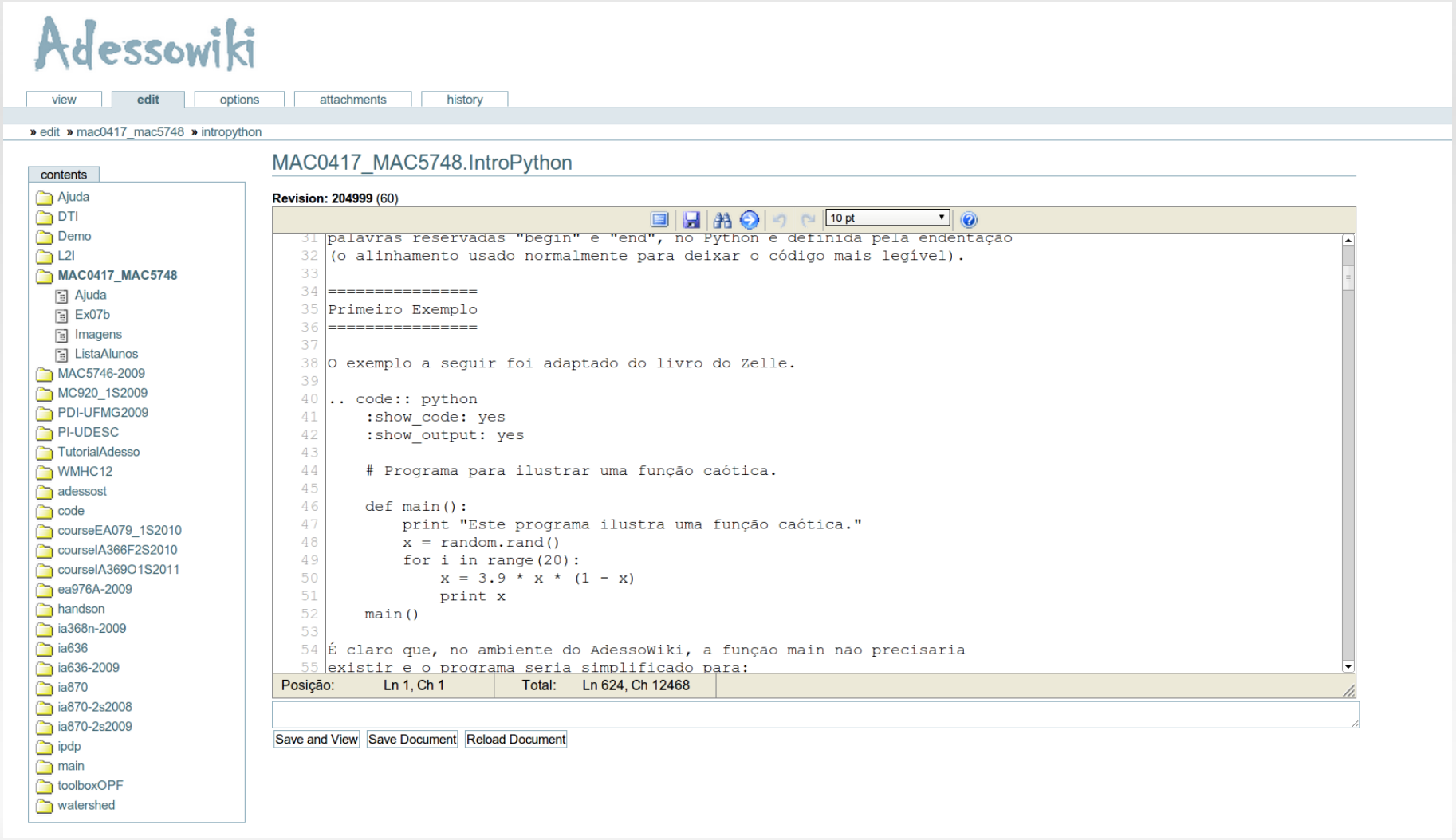
Adessowiki

- Ambiente de EAD desenvolvido por FEE-UNICAMP e CTI

Adessowiki

- Ambiente de EAD desenvolvido por FEE-UNICAMP e CTI
- Wiki programável
  - Python
  - C++
  - Textos (linguagem natural)

Adessowiki


Adessowiki
view
edit
options
attachments
history
» edit » mac0417_mac5748 » intropython
contents
Ajuda
DTI
Demo
L2I
MAC0417_MAC5748
Ajuda
Ex07b
Imagens
ListaAlunos
MAC5746-2009
MC920_1S2009
PDI-UFMG2009
PI-UDESC
TutorialAdesso
WMHC12
adessost
code
courseEA079_1S2010
courseIA366F2S2010
courseIA369O1S2011
ea976A-2009
handson
ia368n-2009
ia636
ia636-2009
ia870
ia870-2s2008
ia870-2s2009
ipdp
main
toolboxOPF
watershed
MAC0417_MAC5748.IntroPython
Revision: 204999 (60)
10 pt
palavras reservadas "begin" e "end", no Python é definida pela endentação
(o alinhamento usado normalmente para deixar o código mais legível).
==========
Primeiro Exemplo
==========
O exemplo a seguir foi adaptado do livro do Zelle.
.. code:: python
:show_code: yes
:show_output: yes
# Programa para ilustrar uma função caótica.
def main():
print "Este programa ilustra uma função caótica."
x = random.rand()
for i in range(20):
x = 3.9 * x * (1 - x)
print x
main()
É claro que, no ambiente do AdessoWiki, a função main não precisaria
existir e o programa seria simplificado para:
Posição: Ln 1, Ch 1
Total: Ln 624, Ch 12468
Save and View
Save Document
Reload Document

Adessowiki

Adessowiki

view | edit | options | attachments | history

» view » mac0417_mac5748 » intropython

contents

- Ajuda
- DTI
- Demo
- L2I
- **MAC0417_MAC5748**
  - Ajuda
  - Ex07b
  - Imagens
  - ListaAlunos
- MAC5746-2009
- MC920_1S2009
- PDI-UFMG2009
- PI-UDESC
- TutorialAdesso
- WMHC12
- adesso
- adessost
- code
- courseEA079_1S2010
- courseIA366F2S2010
- courseIA369O1S2011
- ea976A-2009
- handson
- ia368n-2009
- ia636
- ia636-2009
- ia870
- ia870-2s2008
- ia870-2s2009
- ipdp
- main
- toolboxOPF
- watershed

Introdução rápida à linguagem Python

Data: 04/03/2010

Nos últimos anos, uma grande quantidade de linguagens de programação dinâmicas surgiram no mundo. APL, Lisp (e afins), VB (e afins) eram as mais conhecidas e, nos anos 90, Perl fez muito sucesso junto a elas. De meados dos anos 90 para cá, muitas outras têm se sobressaído.

Python surgiu no início dos anos 90 e seu inventor é o holandês, Guido Van Rossum. Ela é uma linguagem que permite vários modelos de programação, quais sejam: funcional, imperativa, ou orientada a objetos. Além disso, ela é uma linguagem dinâmica, isto é, ela faz a verificação de tipo em tempo de execução, ao contrário de uma linguagem compilada que faz a verificação em tempo de compilação.

Estrutura básica de um programa em Python

Um programa Python é um conjunto de instruções na linguagem Python, cujos comandos básicos são bastante semelhantes aos de qualquer linguagem de programação. Talvez, a única novidade seja que a estrutura de um bloco de execução, que em C é definida pelas chaves ({ para abrir o bloco e } para fechar o bloco), ou em Pascal, ou Algol, é definada pelas palavras reservadas "begin" e "end", no Python é definida pela endentação (o alinhamento usado normalmente para deixar o código mais legível).

Primeiro Exemplo

O exemplo a seguir foi adaptado do livro do Zelle.

```
1 # Programa para ilustrar uma função caótica.
2
3 def main():
4     print "Este programa ilustra uma função caótica."
5     x = random.rand()
6     for i in range(20):
7         x = 3.9 * x * (1 - x)
8         print x
9 main()
```

```
Este programa ilustra uma função caótica.
0.280703137499
0.787444655778
0.652764722471
0.883985464316
0.399965136456
0.935972801695
0.233718093109
0.698466389644
0.821383259509
0.572179921979
```

Adessowiki

- Ambiente de EAD desenvolvido por FEE-UNICAMP e CTI
- Wiki programável
  - Python
  - C++
  - Textos (linguagem natural)
- Usado no IME, UNICAMP, UFMG, UDESC

Adessowiki

- Ambiente de EAD desenvolvido por FEE-UNICAMP e CTI
- Wiki programável
  - Python
  - C++
  - Textos (linguagem natural)
- Usado no IME, UNICAMP, UFMG, UDESC
- Conceito wiki pode agravar o problema

Solução

- Uma ferramenta que permita detectar casos de **possível** plágio entre páginas do Adessowiki

Solução

- Uma ferramenta que permita detectar casos de **possível** plágio entre páginas do Adessowiki
- Multiplataforma

Solução

- Uma ferramenta que permita detectar casos de **possível** plágio entre páginas do Adessowiki
- Multiplataforma
- Gere um relatório indicando os casos suspeitos

Técnicas de identificação

- Propriedades desejadas:

Técnicas de identificação

- Propriedades desejadas:
  - Ignorar trechos desinteressantes

Técnicas de identificação

- Propriedades desejadas:
  - Ignorar trechos desinteressantes
  - Ignorar reordenações

Técnicas de identificação

- Propriedades desejadas:
  - Ignorar trechos desinteressantes
  - Ignorar reordenações
  - Ignorar ruído

Técnicas de identificação

- Métrica
- Árvore sintática
- Latent Semantic Indexing (espaço vetorial)
- Winnowing

Winnowing

- Características:
  - Serve tanto para linguagem natural quanto código-fonte

Winnowing

- Características:
  - Serve tanto para linguagem natural quanto código-fonte
  - Eficiente

Winnowing

- Características:
  - Serve tanto para linguagem natural quanto código-fonte
  - Eficiente
  - Técnica já consagrada pelo Moss

Winnowing

- Características:
  - Serve tanto para linguagem natural quanto código-fonte
  - Eficiente
  - Técnica já consagrada pelo Moss
  - Fácil de implementar ☺

Winnowing

- Ideia geral do algoritmo:

Winnowing

- Ideia geral do algoritmo:
  - Dividir a entrada em k-gramas (substrings)

Winnowing

- Ideia geral do algoritmo:
  - Dividir a entrada em k-gramas (substrings)
  - Para cada k-grama, associá-lo a um valor (hash)

Winnowing

- Ideia geral do algoritmo:
  - Dividir a entrada em k-gramas (substrings)
  - Para cada k-grama, associá-lo a um valor (hash)
  - Selecionar os valores mínimos (em cada janela) como fingerprints do documento

Winnowing

- Ideia geral do algoritmo:
  - Dividir a entrada em k-gramas (substrings)
  - Para cada k-grama, associá-lo a um valor (hash)
  - Selecionar os valores mínimos (em cada janela) como fingerprints do documento
  - Para comparar a similaridade de arquivos, basta examinar os fingerprints correspondentes

Winnowing

- Pré-processamento visa eliminar dados sem importância, e depende do tipo de conteúdo:

Winnowing

- Pré-processamento visa eliminar dados sem importância, e depende do tipo de conteúdo:
  - Código-fonte:
    - Nome de variável → id
    - Nome de função → id
    - Comentários → λ

Winnowing

- Pré-processamento visa eliminar dados sem importância, e depende do tipo de conteúdo:
  - Código-fonte:
    - Nome de variável → id
    - Nome de função → id
    - Comentários → λ
  - Linguagem natural
    - Espaços em branco → λ
    - Pontuação → λ
    - A – Z → a – z

Exemplo de Winnowing

- original: 'Et omnia dicta fortiora si dicta latine.'

Exemplo de Winnowing

- original: 'Et omnia dicta fortiora si dicta latine.'
- pré-processado: 'etomniadictafortiorasidictalatine'

Exemplo de Winnowing

- original: 'Et omnia dicta fortiora si dicta latine.'
- pré-processado: 'etomniadictafortiorasidictalatine'
- k-gramas (k = 15):
  - 'etomniadictafor', 'tomniadictafort', 'omniadictaforti', 'mniadictafortio', 'niadictafortior', 'iadictafortiora', 'adictafortioras', 'dictafortiorasi', 'ictafortiorasid', 'ctafortiorasidi', 'tafortiorasidic', 'afortiorasidict', 'fortiorasidicta', 'ortiorasidictal', 'rtiorasidictala', 'tiorasidictalat', 'orasidictalati', 'orasidictalatin', 'rasidictalatine'

Exemplo de Winnowing

- original: 'Et omnia dicta fortiora si dicta latine.'
- pré-processado: 'etomniadictafortiorasidictalatine'
- k-gramas (k = 15):
  - 'etomniadictafor', 'tomniadictafort', 'omniadictaforti', 'mniadictafortio', 'niadictafortior', 'iadictafortiora', 'adictafortioras', 'dictafortiorasi', 'ictafortiorasid', 'ctafortiorasidi', 'tafortiorasidic', 'afortiorasidict', 'fortiorasidicta', 'ortiorasidictal', 'rtiorasidictala', 'tiorasidictalat', 'orasidictalati', 'orasidictalatin', 'rasidictalatine'
- hashes:
  - 33, 88, 9, 23, 36, 92, 82, 46, 36, 68,15, 13, 91, 46, 75, 86, 62, 11, 20

Exemplo de Winnowing

- janelas de hashes (w = 7):
  - 33, 88, **9**, 23, 36, 92, 82
  - 88, **9**, 23, 36, 92, 82, 46
  - **9**, 23, 36, 92, 82, 46, 36
  - **23**, 36, 92, 82, 46, 36, 68
  - **36**, 92, 82, 46, 36, 68, 75
  - 92, 82, 46, 36, 68, 75, **13**
  - 82, 46, 36, 68, 75, **13**, 91
  - 46, 36, 68, 75, **13**, 91, 46
  - 36, 68, 75, **13**, 91, 46, 75
  - 68, 75, **13**, 91, 46, 75, 86
  - 75, **13**, 91, 46, 75, 86, 62
  - 13, 91, 46, 75, 86, 62, **11**
  - 91, 46, 75, 86, 62, **11**, 20
- fingerprints:
  - **9**, **23**, **36**, **13**, **11**

Exemplo de Winnowing

- Para um outro texto, temos:
  - original: 'Omnia dicta fortiora, si graecis conscript.'
  - fingerprints: 9, 23, 36, 39, 34, 18, 6, 30

Exemplo de Winnowing

- Comparando os fingerprints:
  - Entrada A: 9, 23, 36, 13, 11
  - Entrada B: 9, 23, 36, 39, 34, 18, 6, 30

Exemplo de Winnowing

- Comparando os fingerprints:
  - Entrada A: 9, 23, 36, 13, 11
  - Entrada B: 9, 23, 36, 39, 34, 18, 6, 30
- Fingerprints comuns:
  - 9, 23, 36
  - Recuperando o trecho comum detectado nos textos originais:
    - '...omnia dicta fortiora...'

Exemplo de Winnowing

- Comparando os fingerprints:
  - Entrada A: 9, 23, 36, 13, 11
  - Entrada B: 9, 23, 36, 39, 34, 18, 6, 30
- Fingerprints comuns:
  - 9, 23, 36
  - Recuperando o trecho comum detectado nos textos originais:
    - '...omnia dicta fortiora...'
  - Obs: foram omitidos os dados posicionais dos fingerprints para simplificar a notação

Garantias

- Correção:
  - ∀ substrings correspondentes u, v, com $|u| \geq t, |v| \geq t$ (u, v) é detectado
  - $t = k + w - 1$

Garantias

- Correção:
  - ∀ substrings correspondentes u, v, com $|u| \geq t, |v| \geq t$ (u, v) é detectado
  - $t = k + w - 1$
- Limite inferior:
  - A densidade esperada em um algoritmo de fingerprinting: $$d \geq \frac{1.5}{w+1}$$
  - Densidade esperada no Winnowing: $$d = \frac{2}{w+1}$$

Implementação

- Objetivo: analisar conteúdos do Adessowiki

Implementação

- Objetivo: analisar conteúdos do Adessowiki
- Necessário separar os tipos de conteúdo de uma página

Implementação

- Objetivo: analisar conteúdos do Adessowiki
- Necessário separar os tipos de conteúdo de uma página
  - Parser: código Python, C++ e linguagem natural

Implementação

- Objetivo: analisar conteúdos do Adessowiki
- Necessário separar os tipos de conteúdo de uma página
  - Parser: código Python, C++ e linguagem natural
  - Criação de arquivos

Implementação

- Objetivo: analisar conteúdos do Adessowiki
- Necessário separar os tipos de conteúdo de uma página
  - Parser: código Python, C++ e linguagem natural
  - Criação de arquivos
- Simplificação: uso do Moss para análise de código-fonte

Implementação

- Objetivo: analisar conteúdos do Adessowiki
- Necessário separar os tipos de conteúdo de uma página
  - Parser: código Python, C++ e linguagem natural
  - Criação de arquivos
- Simplificação: uso do Moss para análise de código-fonte
  - Pré-processamento deste tipo de conteúdo é mais complicado (e Pyton ≠ C++)

Implementação

- Objetivo: analisar conteúdos do Adessowiki
- Necessário separar os tipos de conteúdo de uma página
  - Parser: código Python, C++ e linguagem natural
  - Criação de arquivos
- Simplificação: uso do Moss para análise de código-fonte
  - Pré-processamento deste tipo de conteúdo é mais complicado (e Pyton ≠ C++)
- Texto em linguagem natural é analisado localmente (Moss não suporta)

Implementação

- Saída: relatório com os casos suspeitos

Implementação

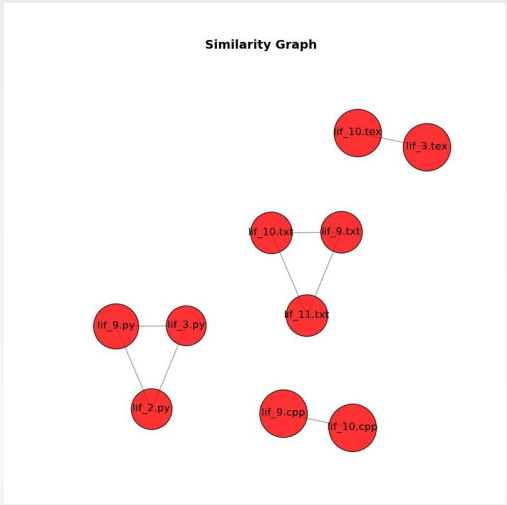
- Saída: relatório com os casos suspeitos
  - Deve agregar:
    - Resultado do Moss (mais um parser...)

Implementação

- Saída: relatório com os casos suspeitos
  - Deve agregar:
    - Resultado do Moss (mais um parser...)
    - Resultado da implementação local de Winnowing

Implementação

- Saída: relatório com os casos suspeitos
  - Deve agregar:
    - Resultado do Moss (mais um parser...)
    - Resultado da implementação local de Winnowing
  - Um grafo ajuda a visualizar as n-uplas suspeitas:



Resultados

- Ferramenta multiplataforma, pode ser integrada ao Adessowiki

Resultados

- Ferramenta multiplataforma, pode ser integrada ao Adessowiki
- Ainda em fase de testes...

FIM

Dúvidas

?