

Примена програмског језика Python у реализацији алгоритама за рангирање веб страница

Игор Илић

Математички факултет
Универзитет у Београду



Одбрана Мастер рада, 2016.



Садржај

- 1 Програмски језик Python
 - Синтакса програмског језика Python
 - Типови података у програмском језику Python
 - Употреба програмског језика Python
- 2 Претраживање и рангирање веб страница
 - Креирање и начин рада веб-паука
 - Рангирање страница
- 3 Закључак



Садржај

- 1 Програмски језик Python
 - Синтакса програмског језика Python
 - Типови података у програмском језику Python
 - Употреба програмског језика Python
- 2 Претраживање и рангирање веб страница
 - Креирање и начин рада веб-паука
 - Рангирање страница
- 3 Закључак



Рачунање и променљиве у Python-у

- Python поседује моћан калкулатор
- Коментари могу бити једнолинијски (`#`) или вишелинијски (`"""`)
- Променљиве реферишу на вредност која им се додељује

Primer

- `>>> p = 1`
- `>>> q = 2`
- `>>> p = q` `#` променљива `p` referise на `q`
- `>>> print p`
- `>>> 2`



Рачунање и променљиве у Python-у

- Python поседује моћан калкулатор
- Коментари могу бити једнолинијски (`#`) или вишелинијски (`"""`)
- Променљиве реферишу на вредност која им се додељује

Primer

- `>>> p = 1`
- `>>> q = 2`
- `>>> p = q # променљива p referise на q`
- `>>> print p`
- `>>> 2`



Рачунање и променљиве у Python-у

- Python поседује моћан калкулатор
- Коментари могу бити једнолинијски (`#`) или вишелинијски (`"""`)
- Променљиве реферишу на вредност која им се додељује

Primer

- `>>> p = 1`
- `>>> q = 2`
- `>>> p = q` `#` променљива `p` referise на `q`
- `>>> print p`
- `>>> 2`



Рачунање и променљиве у Python-у

- Python поседује моћан калкулатор
- Коментари могу бити једнолинијски (`#`) или вишелинијски (`"""`)
- Променљиве реферишу на вредност која им се додељује

Primer

- `>>> p = 1`
- `>>> q = 2`
- `>>> p = q # promenljiva p referise na q`
- `>>> print p`
- `>>> 2`



Услови и петље у Python-у

- Програмски блок је назубљен, са тачно 4 празна знака у односу на претходну линију
- Услов се реализује помоћу команди *if* и *else*
- *for*, *while* и *do* петље



Услови и петље у Python-у

- Програмски блок је назубљен, са тачно 4 празна знака у односу на претходну линију
- Услов се реализује помоћу команди *if* и *else*
- *for*, *while* и *do* петље



Услови и петље у Python-у

- Програмски блок је назубљен, са тачно 4 празна знака у односу на претходну линију
- Услов се реализује помоћу команди *if* и *else*
- *for*, *while* и *do* петље



Садржај

- 1 Програмски језик Python
 - Синтакса програмског језика Python
 - Типови података у програмском језику Python
 - Употреба програмског језика Python
- 2 Претраживање и рангирање веб страница
 - Креирање и начин рада веб-паука
 - Рангирање страница
- 3 Закључак



Уређене n-торке

- Дефинишу се као низ елемената раздвојених зарезом
- Непроменљиве у потпуности

Primer

- `>>> jedan, dva = (1, 2)`
- `>>> jedan`
- `>>> 1`



Уређене n-торке

- Дефинишу се као низ елемената раздвојених зарезом
- Непроменљиве у потпуности

Primer

- `>>> jedan, dva = (1, 2)`
- `>>> jedan`
- `>>> 1`



Уређене n-торке

- Дефинишу се као низ елемената раздвојених зарезом
- Непроменљиве у потпуности

Primer

- `>>> jedan, dva = (1, 2)`
- `>>> jedan`
- `>>> 1`



Ниске

- Низ знакова у оквиру знакова навода (' или ")
- Први знак ниске има индексни број 0, а последњи -1
- Операције са нискама: спајање, мултипликација, исецање, итд.

Primer

- ```
>>> str = "Dobar dan"
```
- ```
>>> print(str[0 : 5])
```
- ```
>>> Dobar
```



# Ниске

- Низ знакова у оквиру знакова навода (' или ")
- Први знак ниске има индексни број 0, а последњи -1
- Операције са нискама: спајање, мултипликација, исецање, итд.

## Primer

- ```
>>> str = "Dobar dan"
```
- ```
>>> print(str[0 : 5])
```
- ```
>>> Dobar
```



Ниске

- Низ знакова у оквиру знакова навода (' или ")
- Први знак ниске има индексни број 0, а последњи -1
- Операције са нискама: спајање, мултипликација, исецање, итд.

Primer

- ```
>>> str = "Dobar dan"
```
- ```
>>> print(str[0 : 5])
```
- ```
>>> Dobar
```



# Ниске

- Низ знакова у оквиру знакова навода (' или ")
- Први знак ниске има индексни број 0, а последњи -1
- Операције са нискама: спајање, мултипликација, исецање, итд.

## Primer

- ```
>>> str = "Dobar dan"
```
- ```
>>> print(str[0 : 5])
```
- ```
>>> Dobar
```



Листе

- Низ произвољних елемената између угластих заграда
- Индексни бројеви исти као и код ниски
- Операције и методи са листама: исецање, спајање, додавање новог члана, брисање одређеног члана, сортирање, итд.



Листе

- Низ произвољних елемената између угластих заграда
- Индексни бројеви исти као и код ниски
- Операције и методи са листама: исецање, спајање, додавање новог члана, брисање одређеног члана, сортирање, итд.



Листе

- Низ произвољних елемената између угластих заграда
- Индексни бројеви исти као и код ниски
- Операције и методи са листама: исецање, спајање, додавање новог члана, брисање одређеног члана, сортирање, итд.



Листе

- Низ произвољних елемената између угластих заграда
- Индексни бројеви исти као и код ниски
- Операције и методи са листама: исецање, спајање, додавање новог члана, брисање одређеног члана, сортирање, итд.



Мапе

- Мапе чине парови кључ-вредност, записани између витичастих заграда
- Операције и методи са мапама: додељивање вредности кључу, брисање кључа, враћање листе свих кључева, итд.



Мапе

- Мапе чине парови кључ-вредност, записани између витичастих заграда
- Операције и методи са мапама: додељивање вредности кључу, брисање кључа, враћање листе свих кључева, итд.



Садржај

- 1 Програмски језик Python
 - Синтакса програмског језика Python
 - Типови података у програмском језику Python
 - Употреба програмског језика Python
- 2 Претраживање и рангирање веб страница
 - Креирање и начин рада веб-паука
 - Рангирање страница
- 3 Закључак



Коришћење Python-а данас

- Развој веб апликација кроз употребу фрејмворка Django, Pyramid ili Bottle
- У прикупљању и анализирању података приликом научних истраживања
- Писање десктоп апликација, извршних скрипти, итд.



Коришћење Python-а данас

- Развој веб апликација кроз употребу фрејмворка Django, Pyramid ili Bottle
- У прикупљању и анализирању података приликом научних истраживања
- Писање десктоп апликација, извршних скрипти, итд.



Коришћење Python-а данас

- Развој веб апликација кроз употребу фрејмворка Django, Pyramid ili Bottle
- У прикупљању и анализирању података приликом научних истраживања
- Писање десктоп апликација, извршних скрипти, итд.



Садржај

- 1 Програмски језик Python
 - Синтакса програмског језика Python
 - Типови података у програмском језику Python
 - Употреба програмског језика Python
- 2 Претраживање и рангирање веб страница
 - Креирање и начин рада веб-паука
 - Рангирање страница

- 3 Закључак



Веб-паук

- 1 Проналажење и смештање свих хипервеза и кључних речи са дате странице
- 2 Оне странице које нису обрађене смештају се у листу необрађених *tocrawl*, а оне које обрадимо стављамо у листу обрађених *crawled*
- 3 Ограничавање рада по дубини или по броју страна



Веб-паук

- 1 Проналажење и смештање свих хипервеза и кључних речи са дате странице
- 2 Оне странице које нису обрађене смештају се у листу необрађених **tocrawl**, а оне које обрадимо стављамо у листу обрађених **crawled**
- 3 Ограничавање рада по дубини или по броју страна



Веб-паук

- 1 Проналажење и смештање свих хипервеза и кључних речи са дате странице
- 2 Оне странице које нису обрађене смештају се у листу необрађених **tocrawl**, а оне које обрадимо стављамо у листу обрађених **crawled**
- 3 Ограничавање рада по дубини или по броју страна



Садржај

- 1 Програмски језик Python
 - Синтакса програмског језика Python
 - Типови података у програмском језику Python
 - Употреба програмског језика Python
- 2 Претраживање и рангирање веб страница
 - Креирање и начин рада веб-паука
 - Рангирање страница
- 3 Закључак



Добијање одговора на упите

- Претражују се све странице које садрже одговарајућу кључну реч
- Резултат је листа страница
- Индекс свих кључних речи и хипервеза које им одговарају
- Убрзава се претраживање индекса коришћењем хеш табеле
- Хеш табела се реализује помоћу листи или мапа



Добијање одговора на упите

- Претражују се све странице које садрже одговарајућу кључну реч
- Резултат је листа страница
- Индекс свих кључних речи и хипервеза које им одговарају
- Убрзава се претраживање индекса коришћењем хеш табеле
- Хеш табела се реализује помоћу листи или мапа



Добијање одговора на упите

- Претражују се све странице које садрже одговарајућу кључну реч
- Резултат је листа страница
- Индекс свих кључних речи и хипервеза које им одговарају
- Убрзава се претраживање индекса коришћењем хеш табеле
- Хеш табела се реализује помоћу листи или мапа



Добијање одговора на упите

- Претражују се све странице које садрже одговарајућу кључну реч
- Резултат је листа страница
- Индекс свих кључних речи и хипервеза које им одговарају
- Убрзава се претраживање индекса коришћењем хеш табеле
- Хеш табела се реализује помоћу листи или мапа



Добијање одговора на упите

- Претражују се све странице које садрже одговарајућу кључну реч
- Резултат је листа страница
- Индекс свих кључних речи и хипервеза које им одговарају
- Убрзава се претраживање индекса коришћењем хеш табеле
- Хеш табела се реализује помоћу листи или мапа



Добијање одговора на упите

- Претражују се све странице које садрже одговарајућу кључну реч
- Резултат је листа страница
- Индекс свих кључних речи и хипервеза које им одговарају
- Убрзава се претраживање индекса коришћењем хеш табеле
- Хеш табела се реализује помоћу листи или мапа



Рангирање претражених страница

- PageRank алгоритам, оснивачи Гугла: Сергеј Брин и Лари Пејџ
- Страница је важна, ако се на њу показује са других важних страница
- Све странице иницијално добију вредност $\frac{1}{n}$, где је n број страница у индексу
- ранг се рачуна итеративно као сума количника ранга страница које показују на дату страницу и броја свих излазних линкова:
$$rank_{k+1}(P_i) = \alpha \sum_{P_j \in B_{P_i}} \frac{rank_k(P_j)}{|P_j|} + (1 - \alpha) \frac{1}{n}$$
- Уводи се граф у веб-паук, где се бележи на који начин се прескакало са странице на страницу
- Рачуна се ранг за сваку страницу, резултати се сортирају



Рангирање претражених страница

- PageRank алгоритам, оснивачи Гугла: Сергеј Брин и Лари Пејџ
- *Страница је важна, ако се на њу показује са других важних страница*
- Све странице иницијално добију вредност $\frac{1}{n}$, где је n број страница у индексу
- ранг се рачуна итеративно као сума количника ранга страница које показују на дату страницу и броја свих излазних линкова:
$$rank_{k+1}(P_i) = \alpha \sum_{P_j \in B_{P_i}} \frac{rank_k(P_j)}{|P_j|} + (1 - \alpha) \frac{1}{n}$$
- Уводи се граф у веб-паук, где се бележи на који начин се прескакало са странице на страницу
- Рачуна се ранг за сваку страницу, резултати се сортирају



Рангирање претражених страница

- PageRank алгоритам, оснивачи Гугла: Сергеј Брин и Лари Пејџ
- *Страница је важна, ако се на њу показује са других важних страница*
- Све странице иницијално добију вредност $\frac{1}{n}$, где је n број страница у индексу
- ранг се рачуна итеративно као сума количника ранга страница које показују на дату страницу и броја свих излазних линкова:
$$rank_{k+1}(P_i) = \alpha \sum_{P_j \in B_{P_i}} \frac{rank_k(P_j)}{|P_j|} + (1 - \alpha) \frac{1}{n}$$
- Уводи се граф у веб-паук, где се бележи на који начин се прескакало са странице на страницу
- Рачуна се ранг за сваку страницу, резултати се сортирају



Рангирање претражених страница

- PageRank алгоритам, оснивачи Гугла: Сергеј Брин и Лари Пејџ
- *Страница је важна, ако се на њу показује са других важних страница*
- Све странице иницијално добију вредност $\frac{1}{n}$, где је n број страница у индексу
- ранг се рачуна итеративно као сума количника ранга страница које показују на дату страницу и броја свих излазних линкова:
$$rank_{k+1}(P_i) = \alpha \sum_{P_j \in B_{P_i}} \frac{rank_k(P_j)}{|P_j|} + (1 - \alpha) \frac{1}{n}$$
- Уводи се граф у веб-паук, где се бележи на који начин се прескакало са странице на страницу
- Рачуна се ранг за сваку страницу, резултати се сортирају



Рангирање претражених страница

- PageRank алгоритам, оснивачи Гугла: Сергеј Брин и Лари Пејџ
- *Страница је важна, ако се на њу показује са других важних страница*
- Све странице иницијално добију вредност $\frac{1}{n}$, где је n број страница у индексу
- ранг се рачуна итеративно као сума количника ранга страница које показују на дату страницу и броја свих излазних линкова:
$$rank_{k+1}(P_i) = \alpha \sum_{P_j \in B_{P_i}} \frac{rank_k(P_j)}{|P_j|} + (1 - \alpha) \frac{1}{n}$$
- Уводи се граф у веб-паук, где се бележи на који начин се прескакало са странице на страницу
- Рачуна се ранг за сваку страницу, резултати се сортирају



Рангирање претражених страница

- PageRank алгоритам, оснивачи Гугла: Сергеј Брин и Лари Пејџ
- *Страница је важна, ако се на њу показује са других важних страница*
- Све странице иницијално добију вредност $\frac{1}{n}$, где је n број страница у индексу
- ранг се рачуна итеративно као сума количника ранга страница које показују на дату страницу и броја свих излазних линкова:
$$rank_{k+1}(P_i) = \alpha \sum_{P_j \in B_{P_i}} \frac{rank_k(P_j)}{|P_j|} + (1 - \alpha) \frac{1}{n}$$
- Уводи се граф у веб-паук, где се бележи на који начин се прескакало са странице на страницу
- Рачуна се ранг за сваку страницу, резултати се сортирају



Закључак

- Програмски језик Python
- Претраживање Интернета

Закључак

- Програмски језик Python
- Претраживање Интернета