Основи програмирања

Вежбе 2

Исидора Грујић
isidora@uni.kg.ac.rs

Лазар Илић
lazar@uni.kg.ac.rs

Филип Милић
milicf@uni.kg.ac.rs

Катедра за електротехнику и рачунарство
Факултет инжењерских наука Универзитета у Крагујевцу

Крагујевац, 14. октобар 2024.

Садржај

Садржај

Вежбе 1 - резиме

- врсте програмских језика

Вежбе 1 - резиме

- врсте програмских језика
- Пајтон 3/Python 3, инсталација, окружење

Вежбе 1 - резиме

- врсте програмских језика
- Пајтон 3/Python 3, инсталација, окружење
- учитавање података са стандардног улаза/исписивање на стандардном излазу

Вежбе 1 - резиме

- врсте програмских језика
- Пајтон 3/Python 3, инсталација, окружење
- учитавање података са стандардног улаза/исписивање на стандардном излазу
- типови објеката - скаларни објекти, претварање типова

Вежбе 1 - резиме

- врсте програмских језика
- Пајтон 3/Python 3, инсталација, окружење
- учитавање података са стандардног улаза/исписивање на стандардном излазу
- типови објеката - скаларни објекти, претварање типова
- аритметички оператори над бројевима, оператори поређења над бројевима, уграђене математичке функције

Вежбе 1 - резиме

- врсте програмских језика
- Пајтон 3/Python 3, инсталација, окружење
- учитавање података са стандардног улаза/исписивање на стандардном излазу
- типови објеката - скаларни објекти, претварање типова
- аритметички оператори над бројевима, оператори поређења над бројевима, уграђене математичке функције
- логички оператори

Вежбе 1 - резиме

- врсте програмских језика
- Пајтон 3/Python 3, инсталација, окружење
- учитавање података са стандардног улаза/исписивање на стандардном излазу
- типови објеката - скаларни објекти, претварање типова
- аритметички оператори над бројевима, оператори поређења над бројевима, уграђене математичке функције
- логички оператори
- дефинисање функција

Садржај

Нескаларни објекти: ниске

- сложени објекти
- имају унутрушњу структуру којој се може приступати
- **ниске**/**str** - најједноставнији нескаларни објекти
- низови алфанумеричких знакова

Нескаларни објекти: ниске

- сложени објекти
- имају унутрушњу структуру којој се може приступати
- **ниске**/**str** - најједноставнији нескаларни објекти
- низови алфанумеричких знакова

Пример 1

Разни примери ниски

Нескаларни објекти: ниске

- сложени објекти
- имају унутрушњу структуру којој се може приступати
- **ниске/str** - најједноставнији нескаларни објекти
- низови алфанумеричких знакова

Пример 1

Разни примери ниски

Пример 2

Учитавање података са улаза - тип објекта

Исписивање на стандардном излазу

Пример 3

Функција **print()** - навођење променљивих у витичастим заградама

Пример 4

Функција **print()** - одвајање делова реченице и променљивих запетама

Пример 5

Функција **format()** *vs* функција **round()**

Оператори над нискама

- "аритметички" оператори
  - сабирање: <прва_ниска> + <друга_ниска>

Оператори над нискама

- "аритметички" оператори
  - сабирање: <прва_ниска> + <друга_ниска>
  - множење целим бројем: <цео_број> * <ниска>

Пример 6

Функција **print()** - "додавање" променљивих у реченицу

- уграђене функције
  - функција **len()**

Оператори над нискама

- "аритметички" оператори
  - сабирање: <прва_ниска> + <друга_ниска>
  - множење целим бројем: <цео_број> * <ниска>

Пример 6

Функција **print()** - "додавање" променљивих у реченицу

- уграђене функције
  - функција **len()**
  - функција **lower()**

Оператори над нискама

- "аритметички" оператори
  - сабирање: <прва_ниска> + <друга_ниска>
  - множење целим бројем: <цео_број> * <ниска>

Пример 6

Функција **print()** - "додавање" променљивих у реченицу

- уграђене функције
  - функција **len()**
  - функција **lower()**
  - функција **upper()**

Оператори над нискама

- "аритметички" оператори
  - сабирање: <прва_ниска> + <друга_ниска>
  - множење целим бројем: <цео_број> * <ниска>

Пример 6

Функција **print()** - "додавање" променљивих у реченицу

- уграђене функције
  - функција **len()**
  - функција **lower()**
  - функција **upper()**
  - функција **islower()**

Оператори над нискама

- "аритметички" оператори
  - сабирање: <прва_ниска> + <друга_ниска>
  - множење целим бројем: <цео_број> * <ниска>

Пример 6

Функција **print()** - "додавање" променљивих у реченицу

- уграђене функције
  - функција **len()**
  - функција **lower()**
  - функција **upper()**
  - функција **islower()**
  - функција **isupper()**

Оператори над нискама

- "аритметички" оператори
  - сабирање: <прва_ниска> + <друга_ниска>
  - множење целим бројем: <цео_број> * <ниска>

Пример 6

Функција **print()** - "додавање" променљивих у реченицу

- уграђене функције
  - функција **len()**
  - функција **lower()**
  - функција **upper()**
  - функција **islower()**
  - функција **isupper()**
  - функција **strip()**

Оператори над нискама

- "аритметички" оператори
  - сабирање: <прва_ниска> + <друга_ниска>
  - множење целим бројем: <цео_број> * <ниска>

Пример 6

Функција **print()** - "додавање" променљивих у реченицу

- уграђене функције
  - функција **len()**
  - функција **lower()**
  - функција **upper()**
  - функција **islower()**
  - функција **isupper()**
  - функција **strip()**
  - функција **lstrip()**

Оператори над нискама

- "аритметички" оператори
  - сабирање: <прва_ниска> + <друга_ниска>
  - множење целим бројем: <цео_број> * <ниска>

Пример 6

Функција **print()** - "додавање" променљивих у реченицу

- уграђене функције
  - функција **len()**
  - функција **lower()**
  - функција **upper()**
  - функција **islower()**
  - функција **isupper()**
  - функција **strip()**
  - функција **lstrip()**
  - функција **rstrip()**
- оператори поређења

Извлачење делова ниски

- индексирање: <ниска>[<индекс>]

Пример 7

Извлачење делова ниски индексирањем - потенцијална грешка.

Извлачење делова ниски

- индексирање: <ниска>[<индекс>]

Пример 7

Извлачење делова ниски индексирањем - потенцијална грешка.

- одсецање:
  - <ниска>[<почетак>:<крај>]
  - <ниска>[<почетак>:<крај>:<корак>]

Пример 8

Извлачење делова ниски одсецањем - "обртање" ниске

Садржај

Гранање

- програм се не извршава увек праволинијски

Гранање

- програм се не извршава увек праволинијски
- најједноставнија наредба гранања је прост услов

Гранање

- програм се не извршава увек праволинијски
- најједноставнија наредба гранања је прост услов
- провера услова који као резултат даје објекат типа bool
  - услов је логички израз

Гранање

- програм се не извршава увек праволинијски
- најједноставнија наредба гранања је прост услов
- провера услова који као резултат даје објекат типа bool
  - услов је логички израз
- у логичком језику Пајтон гранање је имплементирано помоћу наредбе **if**

Структура **if** наредбе

- дефинисање блока наредби - **if <логички_израз>:**
  наредбе које желимо да се извршавају у случају да је испуњен услов представљен логичким изразом

Структура **if** наредбе

- дефинисање блока наредби - **if <логички_израз>:**
  наредбе које желимо да се извршавају у случају да је испуњен услов представљен логичким изразом
- опционо - додатни услови - **elif <други_логички_израз>:**
  наредбе које желимо да се извршавају у случају да је испуњен услов представљен логичким изразом у другом условном блоку

Гранање може бити **угнеждено** (енгл. *nested*).

Структура **if** наредбе

- дефинисање блока наредби - **if <логички_израз>:**
  наредбе које желимо да се извршавају у случају да је испуњен услов представљен логичким изразом
- опционо - додатни услови - **elif <други_логички_израз>:**
  наредбе које желимо да се извршавају у случају да је испуњен услов представљен логичким изразом у другом условном блоку
- коначно - **else:**
  наредбе које желимо да се извршавају у случају да ниједан од услова постављених логичким изразима није испуњен

Гранање може бити **угнеждено** (енгл. *nested*).

Садржај

Понављање

- неопходан је један или више начина за писање програма произвољне сложености

Понављање

- неопходан је један или више начина за писање програма произвољне сложености
- имплементација понављања користећи **while** петљу

Понављање

- неопходан је један или више начина за писање програма произвољне сложености
- имплементација понављања користећи **while** петљу
- дефинисање тела петље - **while <логички_израз>:**
  наредбе које желимо да се извршавају у итерацијама, све док је задовољен услов представљен логичким изразом

Садржај

Задаци

Задатке са других вежби можете пронаћи на страници предмета.