Tallinna Ülikool

Informaatika Instituut

TEKSTIANALÜSAATORI AUTOMAATMÄRGENDUSELE KASUTAJALIIDESE LOOMINE

Seminaritöö

Autor: Marko Sultsing

Juhendaja: Jaagup Kippar

Tallinn 2010

SISUKORD

LÜHENDID JA VÕÕRKEELSED MÕISTED

EVKK - Eesti vahekeele korpus

HTML - *HyperText Markup Language*, veebilehekülgede märgenduskeel

XHTML - *Extensible HyperText Markup Language*, laiendatud veebilehekülgede märgenduskeel; loodud HTML 4 ja XML 1.0 baasil

XML - *Extensible Markup Language*, märgenduskeel

VAKO - Eesti vahekeele korpuse keeletarkvara ja keeletehnoloogilise ressursi arendamise projekt

TAL - *Template Attribute Language*. Zope'isisene märgenduskeel, mille abil saab veebilehekülgedele dünaamilist sisu luua ja sisestada; kasutab spetsiaalseid atribuute, mis lisatakse XHTML märgenditele

ZPT - *Zope Page Template*; Zope'i tööriist, kus kirjutatakse HTMLi ja TALi kasutades valmis dünaamilise veebilehe lähtekood

SISSEJUHATUS

Käesoleva seminaritöö eesmärgiks on luua arendamisel olevale Eesti vahekeele korpuse tekstianalüsaatori automaatmärgendusele filoloogipoolne kasutajaliides. Antud liideses peab filoloogist kasutaja saama esitada teksti näol päringu keelekorpusele, näha vastuseks tulnud keelelisi andmeid, korrigeerida sealt vastuseks tulnud sõnu ning saata parandatud sõnu uuele ringile analüsaatoris. Tegemist on praktilise tööga, mis sisaldab endas eelkõige koodi kirjutamist vabavaralise Zope'i veebirakenduste serveri jaoks.

Antud töö autori huvi selle töö vastu on seotud sooviga luua midagi kasulikku, samal ajal ennast proovile pannes, proovides kätt huvitavate ja autori jaoks uute programmeerimisalaste lahendustega. Olles varem tegelenud nii programmeerimise kui ka veebi loomisega, meelitas idee tegeleda uudses veebikeskkonnas populaarse, kuid varem vähe kokku puututud programmeerimiskeelega.

Valmistudes kasutajaliidese loomiseks, sai autor esmakordselt lähemalt tutvust teha Zope 2 veebiraamistiku ning programmeerimiskeelega Python, milles Zope täiel määral kirjutatud on.

1. KASULIKUD EELTEADMISED

Alljärgnevas peatükis seletatakse lähemalt lahti nii EVKK olemus ning VAKO projekt , millest ühe osa moodustab käesoleva seminaritöö raames loodud kasutajaliides. Samuti käiakse üle töövahendid, mida töö käigus kasutati.

1.1.Eesti vahekeele korpus

Eesti vahekeele korpus (EVKK) on eesti keele kui teise keele kasutajate elektrooniline kirjalike tekstide kogu, kuhu kuulub üle 700 000 sõne ehk näidissõna. EVKK on valmis tehtud kasutades Tallinna Ülikoolis arendatavat vabavaralist veebitarkvara. Korpuse lähtekood on kõigile kättesaadav BSD litsentsi alusel, mis ei sea selle kasutamisele mingeid piiranguid. Keelekorpuses olevate andmete ja dokumentide esitamiseks kasutakse hüperteksti märgenduskeelt XHTML koos XPath keelega, viimase abil töödeldakse XML tüüpi dokumentides olevat informatsiooni.

Korpusesiseselt on enamik funktsioone kõikidele huvilistele kättesaadavad. Samas on kasutusel erinevad kasutajaliigid, mis võimaldavad anda erinevatele kasutajatele erinevaid õigusi. Kasutajaliidese kaudu saab kätte mitmesugust filoloogilist informatsiooni nii kõikide tekstide kohta kokku kui ka iga teksti kohta eraldi.

1.2.VAKO projekt

VAKO ehk Eesti vahekeele korpuse keeletarkvara ja keeletehnoloogilise ressursi arendamise projekti eesmärk on luua tarkvararakendused korpuse automaatseks töötlemiseks, mis võimaldavad minna üle tekstide käsitsimärgendamiselt üle poolautomaatsele. Üheks põhieesmärgiks on luua vealeidja prototüüp, teise eesmärgina loodetakse suurendada vahekeele korpuses olevate sõnede arvu 5 miljoni, samuti laiendada korpuse funktsionaalsust.

1.3.Python

Python on avatud lähtekoodiga populaarne programmeerimiskeel, milles käesoleva töö kood suures osas kirjutatud on. Python töötab kõikides enimlevinud operatsioonisüsteemides: Windows, Linux/Unix ja Mac OS X. Hetkel on sellel keelel kasutusel 2 põhiversiooni. Üks on vanem ning mitte enam aktiivselt uuendatav, kuid siiamaani populaarne Python 2. Uuem Python 3 on täiesti eraldiseisev versioon, milles kirjutatud kood ei pruugi tööle minna Python 2-s ja vastupidi. Selle seminaritöö raames loodud programmikood on kirjutatud Python 2.4.4-s, kuna tööks kasutatav veebirakenduste server Zope nõuab seda versiooni.

1.4.Zope

Zope on Pythonil põhinev veebirakenduste server, mõeldud eelkõige Internetis kasutatavate turvaliste rakenduste loomiseks ja jooksutamiseks. Siingi puhul on kasutusel kaks põhilist ning üpris erinevat versiooni: Zope 2 ja Zope 3. Mõlemad toetavaid vaid vanemaid Python 2 versioone.

Zope'i haldusliideses saab luua ning lisada mitmeid eri tüüpi objekte, mida saab seejärel omavahel koostööd tegema panna. Igale objektile määratakse unikaalne Id, mida kasutades on võimalik neid terve serverisisese failisüsteemi ulatuses välja kutsuda.

EVKK tekstianalüsaator, millele kood sai kirjutatud on üles ehitatud Zope 2 peal.

1.5.Programmeerimisalased mõisted

Siinkohal on lahti seletatud enimkasutatavad mõisted, millega käesoleva seminaritöö raames kirjutatud koodis kokku puututakse.

def - Pythoni käsk, millega alustatakse funktsiooni loomist, lühend sõnast definition (definitsioon, määratlus)

append() - Pythoni käsk, millega lisatakse mingi väärtus massiivi

count() - Pythoni käsk, millega loetakse, mitu korda mingi väärtus muutujas esineb

find() - Pythoni käsk, millega leitakse muutujast mingi väärtus või osa väärtusest

replace() - Pythoni käsk, millega asendatakse muutujas mingi väärtus teisega

tal:attributes - TAL atribuut, mida kasutades saab HTMLi atribuutidele lisada dünaamilisi väärtuseid

tal:condition - TAL atribuut, mida kasutades saab koodi panna kuvama vaid juhul, kui mingi nõudlus on täidetud

tal:content - TAL atribuut, mida kasutades saab märgendite sees oleva staatilise info täita selle olemasolu korral dünaamilisega

tal:define - TAL atribuut, mida kasutades saab erinevaid osasid koodis, näiteks pikema koodijupi defineerida ühesõnalise väärtusega, selleks, et seda hiljem mugavalt koodis kasutada

tal:repeat - TAL atribuut, mida kasutades saab korrata ühte osa koodist soovitud hulga kordi

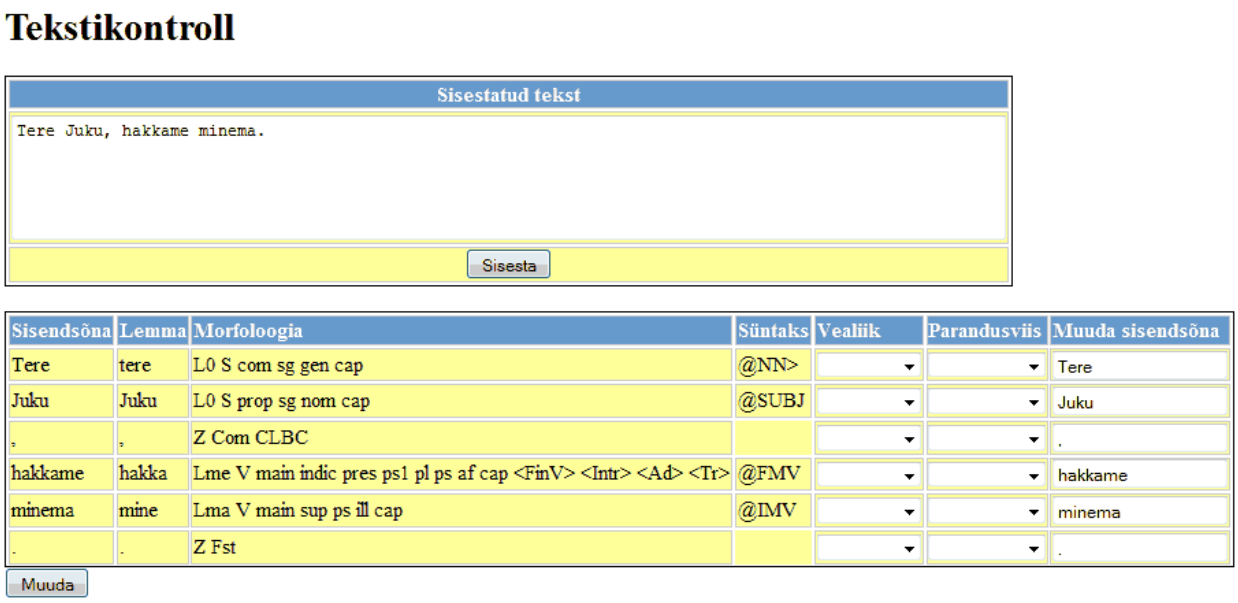
tal:replace - TAL atribuut, mida kasutades saab märgendite sees oleva staatilise info asendada selle olemasolu korral dünaamilisega

2. RAKENDUSE KOODI ÜLESEHITUS

Antud peatükk on pühendatud Zope'i keskkonna jaoks kirjutatud koodi lahti seletamisele

.

2.1.Kasutajaliides



Joonis 1. Pilt kasutajaliidesest, mis kuvab vastust päringule

Kasutajaliidese (vt joonis 1) väline pool on lihtsa ülesehitusega veebileht, kuvades iga analüsaatorist läbi lastud sõna kohta käiva info eraldi real, mis on omakorda infotüübi järgi ära jaotatud lahtritesse. Kasutaja saab kõik vajalikud funktsioonid ära teha ühel veebilehel. Päringteksti saab sisestada nii lehe ülaosas oleva tekstiala kui ka brauseri aadressiriba kaudu. Viimast lahendust on hiljem lihtne tervikliku süsteemiga kokku liita. Analüüsi läbinud sõnu saab vastavates tekstilahtrites korrigeerida ning tabeli all olevat nuppu „Muuda" vajutades uuele ringile saata.

Liidese taustal aga töötavad mitu Pythonis kirjutatud funktsiooni (*def*), mis on omavahel tööle pandud kasutades Zope'i enda dünaamiliste veebilehekülgede loomise tööriista *Zope Page Templates (ZPT)*, mis kasutab keelt *Template Attribute Language* (*TAL*).

Loodud mall-lehekülg (vt lisa 1) on kogu liidese keskosa, mis annab käsu nii Pythoni funktsioonide käivitamiseks, jagab nende vahel vajalikku informatsiooni ning genereerib lehekülje filoloogile kasutamiseks.

```
<tr tal:repeat="row python:range(len(tekstiAndmedMassiiv))">
          <td tal:define="sonaNr repeat/row/number;
                          sonaAndmedMassiiv
python:context.jupitamineSonaAndmeteks('',tekstiAndmedMassiiv,sonaNr-1)"
            tal:repeat="j sonaAndmedMassiiv"
            tal:content="j">
          </td>
```

Koodinäide 1. Sisestatud tekstis sõnade kätte saamise ja kuvamise eest vastutav osa kasutajaliidese koodis

Mall-lehekülg kasutab alguses loetavama koodi saamiseks *tal:define* atribuuti, et määrata Pythoni funktsioonidele ning neis kasutatavatele muutujatele lihtsad tunnussõnad. Hiljem sisestatakse need sobivates kohtades koodi, kasutades *tal:condition*, *tal:content*, *tal:replace* ja *tal:repeat* atribuute.

2.2.Kasutaja sisestatud teksti kätte saamine ning töötlemine

Kui kasutaja on sisestanud mingi teksti aadressireale, saadakse see *TAL*i poolt sealt käsku *request/QUERY_STRING* kasutades kätte ning saadetakse seejärel Pythoni skripti (vt koodinäide 2).

```
def tekstiSaamine(self, query):
```

```
""" Päringust teksti kätte saamine. """
if len(query)>6:
  if query.count("sisesta="):
    tekst=query[query.find("tekst=")+6:query.find("&sisesta=")]
    tekst=tekst.replace("%20"," ")
    tekst=tekst.replace("%2C",",")
    tekst=tekst.replace("+"," ")
  elif query.count("tekst=")>0 and query.count("uus_sisendsona_")==0:
    tekst=query[query.find("tekst=")+6:]
    tekst=tekst.replace("%20"," ")
  elif query.count("uus_sisendsona_")>0:
    tekst=query[query.find("uus_sisendsona_1=")+17:query.find("&muuda=")]
    tekst=tekst.replace("%20"," ")
    tekst=tekst.replace("%2C",",")
    tekst=tekst.replace("uus_sisendsona_","")
    while tekst.count("&")>0:
      tekst=tekst.replace(tekst[tekst.find("&"):tekst.find("=")+1]," ")
    tekst=tekst.replace(" .",".")
    tekst=tekst.replace(" ,",",")
else:
  tekst="Sisesta tekst."
return tekst
```

Koodinäide 2. Päringust teksti kätte saamine ning sobivale kujule vormindamine

Mainitud koodis uuritakse esimese asjana välja, kas päring on ka tegelikult tehtud. Kui päring on tehtud, siis uuritakse mil viisil seda tehti, vastavalt sellele meetodit valides eraldatakse sisestatud tekst ülejäänud päringust, töödeldakse järgmiseks etapiks vajalikule kujule ning seejärel väljastatakse veebilehele. Kui päringut pole tehtud, on väljundiks aga hoopis tekst „Sisesta tekst.“.

2.3.Tekstianalüsaatorisse päringu saatmine ning vastuse lugemine

Järgmise sammuna, peale kasutajaliidese poolt käsu saamist võtab asja üle Pythoni funktsioon, mis saadab tekstianalüsaatorisse kasutaja poolt sisestatud teksti kujul mida analüsaator oskab töödelda (vt koodinäide 3). Kasutades käske *urllib.urlopen()* ja *read()* saadakse kätte töötlemata vastus, mis sisaldab endas lisaks veebilehe koodile kõiki tekstis olnud sõnu ja kirjavahemärke ning iga sõna kohta eraldi käivaid morfosüntaktilisi andmeid.

```
params=urllib.urlencode({"tekst": tekst})
addr="http://evkk.tlu.ee/testcorpus/Search/morfoVastus?"+params
f=urllib.urlopen(addr)              ## saadab päringu tekstianalüsaatorile
tekstiAndmed=f.read()               ## loeb tekstianalüsaatori vastust
```

Koodinäide 3. Veebiaadressile teksti lisamisega analüsaatorile päringu saatmine ja vastuse lugemine

2.4.Morfosüntaktilise info esitamiseks sobivale kujule saamine

Tekstianalüsaatorist tulnud vastus vormindatakse nüüd sellisele kujule, et seda oleks hiljem lihtne töödelda. Seda tehakse koodis sedasi, et eemaldatakse tekstist üleliigsed osad, kasutades korduvalt *replace()* (vt koodinäide 4) ning mõningatel juhtudel abivahendina ka *find()* käsku.

```
tekstiAndmed=tekstiAndmed.replace("\n\n","")
```

Koodinäide 4. Kõik kahekordsed reavahetused asendatakse tühja väärtusega

Järgmiseks jupitatakse sobivale kujule töödeldud lauseosad massiivi sedasi, et iga väärtus koosneb sõnast, lemmast ehk algsõnast, morfoloogiast ning süntaksist, omavahel eraldatud tähisega |. Selline märk on kasutusel seetõttu, kuna on vähetõenäoline, et seda on sisestatud teksti

sees kasutatud. Peale seda, kui kõik andmed on massiivi lisatud, edastatakse see kasutajaliidesele.

Kasutajaliides võtab selle massiivi, loeb üle, mitu sõna seal kokku on, ning saadab seda arvu teades massiivi täpselt nii mitu korda järgmisesse Pythoni funktsiooni (vt koodinäide 5). Selles funktsioonis võetakse massiivist välja andmed täpselt nii mitmenda sõna kohta, mitmes ring funktsiooni jooksutamisel käib. Igal ringil jupitatakse info ühe sõna kohta neljaks, ning lisatakse eraldiseisvasse massiivi. Sedasi on kõik andmed valmis väljastamiseks, mida kood ka järgmisena teeb.

```
def jupitamineSonaAndmeteks(self,tekstiAndmedMassiiv,sonaNr):

    """ Massiivist sõnade andmete välja võtmine ning sobivale kujule
vormistamine. """

    sonaAndmedMassiiv=tekstiAndmedMassiiv[sonaNr].split("|")

    if len(sonaAndmedMassiiv)==3:  ## kui sõna kohta on vähem infot,
lisatakse üks tühi väärtus, et veebilehekülg kuvataks korrektselt

      sonaAndmedMassiiv.append("")

    return sonaAndmedMassiiv
```

Koodinäide 5. Kindel sõna koos andmetega jupitatakse ja väljastatakse

2.5.Morfosüntaktilise info esitamine kasutajaliideses

Kui ühe sõna kohta on kõik andmed esitatud, lisatakse selle kohta kasutajaliidesesse veel kolm lahtrit. Esimeses neist saab määrata sõna vigaseks, valides rippmenüüst selle kohta kehtiva vealiigi. Teine lahter sisaldab samuti rippmenüüd, sealt on võimalik valida parandusviisi, millega määratakse, kas on vaja parandada sõna ennast või selle kohta käivat morfoloogilist või/ja süntaktilist infot. Kolmas on tekstilahter, mis kuvab sisestatud sõna ning võimaldab seda parandada või hoopis teise sõnaga asendada (vt koodinäide 6).

Viimasena, peale kõikide sõnade esitamist lisatakse sõnade alla nupp, mis võimaldab muudetud sõnad uuesti tekstianalüsaatorisse saata. Selle nupu vajutamine laseb käima uue protsessi, mis kogub kokku kõik uued sisestatud sõnad, laseb need läbi tekstianalüsaatori ning kuvab uuesti genereeritud lehel.

```
<td tal:define="sonaNr repeat/row/number;

                            sonaAndmedMassiiv
python:context.jupitamineSonaAndmeteks('',tekstiAndmedMassiiv,sonaNr-1);

                            getSona
python:context.getSona('',sonaAndmedMassiiv,0)">

            <input type="text" name="UUS_SIDESONA_NR"
tal:condition="python:len(getSona)>0" tal:attributes="name
python:'uus_sisendsona_'+str(sonaNr);value getSona" />

          </td>
```

Koodinäide 6. Kasutajaliidesesse lisatakse tekstilahter, kus kuvatakse sisestatud sõna

3. EDASISED PLAANID KASUTAJALIIDESE ARENDAMISEKS

Aja jooksul on plaanis lisada kasutajaliidesele mõningad lisavõimalused ja täiendused. Nende lisamine jääb tõenäoliselt autori bakalaureusetöö raamidesse.

Üks eesmärk on täiendada vealiikide valimist, teine vastavalt parandusviisi valikule võimaldada parandada kas sõna morfoloogilist või süntaktilist osa. Arendamisel on ka lahendus, kus sõnades muutuste tegemisel jäetakse meelde nende originaalinfo ning kuvatakse see kasutajaliideses koos uutele andmetega. Lisaks, kuna seminaritöö tegemise ajal ei saanud võimaldada kahesuunalist ligipääsu korpuse andmebaasile, tuleb see hiljem sisse kodeerida. Vaheetapina võib kasutusel näha ka varianti, kus analüsaatorist saadud andmed salvestatakse tekstifaili.

Kuna keelekorpuse analüsaator pole käesoleval hetkel veel täiel kujul valmis ning selle väljund võib aja jooksul muutuda, võib hiljem näha ka selle koha pealt kasutajaliidesesse tehtud täiendusi.

KOKKUVÕTE

Käesoleva seminaritöö käigus sai valmis tehtud Zope'i veebirakenduste serveris toimiv kasutajaliides Eesti vahekeele korpuse tekstianalüsaatorile. Antud liides võimaldab filoloogidel sõnadele analüsaatori poolt antud märgendusi vaadata.

Autori jaoks on silmapiiril mõned käesoleva seminaritöö raamest välja jäänud täiendused, millega hakata tegelema bakalaureusetöö raames.

KASUTATUD KIRJANDUS

Eesti vahekeele korpus. Retrieved October 22, 2010 from http://evkk.tlu.ee/

Python Documentation. Retrieved October 15, 2010 from http://docs.python.org/

The Zope2 Book. Retrieved October 15, 2010 from http://docs.zope.org/zope2/zope2book/

VAKO - Eesti vahekeele korpuse keeletarkvara ja keeletehnoloogilise ressursi arendamine. Retrieved October 28, 2010, from http://www.keeletehnoloogia.ee/projektid/vako

LISAD

Lisa 1. Kasutajaliidese kood, XHTML ja TAL

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="ee" xml:lang="ee">

  <head>

    <title tal:content="context/title_or_id">LEHE TIITEL</title>

    <meta http-equiv="Content-Type" content="application/xhtml+xml;
charset=UTF-8" />

    <link rel="stylesheet" href="tekstikontroll.css" type="text/css" />

  </head>

  <body>

    <h1 tal:content="template/title_or_id">LEHE TIITEL VÕI ID</h1>

    <form method="get" action="tekstikontroll_pt"

          tal:define="query request/QUERY_STRING;

                      tekst python:context.tekstiSaamine('',query);

                      tekstiAndmedMassiiv
python:context.jupitamineSonadeks('',tekst);

                      sonaAndmedMassiiv
python:context.jupitamineSonaAndmeteks('',tekstiAndmedMassiiv,0);

                      getSona python:context.getSona('',sonaAndmedMassiiv,0);

                      kirjutaFaili
python:context.kirjutaFaili('',tekstiAndmedMassiiv)">

      <table class="sisestaTekst">

      <tr class="bold"><td>Sisestatud tekst</td></tr>

      <tr><td><textarea class="sisendTekst" name="tekst"  cols="80" rows="5"
tal:content="tekst">TEKST</textarea></td></tr>

      <tr><td><input type="submit" name="sisesta" value="Sisesta"
/></td></tr>

      </table>

      <br />

      <table tal:condition="python: len(query)>6">

        <tr class="bold">
```

```
<td>Sisendsõna</td><td>Lemma</td><td>Morfoloogia</td><td>Süntaks</td><td>Veal
iik</td><td>Parandusviis</td><td>Muuda sisendsõna</td>

        </tr>

        <tr tal:repeat="row python:range(len(tekstiAndmedMassiiv))">

          <td tal:define="sonaNr repeat/row/number;

                          sonaAndmedMassiiv
python:context.jupitamineSonaAndmeteks('',tekstiAndmedMassiiv,sonaNr-1)"

            tal:repeat="j sonaAndmedMassiiv"

            tal:content="j">

          </td>

          <td><select><option value="VEALIIK0"></option><option
value="VEALIIK1">VEALIIK1</option><option
value="VEALIIK2">JNE</option></select></td>

          <td><select><option value="PARANDUSVIIS0"></option><option
value="PARANDUSVIIS1">W_ERR</option><option
value="PARANDUSVIIS2">ERR_S</option><option
value="PARANDUSVIIS3">ERR_M</option><option
value="PARANDUSVIIS4">ERR_M_S</option></select></td>

          <td tal:define="sonaNr repeat/row/number;

                          sonaAndmedMassiiv
python:context.jupitamineSonaAndmeteks('',tekstiAndmedMassiiv,sonaNr-1);

                          getSona
python:context.getSona('',sonaAndmedMassiiv,0)">

            <input type="text" name="UUS_SIDESONA_NR"
tal:condition="python:len(getSona)>0" tal:attributes="name
python:'uus_sisendsona_'+str(sonaNr);value getSona" />

          </td>

        </tr>

      </table>

      <input type="submit" name="muuda" value="Muuda" tal:condition="python:
len(query)>6" />

    </form>

  </body>

</html>
```

Lisa 2. Kasutajaliidese kujunduse kood, CSS

```
table{
  border:1px solid #000;
}
td{
  border:1px solid #ccc;
  background-color:#ff9
}
tr.bold{
  font-weight:bold;
}
tr.bold td{
  background-color:#69c;
  color:#fff;
}
textarea.sisendTekst{
  width:800px;
  height:100px
}
table.sisestaTekst{
  text-align:center;
}
input[type=text]{
  width:100%
}
```

Lisa 3. Päringust teksti kätte saav kood, Python

```
def tekstiSaamine(self, query):
    """ Päringust teksti kätte saamine. """
    if len(query)>6:
      if query.count("sisesta="):
        tekst=query[query.find("tekst=")+6:query.find("&sisesta=")]
        tekst=tekst.replace("%20"," ")
        tekst=tekst.replace("%2C",",")
        tekst=tekst.replace("+"," ")
      elif query.count("tekst=")>0 and query.count("uus_sisendsona_")==0:
        tekst=query[query.find("tekst=")+6:]
        tekst=tekst.replace("%20"," ")
      elif query.count("uus_sisendsona_")>0:
        tekst=query[query.find("uus_sisendsona_1=")+17:query.find("&muuda=")]
        tekst=tekst.replace("%20"," ")
        tekst=tekst.replace("%2C",",")
        tekst=tekst.replace("uus_sisendsona_","")
        while tekst.count("&")>0:
          tekst=tekst.replace(tekst[tekst.find("&"):tekst.find("=")+1]," ")
        tekst=tekst.replace(" .",".")
        tekst=tekst.replace(" ,",",")
    else:
      tekst="Sisesta tekst."
    return tekst
```

Lisa 4. Keeleanalüsaatorile päringu saatev ning saadud vastust töötlev kood, Python

```
def jupitamineSonadeks(self, tekst="Tere Juku, hakkame minema"):

    """ Saadab päringu tekstianalüsaatorile, saadud vastuse vormindab
sobivale kujule ja paigutab sõnadekaupa massiivi. """

    import urllib

    params=urllib.urlencode({"tekst": tekst})

    addr="http://evkk.tlu.ee/testcorpus/Search/morfoVastus?"+params

    f=urllib.urlopen(addr)                                          ## saadab
päringu tekstianalüsaatorile

    tekstiAndmed=f.read()                                           ## loeb
tekstianalüsaatori vastust

    tekstiAndmed=tekstiAndmed.replace("\"</s>\"\n\n\"<s>\"","")

    tekstiAndmed=tekstiAndmed.replace("\"<s>\"\n\n","")

    tekstiAndmed=tekstiAndmed.replace("\"</s>\"","")

    tekstiAndmed=tekstiAndmed.replace("\n\"</s>\"","")

    tekstiAndmed=tekstiAndmed.replace(">\"\n","")

    tekstiAndmed=tekstiAndmed.replace("\t","")

    tekstiAndmed=tekstiAndmed.replace("\"<","")

    tekstiAndmed=tekstiAndmed.replace("\" ","|")

    tekstiAndmed=tekstiAndmed.replace("\"","|")

    tekstiAndmed=tekstiAndmed.replace("\n\n","")

    tekstiAndmed=tekstiAndmed.replace("Vigased","")

    tekstiAndmedMassiiv=tekstiAndmed.split("\n")                    ##paneb
sobivale kujule vormindatud tekstianalüsaatori vastuse massiivi

    ajutineMassiiv=[]

    for x in tekstiAndmedMassiiv:                                   ##
tsükkel, mis teeb viimased täiustused massiivi andmetele

      x=x[0:x.find(" #")]
```

```
      if(x.count("@")>0):                                              ## kui
leiab stringist "@"-i

        x=x[0:x.find(" @")]+"|@"+x[x.find(" @")+2:]      ## asendab " @" "|@"-
ga

      if len(x)>0:

        ajutineMassiiv.append(x)

    tekstiAndmedMassiiv=ajutineMassiiv

    return tekstiAndmedMassiiv
```

Lisa 5. Massiivis olevaid sõnaandmeid lugev ning töötlev kood, Python

```
def jupitamineSonaAndmeteks(self,tekstiAndmedMassiiv,sonaNr):

    """ Massiivist sõnade andmete välja võtmine ning sobivale kujule
vormistamine. """

    sonaAndmedMassiiv=tekstiAndmedMassiiv[sonaNr].split("|")

    if len(sonaAndmedMassiiv)==3:                                    ## kui
sõna kohta on vähem infot, lisatakse üks tühi väärtus, et veebilehekülg
kuvataks korrektselt

      sonaAndmedMassiiv.append("")

    return sonaAndmedMassiiv
```

Lisa 6. Massiivist sõna välja võttev kood

```
def getSona(self,sonaAndmedMassiiv,sonaAndmeNr=0):

    return sonaAndmedMassiiv[sonaAndmeNr]
```