



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Αναφορά Project

Μέλη ομάδας

Επώνυμο - Όνομα	ΑΜ	Έτος
Βακαλόπουλος Δημήτριος	1059564	3ο
Τάσσιος Βλάσιος	1047178	5ο
Τριμίντζιος Ηλίας	1047200	5ο
Τσουβάλας Άγγελος	1047049	5ο

Πίνακας περιεχομένων

Κατάλογος Εικόνων	3
Περιγραφή της γραμματικής της γλώσσας σε BNF.....	4
Περιγραφή του αντίστοιχου υποσυνόλου της Python σε BNF	8
<i>Προαιρετική εισαγωγή από modules</i>	<i>8</i>
<i>Αρχικοποίηση μεταβλητών.....</i>	<i>8</i>
Boolean operations:	8
<i>Ορισμός κλάσης με τον επικείμενο constructor και δημιουργία αντικειμένου</i>	<i>8</i>
<i>Ορισμός συνάρτησης και κλήση της</i>	<i>9</i>
<i>If statement</i>	<i>9</i>
<i>For statement</i>	<i>9</i>
<i>Υποστήριξη εμφάνισης μηνυμάτων</i>	<i>9</i>
<i>Όλες οι δεσμευμένες λέξεις της γλώσσας είναι υποχρεωτικά με πεζούς</i>	<i>10</i>
Return statement	10
Break statement.....	10
Continue statement	10
Import statement	10
Global statement.....	10
Try statement.....	10
With statement	10
<i>Χαρακτήρες.....</i>	<i>11</i>
Screenshots παραδειγμάτων εφαρμογής.....	12
<i>Παραδείγματα με ανάθεση.....</i>	<i>12</i>
<i>Παραδείγματα με δεσμευμένες λέξεις</i>	<i>13</i>
<i>Παραδείγματα με “dictionary”.....</i>	<i>14</i>
<i>Παραδείγματα με “for statement”</i>	<i>14</i>
<i>Παραδείγματα με “if statement”</i>	<i>15</i>
<i>Παραδείγματα με “import”</i>	<i>15</i>
<i>Παραδείγματα με τον επικείμενο constructor.....</i>	<i>16</i>
<i>Παραδείγματα με “Lambda Calculus”.....</i>	<i>16</i>
<i>Παραδείγματα με “object”</i>	<i>17</i>
<i>Παραδείγματα με “print”</i>	<i>17</i>
<i>Παραδείγματα με συνάρτηση</i>	<i>18</i>
<i>Παραδείγματα με σχόλια</i>	<i>18</i>

Κατάλογος Εικόνων

Εικόνα 1: Ανάθεση_string	12
Εικόνα 2: Ανάθεση_integer	12
Εικόνα 3: Ανάθεση_sum	12
Εικόνα 4: Ανάθεση_float	12
Εικόνα 5: δεσμευμένη λέξη_def	13
Εικόνα 6: δεσμευμένη λέξη_print	13
Εικόνα 7: δεσμευμένη λέξη_return	13
Εικόνα 8: dictionary χωρίς όρισμα	14
Εικόνα 9: dictionary με 1 όρισμα	14
Εικόνα 10: dictionary με 2 ορίσματα	14
Εικόνα 11: dictionary με 3 ορίσματα	14
Εικόνα 12: dictionary με 4 ορίσματα	14
Εικόνα 13: for stmt	14
Εικόνα 14: for stmt with range	14
Εικόνα 15: if_stmt_EQ	15
Εικόνα 16: if_stmt_GT	15
Εικόνα 17: if_stmt_GE	15
Εικόνα 18: if_stmt_LT	15
Εικόνα 19: if_stmt_LE	15
Εικόνα 20: import_I	15
Εικόνα 21: import_II	15
Εικόνα 22: constructor χωρίς ορίσματα	16
Εικόνα 23: constructor με ορίσματα	16
Εικόνα 24: Lambda_Calculus_I	16
Εικόνα 25: Lambda_Calculus_II	16
Εικόνα 26: Lambda_Calculus_III	16
Εικόνα 27: Lambda_Calculus_IV	16
Εικόνα 28: δημιουργία αντικειμένου	17
Εικόνα 29: κάλεσμα αντικειμένου	17
Εικόνα 30: print_I	17
Εικόνα 31: print_II	17
Εικόνα 32: print_III	17
Εικόνα 33: συνάρτηση χωρίς ορίσματα	18
Εικόνα 34: συνάρτηση με όρισμα	18
Εικόνα 35: συνάρτηση με ορίσματα	18
Εικόνα 36: σχόλιο_I	18
Εικόνα 37: σχόλιο_II	18

Περιγραφή της γραμματικής της γλώσσας σε BNF

Οι παρακάτω εντολές εκτυπώνουν το μήνυμα *"This is a valid python expression"* , υπό την προϋπόθεση ότι η έκφραση που εκχωρήσαμε είναι αποδεκτή από την γραμματική της Python:

```
start: stmt NEWLINE {printf("\nThis is a valid python expression\n"); border; YYACCEPT;}  
      | stmt2 NEWLINE {printf("\nThis is valid expression \n"); border; YYACCEPT;}
```

Παρακάτω χωρίζουμε το statement σε 3 διαφορετικά κομμάτια (αριθμητική ανάθεση, string και εμφάνιση):

```
stmt: assign_arithmetic  
      | assign_str  
      | display
```

Παρακάτω ορίζουμε τις υλοποιήσεις της Python (σχόλια, κλάση, συνάρτηση, δημιουργία αντικειμένου, κάλεσμα αντικειμένου, ανάθεση if, lamda, for και import):

```
stmt2: Comment  
       | klasi  
       | sinartisi  
       | objectCreation  
       | objectCalling  
       | Ifstatement  
       | lambdastm  
       | forStatement  
       | importStatement
```

Παρακάτω είναι το id που έχουμε ορίσει στη flex. Με βάση αυτό αν γραφτεί μια δεσμευμένη λέξη τότε βγάζει error:

```
identifier: ID  
           | keyword {yyerror("\nkeyword can't be used as a identifier\n"); YYABORT;}
```

Παρακάτω ορίζονται οι δεσμευμένες λέξεις:

```
keyword: PRINT
        | RARE
        | CLASS
        | DEF
        | IF
        | LAMBDA
        | IN
        | FOR
        | RANGE
        | IMPORT
```

Παρακάτω γίνεται η ανάθεση string:

```
assign_str: identifier EQ strings
```

Παρακάτω είναι η περίπτωση που χρησιμοποιήσουμε την εντολή print:

```
display: PRINT strings
        | PRINT strings MUL NUMBER
        | PRINT strings PLUS strings
        | PRINT expr
```

Παρακάτω φαίνεται το πως ορίζεται το string:

```
strings: STRING_LIT
        | STRING_VAR
        | SSS
```

Παρακάτω γίνεται αριθμητική ανάθεση:

```
assign_arithmetic: identifier EQ expr
```

Παρακάτω φαίνεται το πως μπορεί να οριστεί μια έκφραση στην Python:

```
expr: expr PLUS expr
      | expr MINUS expr
      | expr MUL expr
      | expr DIVIDE expr
      | factor
      | LBRACKET expr RBRACKET
      | SIGN factor
```

Παρακάτω διαχωρίζουμε τους τύπους των αριθμών:

```
factor: identifier
        | NUMBER
        | FLOAT
```

Παρακάτω φαίνονται οι περιπτώσεις των “if”:

```
ifstatement: IF identifier EQ EQ identifier COLON
             | IF identifier LT identifier COLON
             | IF identifier LE identifier COLON
             | IF identifier GE identifier COLON
             | IF identifier GT identifier COLON
```

Παρακάτω φαίνονται κάποιες από τις περιπτώσεις της “lamda calculus”:

```
lambdastm: LAMBDA factor COLON factor
           | LAMBDA factor COLON factor PLUS factor
           | LAMBDA factor COLON factor MINUS factor
           | LAMBDA factor COLON factor MUL factor
           | LAMBDA factor COLON factor DIVIDE factor
           | LAMBDA factor COMMA factor COLON factor PLUS factor
           | LAMBDA factor COMMA factor COLON factor MINUS factor
           | LAMBDA factor COMMA factor COLON factor MUL factor
           | LAMBDA factor COMMA factor COLON factor DIVIDE factor
```

| LBRACKET LAMBDA factor COLON factor PLUS factor RBRACKET

Παρακάτω φαίνεται ο ορισμός της κλάσης:

klasi: CLASS identifier COLON

Παρακάτω φαίνεται ο ορισμός της συνάρτησης:

sinartisi: DEF identifier SSS COLON

Παρακάτω φαίνεται ο ορισμός της δημιουργίας αντικειμένου, καθώς και το κάλεσμά του:

objectCreation: identifier EQ identifier SSS

objectCalling: identifier DOT identifier SSS

Παρακάτω φαίνεται ο ορισμός της “for”:

forStatement: FOR identifier IN STRING_LIT COLON

| FOR identifier IN identifier COLON

| FOR identifier IN RANGE SSS COLON

Παρακάτω φαίνεται ο ορισμός του “import”:

importStatement: IMPORT identifier

| IMPORT identifier AS identifier

Παρακάτω φαίνεται ο ορισμός του “dictionary”:

dictCreation: identifier EQ LHOOK RHOOK

| identifier EQ LHOOK factor COLON factor RHOOK

| identifier EQ LHOOK factor COLON factor COMMA factor COLON factor RHOOK

| identifier EQ LHOOK factor COLON factor COMMA factor COLON factor COMMA factor COLON factor RHOOK

| identifier EQ LHOOK factor COLON factor COMMA factor COLON factor COMMA factor COLON factor COMMA factor COLON factor RHOOK

Περιγραφή του αντίστοιχου υποσυνόλου της Python σε BNF

Προαιρετική εισαγωγή από modules

```
<keyword>:= "AND" | "DEL" | "FROM" | "NOT" | "WHILE"  
| "AS" | "ELIF" | "GLOBAL" | "OR" | "WITH"  
| "ASSERT" | "ELSE" | "IF" | "PASS" | "YIELD"  
| "BREAK" | "EXCEPT" | "IMPORT" | "PRINT"  
| "CLASS" | "EXEC" | "IN" | "RAISE"  
| "CONTINUE" | "FINALLY" | "IS" | "RETURN"  
| "DEF" | "FOR" | "LAMBDA" | "TRY";
```

Αρχικοποίηση μεταβλητών

integer ::= decimalinteger | octinteger | hexinteger

floatnumber ::= pointfloat | exponentfloat

Boolean operations:

or_test ::= and_test | or_test "or" and_test

and_test ::= not_test | and_test "and" not_test

not_test ::= comparison | "not" not_test

Ορισμός κλάσης με τον επικείμενο constructor και δημιουργία αντικειμένου

classdef ::= [decorators] "class" classname [inheritance] ":" suite

inheritance ::= "(" [argument_list] ")"

classname ::= identifier

Ορισμός συνάρτησης και κλήση της

```
funcdef      ::= [decorators] "def" funcname "(" [parameter_list] ")"  
              ["->" expression] ":" suite  
decorators   ::= decorator+  
decorator    ::= "@" dotted_name "(" [argument_list [","]] ")" NEWLINE  
dotted_name  ::= identifier "." identifier)*  
parameter_list ::= defparameter ("," defparameter)* "," "/" ["," [parameter_list_no_posonly]]  
              | parameter_list_no_posonly  
parameter_list_no_posonly ::= defparameter ("," defparameter)* ["," [parameter_list_starargs]]  
              | parameter_list_starargs  
parameter_list_starargs ::= "*" [parameter] ("," defparameter)* ["," ["**" parameter [","]]]  
              | "**" parameter [","]  
parameter    ::= identifier ":" expression  
defparameter ::= parameter ["=" expression]  
funcname     ::= identifier
```

If statement

```
if_stmt ::= "if" assignment_expression ":" suite  
         ("elif" assignment_expression ":" suite)*  
         ["else" ":" suite]
```

For statement

```
for_stmt ::= "for" target_list "in" expression_list ":" suite  
          ["else" ":" suite]
```

Υποστήριξη εμφάνισης μηνυμάτων

```
<simple_stmt>:=<expression_stmt>|<assert_stmt>|<assignment_stmt>|<augmented_assignment_stmt>|  
<pass_stmt>|<del_stmt>|<print_stmt>|<return_stmt>|  
<yield_stmt>|<raise_stmt>|<break_stmt>|<continue_stmt>|<import_stmt>|  
<global_stmt>|<exec_stmt>;
```

```
<print_stmt>:="print" ([<expression>(","<expression>)* [","]]  
| ">>"<expression>[(","<expression>)+ [","]] );
```

Όλες οι δεσμευμένες λέξεις της γλώσσας είναι υποχρεωτικά με πεζούς

Return statement

return_stmt ::= "return" [expression_list]

Break statement

break_stmt ::= "break"

Continue statement

continue_stmt ::= "continue"

Import statement

```
import_stmt ::= "import" module ["as" identifier] ("," module ["as" identifier])*  
              | "from" relative_module "import" identifier ["as" identifier]  
              ("," identifier ["as" identifier])*  
              | "from" relative_module "import" "(" identifier ["as" identifier]  
              ("," identifier ["as" identifier])* [","] ")"  
              | "from" module "import" "*"
module      ::= (identifier ".")* identifier
relative_module ::= "."* module | "."+
```

Global statement

global_stmt ::= "global" identifier ("," identifier)*

Try statement

```
try_stmt ::= try1_stmt | try2_stmt
try1_stmt ::= "try" ":" suite  
              ("except" [expression ["as" identifier]] ":" suite)+  
              ["else" ":" suite]  
              ["finally" ":" suite]
try2_stmt ::= "try" ":" suite  
              "finally" ":" suite
```

With statement

with_stmt ::= "with" with_item ("," with_item)* ":" suite

with_item ::= expression ["as" target]

Χαρακτήρες

stringliteral ::= [stringprefix](shortstring | longstring)

stringprefix ::= "r" | "u" | "R" | "U" | "f" | "F"
| "fr" | "Fr" | "fR" | "FR" | "rf" | "rF" | "Rf" | "RF"

shortstring ::= "" shortstringitem* "" | "" shortstringitem* ""

longstring ::= "" longstringitem* "" | "" longstringitem* ""

shortstringitem ::= shortstringchar | stringescapeseq

longstringitem ::= longstringchar | stringescapeseq

shortstringchar ::= <any source character except "\" or newline or the quote>

longstringchar ::= <any source character except "\">

stringescapeseq ::= "\" <any source character>

bytesliteral ::= bytesprefix(shortbytes | longbytes)

bytesprefix ::= "b" | "B" | "br" | "Br" | "bR" | "BR" | "rb" | "rB" | "Rb" | "RB"

shortbytes ::= "" shortbytesitem* "" | "" shortbytesitem* ""

longbytes ::= "" longbytesitem* "" | "" longbytesitem* ""

shortbytesitem ::= shortbyteschar | bytesescapeseq

longbytesitem ::= longbyteschar | bytesescapeseq

shortbyteschar ::= <any ASCII character except "\" or newline or the quote>

longbyteschar ::= <any ASCII character except "\">

bytesescapeseq ::= "\" <any ASCII character>

Screenshots παραδειγμάτων εφαρμογής

Παραδείγματα με ανάθεση

```
-----ENTER HERE-----  
  
Enter a python expression:  
h = "Greetings"  
  
This is a valid python expression
```

Εικόνα 1: Ανάθεση_string

```
-----ENTER HERE-----  
  
Enter a python expression:  
x = 5  
  
This is a valid python expression
```

Εικόνα 2: Ανάθεση_integer

```
-----ENTER HERE-----  
  
Enter a python expression:  
x = 3+4  
  
This is a valid python expression
```

Εικόνα 3: Ανάθεση_sum

```
Enter a python expression:  
x = x+5.36  
  
This is a valid python expression
```

Εικόνα 4: Ανάθεση_float

Παραδείγματα με δεσμευμένες λέξεις

```
-----ENTER HERE-----  
  
Enter a python expression:  
def  
  
keyword can't be used as a identifier
```

Εικόνα 5: δεσμευμένη λέξη_def

```
-----ENTER HERE-----  
  
Enter a python expression:  
print  
  
keyword can't be used as a identifier
```

Εικόνα 6: δεσμευμένη λέξη_print

```
-----ENTER HERE-----  
  
Enter a python expression:  
return  
  
keyword can't be used as a identifier
```

Εικόνα 7: δεσμευμένη λέξη_return

Παραδείγματα με “dictionary”

```
Enter a python expression:  
dict={}

This is valid expression
```

Εικόνα 8: dictionary χωρίς όρισμα

```
Enter a python expression:  
dict={1:hi}

This is valid expression
```

Εικόνα 9: dictionary με 1 όρισμα

```
Enter a python expression:  
dict={1:a,2:b}

This is valid expression
```

Εικόνα 10: dictionary με 2 ορίσματα

```
Enter a python expression:  
dict={1:a,2:b,3:c}

This is valid expression
```

Εικόνα 11: dictionary με 3 ορίσματα

```
Enter a python expression:  
dict={1:a,2:b,3:c,4:d}

This is valid expression
```

Εικόνα 12: dictionary με 4 ορίσματα

Παραδείγματα με “for statement”

```
-----ENTER HERE-----

Enter a python expression:  
for i in temp:

This is valid expression
```

Εικόνα 13: for stmt

```
-----ENTER HERE-----

Enter a python expression:  
for i in range(1,100):

This is valid expression
```

Εικόνα 14: for stmt with range

Παραδείγματα με “if statement”

```
-----ENTER HERE-----  
  
Enter a python expression:  
if a == b:  
  
This is valid expression
```

Εικόνα 15: if_stmt_EQ

```
Enter a python expression:  
if a > b:  
  
This is valid expression
```

Εικόνα 16: if_stmt_GT

```
-----ENTER HERE-----  
  
Enter a python expression:  
if aa >= BB:  
  
This is valid expression
```

Εικόνα 17: if_stmt_GE

```
-----ENTER HERE-----  
  
Enter a python expression:  
if a<b:  
  
This is valid expression  
-----
```

Εικόνα 18: if_stmt_LT

```
-----ENTER HERE-----  
  
Enter a python expression:  
if a <= b:  
  
This is valid expression  
-----
```

Εικόνα 19: if_stmt_LE

Παραδείγματα με “import”

```
-----ENTER HERE-----  
  
Enter a python expression:  
import mymodule  
  
This is valid expression
```

Εικόνα 20: import_I

```
-----ENTER HERE-----  
  
Enter a python expression:  
import mymodule as mx  
  
This is valid expression
```

Εικόνα 21: import_II

Παραδείγματα με τον επικείμενο constructor

```
-----ENTER HERE-----  
  
Enter a python expression:  
def __init__():  
  
This is valid expression
```

Εικόνα 22: constructor χωρίς ορίσματα

```
-----ENTER HERE-----  
  
Enter a python expression:  
def __init__(f,d):  
  
This is valid expression  
-----
```

Εικόνα 23: constructor με ορίσματα

Παραδείγματα με “Lambda Calculus”

```
-----ENTER HERE-----  
  
Enter a python expression:  
lambda x:x  
  
This is valid expression
```

Εικόνα 24: Lambda_Calculus_I

```
-----ENTER HERE-----  
  
Enter a python expression:  
lambda x:y+1  
  
This is valid expression
```

Εικόνα 25: Lambda_Calculus_II

```
-----ENTER HERE-----  
  
Enter a python expression:  
lambda x:y/z  
  
This is valid expression
```

Εικόνα 26: Lambda_Calculus_III

```
-----ENTER HERE-----  
  
Enter a python expression:  
lambda x,y:2*5  
  
This is valid expression
```

Εικόνα 27: Lambda_Calculus_IV

Παραδείγματα με “object”

```
-----ENTER HERE-----  
  
Enter a python expression:  
obj = Greetings()  
  
This is valid expression
```

Εικόνα 28: δημιουργία αντικειμένου

```
-----ENTER HERE-----  
  
Enter a python expression:  
obj.temp()  
  
This is valid expression  
  
-----
```

Εικόνα 29: κάλεσμα αντικειμένου

Παραδείγματα με “print”

```
-----ENTER HERE-----  
  
Enter a python expression:  
print(4/5)  
  
This is a valid python expression  
  
-----
```

Εικόνα 30: print_I

```
-----ENTER HERE-----  
  
Enter a python expression:  
print("Hello")  
  
This is a valid python expression  
  
-----
```

Εικόνα 31: print_II

```
-----ENTER HERE-----  
  
Enter a python expression:  
print(variable)  
  
This is a valid python expression  
  
-----
```

Εικόνα 32: print_III

Παραδείγματα με συνάρτηση

```
-----ENTER HERE-----  
  
Enter a python expression:  
def temp():  
  
This is valid expression  
-----
```

Εικόνα 33: συνάρτηση χωρίς ορίσματα

```
-----ENTER HERE-----  
  
Enter a python expression:  
def temp(a):  
  
This is valid expression  
-----
```

Εικόνα 34: συνάρτηση με όρισμα

```
-----ENTER HERE-----  
  
Enter a python expression:  
def temp(a,b,c,d,e,f,g):  
  
This is valid expression  
-----
```

Εικόνα 35: συνάρτηση με ορίσματα

Παραδείγματα με σχόλια

```
-----ENTER HERE-----  
  
Enter a python expression:  
#  
  
This is valid expression  
-----
```

Εικόνα 36: σχόλιο_I

```
-----ENTER HERE-----  
  
Enter a python expression:  
# temp  
  
This is valid expression  
-----
```

Εικόνα 37: σχόλιο_II