

## Πολυτεχνική Σχολή Τμήμα Μηχανικών Η/Υ & Πληροφορικής

# Αναφορά Project

### Μέλη ομάδας

Επώνυμο - Όνομα	AM	Έτος
Βακαλόπουλος Δημήτριος	1059564	30
Τάσσιος Βλάσιος	1047178	50
Τριμίντζιος Ηλίας	1047200	50
Τσουβάλας Άγγελος	1047049	50

## Πίνακας περιεχομένων

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

## 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_Ι	18
Εικόνα 37: σχόλιο ΙΙ	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:

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

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

assign\_arithmetic: identifier EQ expr

Παρακάτω γίνεται αριθμητική ανάθεση:

#### 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

#### 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

#### 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+
                  ::= "@" dotted_name ["(" [argument_list [","]] ")"] NEWLINE
decorator
                     ::= identifier ("." identifier)*
dotted_name
                     ::= defparameter ("," defparameter)* "," "/" ["," [parameter_list_no_posonly]]
parameter_list
                  | parameter list no posonly
parameter_list_no_posonly ::= defparameter ("," defparameter)* ["," [parameter_list_starargs]]
                 | parameter_list_starargs
parameter_list_starargs ::= "*" [parameter] ("," defparameter)* ["," ["**" parameter [","]]]
                | "**" parameter [","]
                   ::= identifier [":" expression]
parameter
                   ::= parameter ["=" expression]
defparameter
                   ::= identifier
funcname
```

#### If statement

#### For statement

#### Υποστήριξη εμφάνισης μηνυμάτων

```
<simple_stmt>:=<expression_stmt>|<assert_stmt>|<assignment_stmt>|<augmented_assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<assignment_stmt>|<ass
```

#### 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

#### Global statement

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

#### Try statement

#### 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>
```

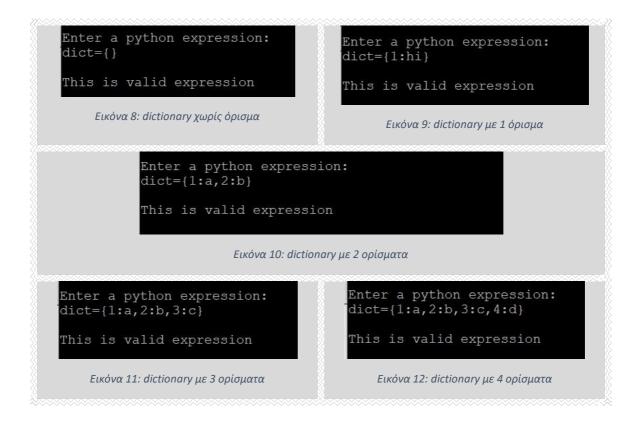
### Screenshots παραδειγμάτων εφαρμογής

#### Παραδείγματα με ανάθεση

---ENTER HERE--------ENTER HERE-----Enter a python expression: Enter a python expression: h = "Greetings" x = 5This is a valid python expression This is a valid python expression Εικόνα 2: Ανάθεση \_integer Εικόνα 1: Ανάθεση\_string ----ENTER HERE----Enter a python expression: x = x+5.36Enter a python expression: This is a valid python expression This is a valid python expression Εικόνα 3: Ανάθεση\_sum Εικόνα 4: Ανάθεση\_float

#### Παραδείγματα με δεσμευμένες λέξεις

#### Παραδείγματα με "dictionary"



#### Παραδείγματα με "for statement"

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

Enter a python expression:
for i in temp:

This is valid expression

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

This is valid expression

Eικόνα 13: for stmt
```

## 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:	ENTER HERE	
This is valid expression	Enter a python expression:  if aa >= BB:	
Εικόνα 16: if_ stmt_GT	This is valid expression	
	Εικόνα 17: if_ stmt_GE	
ENTER HERE	ENTER HERE	
Enter a python expression: if a <b:< td=""><td>Enter a python expression: if a &lt;= b:</td></b:<>	Enter a python expression: if a <= b:	
This is valid expression	This is valid expression	
Εικόνα 18: if_ stmt_LT	Εικόνα 19: if_ stmt_LE	

## Παραδείγματα με "import"

ENTER HERE	ENTER HERE
Enter a python expression: import mymodule	Enter a python expression: import mymodule as mx
This is valid expression	This is valid expression
Εικόνα 20: import_I	Εικόνα 21: import_II

#### Παραδείγματα με τον επικείμενο constructor

```
Enter a python expression:
def __init__():

This is valid expression

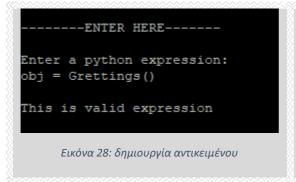
Eικόνα 22: constructor χωρίς ορίσματα

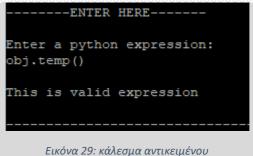
Eκόνα 23: constructor με ορίσματα
```

#### Παραδείγματα με "Lambda Calculus"



#### Παραδείγματα με "object"





### Παραδείγματα με "print"



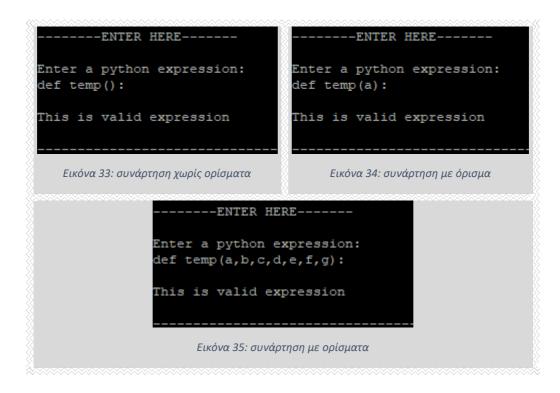
Εικόνα 30: print\_I

----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



#### Παραδείγματα με σχόλια

