

ΟΝΟΜΑ	ΑΜ	ΕΤΟΣ	email
Χαράλαμπος Κωνσταντακόπουλος	1090059	6 <sup>ο</sup>	up1090059@ac.upatras.gr

## Περιεχόμενα

Περιγραφή της γραμματικής της γλώσσας σε BNF.....	2
Τα ΤΕΛΙΚΑ αρχεία εισόδου για το flex και το bison .....	9
Screenshots παραδειγμάτων εφαρμογής .....	23
Σχόλια.....	24

## Περιγραφή της γραμματικής της γλώσσας σε BNF

```
<program> ::= MYHTML_OPEN <head_opt> BODY_OPEN <body_opt> BODY_CLOSE MYHTML_CLOSE

<head_opt> ::= <head_block>
              | ε

<head_block> ::= HEAD_OPEN TITLE_OPEN <text> TITLE_CLOSE <meta_list> HEAD_CLOSE

<meta_list> ::= <meta_list><meta_block>
               | ε

<meta_block> ::= META <meta_attr_list> MEGALYTERO

<meta_attr_list> ::= CHARSET ISON STRING | NAME ISON STRING CONTENT ISON STRING

<body_opt> ::= <body_opt><body_elements>
              | ε

<body_elements> ::=
    <p_block>
    | <div_block>
    | <a_block>
    | <form_block>
    | <img_block>

<p_block> ::= P_OPEN <p_attr_list> MEGALYTERO <p_content> P_CLOSE

<p_attr_list> ::=
    ID ISON STRING
    | ID ISON STRING STYLE ISON STRING
```

```

        | STYLE ISON STRING ID ISON STRING

<p_content> ::= <text>
        | ε

<div_block> ::= DIV_OPEN <div_attr_list> MEGALYTERO <div_opt> DIV_CLOSE

<div_attr_list> ::=
        ID <K>
        | ID <K> STYLE <K>
        | STYLE <K> ID <K>

<div_opt> ::=
        <div_opt><div_elements>
        | <div_elements>

<div_elements> ::=
        <p_block>
        | <a_block>
        | <form_block>
        | <img_block>

<a_block> ::= A_OPEN <a_attr_list> MEGALYTERO <a_opt> A_CLOSE;

<a_attr_list> ::= ID ISON STRING HREF ISON STRING
        | HREF ISON STRING ID ISON STRING

<a_opt> ::=
        <text>
        | <img_block>
        | <text><img_block>
        | ε

```







```

    | HEIGHT <I> SRC <K> ALT <K> ID <K> | SRC <K> HEIGHT <I> ALT <K> ID <K> | SRC <K>
ALT <K> HEIGHT <I> ID <K> | SRC <K> ALT <K> ID <K> HEIGHT <I>
    | HEIGHT <I> SRC <K> ID <K> ALT <K> | SRC <K> HEIGHT <I> ID <K> ALT <K> | SRC <K>
ID <K> HEIGHT <I> ALT <K> | SRC <K> ID <K> ALT <K> HEIGHT <I>
    | HEIGHT <I> ALT <K> SRC <K> ID <K> | ALT <K> HEIGHT <I> SRC <K> ID <K> | ALT <K>
SRC <K> HEIGHT <I> ID <K> | ALT <K> SRC <K> ID <K> HEIGHT <I>
    | HEIGHT <I> ALT <K> ID <K> SRC <K> | ALT <K> HEIGHT <I> ID <K> SRC <K> | ALT <K>
ID <K> HEIGHT <I> SRC <K> | ALT <K> ID <K> SRC <K> HEIGHT <I>
    | ID <K> SRC <K> ALT <K> | ID <K> ALT <K> SRC <K>
    | SRC <K> ALT <K> ID <K> | SRC <K> ID <K> ALT <K>
    | ALT <K> SRC <K> ID <K> | ALT <K> ID <K> SRC <K>

```

```

<I> ::= ISON INT

```

```

<K> ::= ISON STRING

```

```

<form_block> ::= FORM_OPEN <form_attr_list> MEGALYTERO <form_opt> FORM_CLOSE

```

```

<form_attr_list> ::= ID ISON STRING | ID ISON STRING STYLE ISON STRING | STYLE ISON
STRING ID ISON STRING

```

```

<form_opt> ::=
    <form_opt><form_elements>
    | <form_elements>

```

```

<form_elements> ::=
    <input_block>
    | <label_block>

```

```

<input_block> ::= INPUT <input_attr_list> MEGALYTERO

```

```

<input_attr_list> ::=

```

```

TYPE <K> ID <K> | ID <K> TYPE <K>
| VALUE <K> TYPE <K> ID <K> | TYPE <K> VALUE <K> ID <K> | TYPE <K> ID <K> VALUE
<K>
| VALUE <K> ID <K> TYPE <K> | ID <K> VALUE <K> TYPE <K> | ID <K> TYPE <K> VALUE
<K>
| STYLE <K> TYPE <K> ID <K> | TYPE <K> STYLE <K> ID <K> | TYPE <K> ID <K> STYLE
<K>
| STYLE <K> ID <K> TYPE <K> | ID <K> STYLE <K> TYPE <K> | ID <K> TYPE <K> STYLE
<K>
| VALUE <K> STYLE <K> TYPE <K> ID <K> | STYLE <K> VALUE <K> TYPE <K> ID <K> |
STYLE <K> TYPE <K> VALUE <K> ID <K> | STYLE <K> TYPE <K> ID <K> VALUE <K>
| VALUE <K> TYPE <K> STYLE <K> ID <K> | TYPE <K> VALUE <K> STYLE <K> ID <K> |
TYPE <K> STYLE <K> VALUE <K> ID <K> | TYPE <K> STYLE <K> ID <K> VALUE <K>
| VALUE <K> TYPE <K> ID <K> STYLE <K> | TYPE <K> VALUE <K> ID <K> STYLE <K> |
TYPE <K> ID <K> VALUE <K> STYLE <K> | TYPE <K> ID <K> STYLE <K> VALUE <K>
| VALUE <K> STYLE <K> ID <K> TYPE <K> | STYLE <K> VALUE <K> ID <K> TYPE <K> |
STYLE <K> ID <K> VALUE <K> TYPE <K> | STYLE <K> ID <K> TYPE <K> VALUE <K>
| VALUE <K> ID <K> STYLE <K> TYPE <K> | ID <K> VALUE <K> STYLE <K> TYPE <K> | ID
<K> STYLE <K> VALUE <K> TYPE <K> | ID <K> STYLE <K> TYPE <K> VALUE <K>
| VALUE <K> ID <K> TYPE <K> STYLE <K> | ID <K> VALUE <K> TYPE <K> STYLE <K> | ID
<K> TYPE <K> VALUE <K> STYLE <K> | ID <K> TYPE <K> STYLE <K> VALUE <K>

<label_block> ::= LABEL_OPEN <lbl_attr_list> MEGALYTERO <text> LABEL_CLOSE;

<lbl_attr_list> ::=
    ID <K> FOR <K> | FOR <K> ID <K>
    | STYLE <K> ID <K> FOR <K> | ID <K> STYLE <K> FOR <K> | ID <K> FOR <K> STYLE <K>
    | STYLE <K> FOR <K> ID <K> | FOR <K> STYLE <K> ID <K> | FOR <K> ID <K> STYLE <K>

<text> ::=
    <text> TEXT
    | TEXT

```



## Τα ΤΕΛΙΚΑ αρχεία εισόδου για το flex και το bison

Lexer.l

```
%{  
    #include "parser.tab.h" //σύνδεση με το bison  
    #include <stdio.h>  
%}  
  
/*leme ston lexer na metraei automata tis grammes stin metavliti yylineno */  
%option yylineno  
  
/* orizoume thn katastash TAG */  
%x TAG  
/* orizoume thn katastash COMMENT */  
%x COMMENT  
  
%%  
  
"--" {printf("%s", yytext); BEGIN COMMENT;}  
  
"<MYHTML>" {printf("%s", yytext); return MYHTML_OPEN; }  
"</MYHTML>" {printf("%s", yytext); return MYHTML_CLOSE; }  
  
"<head>" {printf("%s", yytext); return HEAD_OPEN; }  
"</head>" {printf("%s", yytext); return HEAD_CLOSE; }  
  
"<title>" {printf("%s", yytext); return TITLE_OPEN; }  
"</title>" {printf("%s", yytext); return TITLE_CLOSE; }  
  
"<body>" { printf("%s", yytext); return BODY_OPEN; }  
"</body>" { printf("%s", yytext); return BODY_CLOSE; }
```

```

"<p" {printf("%s", yytext); BEGIN(TAG); return P_OPEN; } //me to begin tag ksekinaw
thn kataastash TAG

"</p>" {printf("%s", yytext); return P_CLOSE;}

"<div" {printf("%s", yytext); BEGIN(TAG); return DIV_OPEN; }

"</div>" {printf("%s", yytext); return DIV_CLOSE;}

"<a" {printf("%s", yytext); BEGIN(TAG); return A_OPEN; }

"</a>" {printf("%s", yytext); return A_CLOSE;}

"<form" {printf("%s", yytext); BEGIN(TAG); return FORM_OPEN; }

"</form>" {printf("%s", yytext); return FORM_CLOSE;}

"<label" {printf("%s", yytext); BEGIN(TAG); return LABEL_OPEN; }

"</label>" {printf("%s", yytext); return LABEL_CLOSE;}

"<meta" {printf("%s", yytext); BEGIN(TAG); return META;}

"<img" {printf("%s", yytext); BEGIN(TAG); return IMAGE;}

"<input" {printf("%s", yytext); BEGIN(TAG); return INPUT;} // TAG rules next

<COMMENT>"-->" {printf("%s", yytext); BEGIN INITIAL;}

<COMMENT>"--" {fflush(stdout);fprintf(stderr, "\n\nError line %d: Comments cannot
contain double hyphens (--)\n", yylineno); exit(1);}

<COMMENT>\n {printf("%s", yytext);}

<COMMENT>. {printf("%s", yytext);}

<TAG>">" {printf("%s", yytext); BEGIN INITIAL; return MEGALYTERO;} //vrikame > ara
teleiwsa me ta attributes

<TAG>"id" {printf("%s", yytext); return ID;}

<TAG>"href" {printf("%s", yytext); return HREF;}

```

```

<TAG>"src" {printf("%s", yytext); return SRC;}
<TAG>"alt" {printf("%s", yytext); return ALT;}
<TAG>"style" {printf("%s", yytext); return STYLE;}
<TAG>"type" {printf("%s", yytext); return TYPE;}
<TAG>"value" {printf("%s", yytext); return VALUE;}
<TAG>"for" {printf("%s", yytext); return FOR;}
<TAG>"width" {printf("%s", yytext); return WIDTH;}
<TAG>"height" {printf("%s", yytext); return HEIGHT;}
<TAG>"name" {printf("%s", yytext); return NAME;}
<TAG>"content" {printf("%s", yytext); return CONTENT;}
<TAG>"charset" {printf("%s", yytext); return CHARSET;}

<TAG>"=" {printf("%s", yytext); return ISON;}

<TAG>"<--" {fflush(stdout);fprintf(stderr, "\nError line %d: Comments are not allowed
inside tags\n", yylineno); exit(1); }

<TAG>"[1-9][0-9]*" { printf("%s", yytext); return INT; }

<TAG>"'[1-9][0-9]*'" { printf("%s", yytext); return INT; }

<TAG>"\"[^\"]*" {printf("%s", yytext); return STRING;} //anagnwrisi alparithmitikwn

<TAG>"\'[^\']*'" { printf("%s", yytext); return STRING; }

<TAG>"[ \t\n]+" { printf("%s", yytext); } //agnwoume kena mesa sto tag

["^<>\n\t\r"]+ { printf("%s", yytext); return TEXT; }
[" \t\n\r"] { printf("%s", yytext); }
. { printf("%s", yytext); }

```

```
%%
```

```
int yywrap() { return 1; }
```

## Parser.y

```
%{
```

```
    #include <stdio.h>
```

```
    #include <stdlib.h>
```

```
    void yyerror(const char *s);
```

```
    int yylex();
```

```
    extern FILE *yyin;
```

```
    extern int yylineno; //pairnw ton metrhth grammwn apo to flex
```

```
%}
```

```
/*dhlwnoume ta tokens pou mas stelnei o lexer */
```

```
%token MYHTML_OPEN MYHTML_CLOSE
```

```
%token HEAD_OPEN HEAD_CLOSE
```

```
%token TITLE_OPEN TITLE_CLOSE
```

```
%token META
```

```
%token BODY_OPEN BODY_CLOSE
```

```
%token P_OPEN P_CLOSE
```

```
%token DIV_OPEN DIV_CLOSE
```

```
%token A_OPEN A_CLOSE
```

```
%token IMAGE
```

```
%token FORM_OPEN FORM_CLOSE
```

```
%token LABEL_OPEN LABEL_CLOSE
```

```

%token INPUT
%token INT
%token STRING
%token TEXT

/* tokens gia ta attributes */
%token MEGALYTERO /* > */
%token ISON /* = */

%token ID CHARSET NAME CONTENT STYLE HREF SRC ALT WIDTH HEIGHT VALUE TYPE FOR

%%

/* BNF GRAMMAR */
program:
    MYHTML_OPEN head_opt BODY_OPEN body_opt BODY_CLOSE MYHTML_CLOSE
    {
        printf("\n\n**Επιτυχία!**\n");
    }
;

head_opt:
    head_block
    |
;

head_block:
    HEAD_OPEN TITLE_OPEN text TITLE_CLOSE meta_list HEAD_CLOSE
;

meta_list:
    meta_list meta_block

```

```

|
;

meta_block:
    META meta_attr_list MEGALYTERO
;

meta_attr_list:
    CHARSET ISON STRING | NAME ISON STRING CONTENT ISON STRING
;

body_opt:
    body_opt body_elements
|
;

body_elements:
    p_block
|div_block
|a_block
|form_block
|img_block
;

p_block:
    P_OPEN p_attr_list MEGALYTERO p_content P_CLOSE
;

p_attr_list:
    ID ISON STRING | ID ISON STRING STYLE ISON STRING | STYLE ISON STRING ID ISON
STRING
;

```

```

p_content:
    text | ;

div_block:
    DIV_OPEN div_attr_list MEGALYTERO div_opt DIV_CLOSE;

div_attr_list:
    ID K | ID K STYLE K | STYLE K ID K
;

div_opt:
    div_opt div_elements | div_elements
;

div_elements:
    p_block
    |a_block
    |form_block
    |img_block
;

a_block:
    A_OPEN a_attr_list MEGALYTERO a_opt A_CLOSE;

a_attr_list:
    ID ISON STRING HREF ISON STRING | HREF ISON STRING ID ISON STRING
;

a_opt:
    text
    | img_block

```

```

    | text img_block
    |
;

img_block:
    IMAGE img_attr_list MEGALYTERO
;

img_attr_list:
    HEIGHT I WIDTH I ID K SRC K ALT K | WIDTH I HEIGHT I ID K SRC K ALT K | WIDTH I ID
K HEIGHT I SRC K ALT K | WIDTH I ID K SRC K HEIGHT I ALT K | WIDTH I ID K SRC K ALT K
HEIGHT I
    | HEIGHT I ID K WIDTH I SRC K ALT K | ID K HEIGHT I WIDTH I SRC K ALT K | ID K
WIDTH I HEIGHT I SRC K ALT K | ID K WIDTH I SRC K HEIGHT I ALT K | ID K WIDTH I SRC K
ALT K HEIGHT I
    | HEIGHT I ID K SRC K WIDTH I ALT K | ID K HEIGHT I SRC K WIDTH I ALT K | ID K SRC
K HEIGHT I WIDTH I ALT K | ID K SRC K WIDTH I HEIGHT I ALT K | ID K SRC K WIDTH I ALT
K HEIGHT I
    | HEIGHT I ID K SRC K ALT K WIDTH I | ID K HEIGHT I SRC K ALT K WIDTH I | ID K SRC
K HEIGHT I ALT K WIDTH I | ID K SRC K ALT K HEIGHT I WIDTH I | ID K SRC K ALT K WIDTH
I HEIGHT I
    | HEIGHT I WIDTH I ID K ALT K SRC K | WIDTH I HEIGHT I ID K ALT K SRC K | WIDTH I
ID K HEIGHT I ALT K SRC K | WIDTH I ID K ALT K HEIGHT I SRC K | WIDTH I ID K ALT K
SRC K HEIGHT I
    | HEIGHT I ID K WIDTH I ALT K SRC K | ID K HEIGHT I WIDTH I ALT K SRC K | ID K
WIDTH I HEIGHT I ALT K SRC K | ID K WIDTH I ALT K HEIGHT I SRC K | ID K WIDTH I ALT K
SRC K HEIGHT I
    | HEIGHT I ID K ALT K WIDTH I SRC K | ID K HEIGHT I ALT K WIDTH I SRC K | ID K ALT
K HEIGHT I WIDTH I SRC K | ID K ALT K WIDTH I HEIGHT I SRC K | ID K ALT K WIDTH I SRC
K HEIGHT I

```



| HEIGHT I ID K ALT K SRC K WIDTH I | ID K HEIGHT I ALT K SRC K WIDTH I | ID K ALT  
K HEIGHT I SRC K WIDTH I | ID K ALT K SRC K HEIGHT I WIDTH I | ID K ALT K SRC K WIDTH  
I HEIGHT I

| HEIGHT I WIDTH I SRC K ID K ALT K | WIDTH I HEIGHT I SRC K ID K ALT K | WIDTH I  
SRC K HEIGHT I ID K ALT K | WIDTH I SRC K ID K HEIGHT I ALT K | WIDTH I SRC K ID K  
ALT K HEIGHT I

| HEIGHT I SRC K WIDTH I ID K ALT K | SRC K HEIGHT I WIDTH I ID K ALT K | SRC K  
WIDTH I HEIGHT I ID K ALT K | SRC K WIDTH I ID K HEIGHT I ALT K | SRC K WIDTH I ID K  
ALT K HEIGHT I

| HEIGHT I SRC K ID K WIDTH I ALT K | SRC K HEIGHT I ID K WIDTH I ALT K | SRC K ID  
K HEIGHT I WIDTH I ALT K | SRC K ID K WIDTH I HEIGHT I ALT K | SRC K ID K WIDTH I ALT  
K HEIGHT I

| HEIGHT I SRC K ID K ALT K WIDTH I | SRC K HEIGHT I ID K ALT K WIDTH I | SRC K ID  
K HEIGHT I ALT K WIDTH I | SRC K ID K ALT K HEIGHT I WIDTH I | SRC K ID K ALT K WIDTH  
I HEIGHT I

| HEIGHT I WIDTH I SRC K ALT K ID K | WIDTH I HEIGHT I SRC K ALT K ID K | WIDTH I  
SRC K HEIGHT I ALT K ID K | WIDTH I SRC K ALT K HEIGHT I ID K | WIDTH I SRC K ALT K  
ID K HEIGHT I

| HEIGHT I SRC K WIDTH I ALT K ID K | SRC K HEIGHT I WIDTH I ALT K ID K | SRC K  
WIDTH I HEIGHT I ALT K ID K | SRC K WIDTH I ALT K HEIGHT I ID K | SRC K WIDTH I ALT K  
ID K HEIGHT I

| HEIGHT I SRC K ALT K WIDTH I ID K | SRC K HEIGHT I ALT K WIDTH I ID K | SRC K  
ALT K HEIGHT I WIDTH I ID K | SRC K ALT K WIDTH I HEIGHT I ID K | SRC K ALT K WIDTH I  
ID K HEIGHT I

| HEIGHT I SRC K ALT K ID K WIDTH I | SRC K HEIGHT I ALT K ID K WIDTH I | SRC K  
ALT K HEIGHT I ID K WIDTH I | SRC K ALT K ID K HEIGHT I WIDTH I | SRC K ALT K ID K  
WIDTH I HEIGHT I

| HEIGHT I WIDTH I ALT K SRC K ID K | WIDTH I HEIGHT I ALT K SRC K ID K | WIDTH I  
ALT K HEIGHT I SRC K ID K | WIDTH I ALT K SRC K HEIGHT I ID K | WIDTH I ALT K SRC K  
ID K HEIGHT I

| HEIGHT I ALT K WIDTH I SRC K ID K | ALT K HEIGHT I WIDTH I SRC K ID K | ALT K  
WIDTH I HEIGHT I SRC K ID K | ALT K WIDTH I SRC K HEIGHT I ID K | ALT K WIDTH I SRC K  
ID K HEIGHT I

| HEIGHT I ALT K SRC K WIDTH I ID K | ALT K HEIGHT I SRC K WIDTH I ID K | ALT K  
SRC K HEIGHT I WIDTH I ID K | ALT K SRC K WIDTH I HEIGHT I ID K | ALT K SRC K WIDTH I  
ID K HEIGHT I

| HEIGHT I ALT K SRC K ID K WIDTH I | ALT K HEIGHT I SRC K ID K WIDTH I | ALT K  
SRC K HEIGHT I ID K WIDTH I | ALT K SRC K ID K HEIGHT I WIDTH I | ALT K SRC K ID K  
WIDTH I HEIGHT I

| HEIGHT I WIDTH I ALT K ID K SRC K | WIDTH I HEIGHT I ALT K ID K SRC K | WIDTH I  
ALT K HEIGHT I ID K SRC K | WIDTH I ALT K ID K HEIGHT I SRC K | WIDTH I ALT K ID K  
SRC K HEIGHT I

| HEIGHT I ALT K WIDTH I ID K SRC K | ALT K HEIGHT I WIDTH I ID K SRC K | ALT K  
WIDTH I HEIGHT I ID K SRC K | ALT K WIDTH I ID K HEIGHT I SRC K | ALT K WIDTH I ID K  
SRC K HEIGHT I

| HEIGHT I ALT K ID K WIDTH I SRC K | ALT K HEIGHT I ID K WIDTH I SRC K | ALT K ID  
K HEIGHT I WIDTH I SRC K | ALT K ID K WIDTH I HEIGHT I SRC K | ALT K ID K WIDTH I SRC  
K HEIGHT I

| HEIGHT I ALT K ID K SRC K WIDTH I | ALT K HEIGHT I ID K SRC K WIDTH I | ALT K ID  
K HEIGHT I SRC K WIDTH I | ALT K ID K SRC K HEIGHT I WIDTH I | ALT K ID K SRC K WIDTH  
I HEIGHT I

| WIDTH I ID K SRC K ALT K | ID K WIDTH I SRC K ALT K | ID K SRC K WIDTH I ALT K |  
ID K SRC K ALT K WIDTH I

| WIDTH I ID K ALT K SRC K | ID K WIDTH I ALT K SRC K | ID K ALT K WIDTH I SRC K |  
ID K ALT K SRC K WIDTH I

| WIDTH I SRC K ALT K ID K | SRC K WIDTH I ALT K ID K | SRC K ALT K WIDTH I ID K |  
SRC K ALT K ID K WIDTH I

| WIDTH I SRC K ID K ALT K | SRC K WIDTH I ID K ALT K | SRC K ID K WIDTH I ALT K |  
SRC K ID K ALT K WIDTH I

| WIDTH I ALT K SRC K ID K | ALT K WIDTH I SRC K ID K | ALT K SRC K WIDTH I ID K |  
ALT K SRC K ID K WIDTH I

```

    | WIDTH I ALT K ID K SRC K | ALT K WIDTH I ID K SRC K | ALT K ID K WIDTH I SRC K |
ALT K ID K SRC K WIDTH I
    | HEIGHT I ID K SRC K ALT K | ID K HEIGHT I SRC K ALT K | ID K SRC K HEIGHT I ALT
K | ID K SRC K ALT K HEIGHT I
    | HEIGHT I ID K ALT K SRC K | ID K HEIGHT I ALT K SRC K | ID K ALT K HEIGHT I SRC
K | ID K ALT K SRC K HEIGHT I
    | HEIGHT I SRC K ALT K ID K | SRC K HEIGHT I ALT K ID K | SRC K ALT K HEIGHT I ID
K | SRC K ALT K ID K HEIGHT I
    | HEIGHT I SRC K ID K ALT K | SRC K HEIGHT I ID K ALT K | SRC K ID K HEIGHT I ALT
K | SRC K ID K ALT K HEIGHT I
    | HEIGHT I ALT K SRC K ID K | ALT K HEIGHT I SRC K ID K | ALT K SRC K HEIGHT I ID
K | ALT K SRC K ID K HEIGHT I
    | HEIGHT I ALT K ID K SRC K | ALT K HEIGHT I ID K SRC K | ALT K ID K HEIGHT I SRC
K | ALT K ID K SRC K HEIGHT I
    | ID K SRC K ALT K | ID K ALT K SRC K
    | SRC K ALT K ID K | SRC K ID K ALT K
    | ALT K SRC K ID K | ALT K ID K SRC K
;

I:
    ISON INT
;

K:
    ISON STRING
;

form_block:
    FORM_OPEN form_attr_list MEGALYTERO form_opt FORM_CLOSE;

form_attr_list:

```

```

    ID ISON STRING | ID ISON STRING STYLE ISON STRING | STYLE ISON STRING ID ISON
STRING
;

form_opt:
    form_opt form_elements
    | form_elements
;

form_elements:
    input_block
    | label_block
;

input_block:
    INPUT input_attr_list MEGALYTERO
;

input_attr_list:
    TYPE K ID K | ID K TYPE K
    | VALUE K TYPE K ID K | TYPE K VALUE K ID K | TYPE K ID K VALUE K
    | VALUE K ID K TYPE K | ID K VALUE K TYPE K | ID K TYPE K VALUE K
    | STYLE K TYPE K ID K | TYPE K STYLE K ID K | TYPE K ID K STYLE K
    | STYLE K ID K TYPE K | ID K STYLE K TYPE K | ID K TYPE K STYLE K
    | VALUE K STYLE K TYPE K ID K | STYLE K VALUE K TYPE K ID K | STYLE K TYPE K
VALUE K ID K | STYLE K TYPE K ID K VALUE K
    | VALUE K TYPE K STYLE K ID K | TYPE K VALUE K STYLE K ID K | TYPE K STYLE K
VALUE K ID K | TYPE K STYLE K ID K VALUE K
    | VALUE K TYPE K ID K STYLE K | TYPE K VALUE K ID K STYLE K | TYPE K ID K VALUE K
STYLE K | TYPE K ID K STYLE K VALUE K
    | VALUE K STYLE K ID K TYPE K | STYLE K VALUE K ID K TYPE K | STYLE K ID K VALUE
K TYPE K | STYLE K ID K TYPE K VALUE K

```

```

        | VALUE K ID K STYLE K TYPE K | ID K VALUE K STYLE K TYPE K | ID K STYLE K VALUE
K TYPE K | ID K STYLE K TYPE K VALUE K

        | VALUE K ID K TYPE K STYLE K | ID K VALUE K TYPE K STYLE K | ID K TYPE K VALUE K
STYLE K | ID K TYPE K STYLE K VALUE K
;

label_block:
    LABEL_OPEN lbl_attr_list MEGALYTERO text LABEL_CLOSE;

lbl_attr_list:
    ID K FOR K | FOR K ID K

    | STYLE K ID K FOR K | ID K STYLE K FOR K | ID K FOR K STYLE K

    | STYLE K FOR K ID K | FOR K STYLE K ID K | FOR K ID K STYLE K
;

text:
    text TEXT

    | TEXT
;

%%

int main(int argc, char **argv){

    //elegxos an dothike arxeio ws orisma
    if(argc>1){
        //anoigma tou arxeiou pou dwsame
        if(!(yyin = fopen(argv[1], "r"))){
            perror("Error opening file");
            return 1;
        }
    }
}

```

```
/* to yyparse tha diavasei apo to yyin (to arxeio eisodou)*/
yyparse();
return 0;
}

void yyerror(const char *s){
    /*αδειάζουμε τον μπαφέρ της πριντφ για να εμφανιστούν όλα τα ταγκς που βρηκαμε
    μεχρι τώρα σωστα*/
    fflush(stdout);

    /*τυπωνουμε το σφαλμα και τη γραμμη */
    fprintf(stderr, "\n\n***Σφαλμα στην γραμμη %d:%s ***\n", yylineno, s );

    /*διακόπτουμε την αναλυση οπως μας ζηταει η εκφωνηση */
    exit(1);
}
```

## Screenshots παραδειγμάτων εφαρμογής

παραδειγμα επιτυχίας

```
Haris@Haris /cygdrive/c/Users/Haris/Desktop/code/flex-bison
$ ./myParser.exe successful_test.txt
<MYHTML>
<head>
  <title>Principles of Programming Languages & Compilers</title>
  <meta name="description" content="Flex Bison Tutorial">
  <meta charset="UTF-8">
  <meta name='author' content="Input your name here">
</head>
<body>
  <p id='par1'>This is my project.</p>

  <div id="kostas 'div1'" style="color:blue">
    <a id="hyper" href="https://arxes.ceid.gr">
      Here comes an image. <-- kostas -->
      
    </a>
  </div>

  <form id="form1">
    <label for = "name" id="label1">Name:</label>
    <input type="text" id="name">
    <label id="label2" for="lastname">Last Name:</label>
    <input type="text" id="name">
    <input id="submit" type="submit" value='Submit "it" '>
  </form>

</body>
</MYHTML>

**Επιτυχία!**
```

παράδειγμα αποτυχίας (λείπει το χαρακτηριστικό id στο στοιχείο input στην 22<sup>η</sup> γραμμή)

```
Haris@Haris /cygdrive/c/Users/Haris/Desktop/code/flex-bison
$ ./myParser.exe unsuccessful_test.txt
<MYHTML>
<head>
  <title>Principles of Programming Languages & Compilers</title>
  <meta name="description" content="Flex Bison Tutorial">
  <meta charset="UTF-8">
  <meta name='author' content="Input your name here">
</head>
<body>
  <p id='par1'>This is my project.</p>

  <div id="kostas 'div1'" style="color:blue">
    <a id="hyper" href="https://arxes.ceid.gr">
      Here comes an image. <-- kostas -->
      
    </a>
  </div>

  <form id="form1">
    <label for = "name" id="label1">Name:</label>
    <input type="text" id="name">
    <label id="label2" for="lastname">Last Name:</label>
    <input type="text" >

***Σφάλμα στην γραμμή 22:syntax error ***
```

## Σχόλια

Υλοποιήθηκε μόνο το πρώτο κομμάτι της εργασίας δηλαδή Ερωτήματα 1.α. και 1.β. .

Σε περίπτωση που κάποιο αρχείο μέσα απ τον φάκελο zip δεν ανοίγει, τα έχω ανεβάσει και στο github.

[https://github.com/hariskonst/arxes\\_glwsswn\\_programmatismou](https://github.com/hariskonst/arxes_glwsswn_programmatismou)