

<http://ds9a.nl/lex-yacc/cvs/lexyacc.pdf>
<http://epaperpress.com/lexandyacc/download/lexyacc.pdf>
<http://user.it.uu.se/~matkin/programming/yacc+lex/>
<http://home.cogeco.ca/~ve3ll/linuxlex.htm>
<http://www.gnu.org/manual/flex-2.5.4/flex.html>

----- Create file lex.1 -----

```
%{  
#include <stdio.h>  
  
%}  
%%  
[0-9]+           printf("positive integer\n");  
[a-zA-Z][a-zA-Z0-9]*  printf("Identifier\n");  
%%
```

----- To generate the scanner -----

```
$ lex lex.1  
$ ls  
lex.1      lex.yy.c
```

lex.yy.c is the generated scanner. It is a C program. **Take a look at it!**
The main program is only 2 lines long, but the rest is a lot of hard to read stuff. We have to compile this C program:

-----To compile the scanner---

```
$ cc lex.yy.c -ll  
$ ls  
a.out      lex.1      lex.yy.c
```

a.out is the executable version of our scanner. Let's run it!

----- Running the scanner -----

```
$ ./a.out
```

```
23
```

```
positive integer
```

```
r2d2
```

```
Identifier
```

```
2rdr
```

```
positive integer
```

```
Identifier
```

```
==
```

```
==
```

Notice that it just echoes the “==” because we haven't defined what it is in the lex file.

-----Another way -----

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

```
%}
```

```
DIGIT    [0-9]
```

```
LETTER   [a-zA-Z]
```

```
%%
```

```
{DIGIT}+
```

```
printf("positive integer\n");
```

```
{LETTER} ( {LETTER} | {DIGIT})* printf("identifier\n");
```

```
%%
```

```
-----lex.10-----
```

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

```
%}
```

```
DIGIT [0-9]
```

```
LETTER [a-zA-Z]
```

```
%%
```

```
{DIGIT}+  
integer\n");
```

```
printf("positive
```

```
{LETTER}({LETTER}|{DIGIT})*
```

```
printf("identifier\n");
```

```
.
```

```
printf("other\n");
```

```
%%
```

```
$ rm a.out
```

```
$rm lex.yy.c
```

```
$lex lex.10
```

```
$cc lex.yy.c -ll
```

```
$/a.out
```

```
23
```

```
positive integer
```

```
r2d2
```

```
identifier
```

```
=
```

