🧩 一、最常用的特殊变量汇总

变量名	类型	含义	示例
`yytext`	`char*`	指向当前匹配到的**词法单元字符串**	若输入是 `abc+123`，匹配 `[a-zA-Z]+` 后，`yytext="abc"`
`yyleng`	`int`	当前匹配字符串 `yytext` 的**长度**	上例中 `yyleng = 3`
`yylval`	`YYSTYPE`（定义于 Bison）	若与 Bison 联用，用来**向语法分析器传递值**	比如将数字 token 的值赋给 `yylval.num = atoi(yytext)`
`yyin`	`FILE*`	指定**输入源文件指针**，默认是 `stdin`	可用 `yyin = fopen("test.txt","r");`
`yyout`	`FILE*`	指定**输出目标文件指针**，默认是 `stdout`	可重定向到文件输出结果
`yylineno`	`int`	当前**行号计数器**（可启用行号跟踪）	常用于错误提示，如 `printf("Line %d: error\n", yylineno);`
`yywrap()`	函数	Flex 在读到文件结尾时调用，用于决定是否继续读取其他输入文件	返回 1 表示结束，返回 0 表示继续读取
`yyterminate()`	宏	强制立即结束词法分析，相当于 `return 0;`	可在规则动作中调用以终止分析

🧭 二、核心变量详解

🔹 1. `yytext`

- 类型：`char *`
- 功能：指向当前匹配的文本（以 `\0` 结尾的字符串）。
- 在每次规则匹配成功后，Flex 会自动设置它。

示例：

```
[0-9]+   { printf("NUMBER: %s\n", yytext); }
[a-zA-Z]+ { printf("IDENT: %s\n", yytext); }
```

输入：

```
abc 123
```

输出：

```
IDENT: abc
NUMBER: 123
```

---

2. `yyleng`

- 类型：`int`
- 功能：当前匹配文本的长度。
- 常用于需要处理固定长度字符串的场合。

示例：

```
[a-zA-Z]+   { printf("%s 的长度是 %d\n", yytext, yyleng); }
```

输入：

```
hello
```

输出：

```
hello 的长度是 5
```

---

3. `yyin` 和 `yyout`

- 默认：`yyin = stdin`，`yyout = stdout`
- 可在 `main()` 中重定向输入输出流。

示例：

```
int main() {
    yyin = fopen("input.txt", "r");   // 从文件读取
    yyout = fopen("output.txt", "w"); // 输出到文件
    yylex();                          // 启动词法分析
    return 0;
}
```

---

4. `yylineno`

- 用于记录当前分析的行号（默认未启用）。
- 若想使用，需在 `.l` 文件头部加入：

  ```
  %option yylineno
  ```

- 然后在规则动作中可直接使用：

```
\n      { /* 自动增加 yylineno */ }
.       { printf("Line %d: %s\n", yylineno, yytext); }
```

---

5. `yylval`

- 当你用 **Flex + Bison** 组合时使用。
- 它是一个共享变量，用来将词法单元的**属性值（如数字的具体值）**传给语法分析器。

例如：

```
[0-9]+  { yylval.num = atoi(yytext); return NUMBER; }
```

这会把 `NUMBER` token 及其数值一起返回给 `yyparse()`。

---

6. `yywrap()`

- 在 `yylex()` 读到输入文件结尾时被调用。
- 若返回 `1`，表示分析结束；
- 若返回 `0`，Flex 会继续读取下一个文件。

通常定义如下：

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

或更简单地在 `.l` 文件开头加：

```
%option noyywrap
```

表示不需要自己写 `yywrap()` 函数。

---

7. `yyterminate()`

- 用于在词法规则中主动结束分析。
- 等价于在动作中写 `return 0;`。

例如：

```
"exit"  { yyterminate(); }
```

🧠 三、实践建议

✅ **最常用的组合是：**

```
yytext, yyleng, yylineno, yyin, yylval
```

✅ **调试技巧：**
在调试词法规则时，你可以直接在动作中打印：

```
.   { printf("Token: %-10s\tLength: %d\tLine: %d\n", yytext, yyleng, yylineno); }
```

✅ 总结表（快速记忆版）

变量名	功能	是否自动维护	是否常用
`yytext`	当前匹配字符串	✅	⭐⭐⭐⭐
`yyleng`	匹配字符串长度	✅	⭐⭐⭐
`yyin`	输入文件指针	✅	⭐⭐
`yyout`	输出文件指针	✅	⭐
`yylineno`	当前行号（需启用）	✅	⭐⭐⭐
`yylval`	与 Bison 传值用	✅	⭐⭐⭐⭐
`yywrap()`	文件结束时调用	🚫（需实现或禁用）	⭐⭐
`yyterminate()`	主动终止分析	✅	⭐