

# VB.NET زبان Scanner

## نکات کلی:

- در هیچ قسمت این زبان (به جز در داخل رشته ها و کاراکترها) تمایزی بین حروف بزرگ و کوچک نیست، مثلاً **If** و **if** و **IF** معادل محسوب می شوند.
- کلیه خطوط برنامه با یک **Enter** خاتمه می یابند و فرض کنید که طول خطوط از ۳۰۰ حرف بیشتر نخواهد بود.

## قالب Comment:

**Comment ::=** CommentMarker (~Enter)\*

CommentMarker ::= SingleQuote | **Rem**

البته اگر توضیح با کلمه **Rem** شروع شود بعد از آن حداقل یک حرف سفید برای جلوگیری از اتصال به عبارات بعدی باید وجود داشته باشد.

مثال:

```
Dim x as string 'this is a comment.
x = 2 ReM this is a comment.
x = 3Rem.
dim Rem4 as integer 'Rem4 is an identifier.
```

## قالب id ها:

**ID ::=** IdentifierName (ε | TypeCharacter) | EscapedIdentifier

EscapedIdentifier ::= [ IdentifierName ]

IdentifierName ::= IdentifierStart IdentifierCharacter\*

IdentifierStart ::= AlphaCharacter | Underscore IdentifierCharacter

IdentifierCharacter ::= Underscore | AlphaCharacter | NumericCharacter

AlphaCharacter ::= **a..z** | **A..Z**

NumericCharacter ::= **0..9**

Underscore ::= **\_**

TypeCharacter ::= IntegerTypeCharacter | LongTypeCharacter | DecimalTypeCharacter |

SingleTypeCharacter | DoubleTypeCharacter | StringTypeCharacter

IntegerTypeCharacter ::= **%**

LongTypeCharacter ::= **&**

DecimalTypeCharacter ::= **@**

SingleTypeCharacter ::= **!**

DoubleTypeCharacter ::= **#**

StringTypeCharacter ::= **\$**

مثال:

```
[class]          s4$          _a_          _____%          [x_]
```

**قالب اعداد:** اعداد به دو شکل صحیح و اعشاری هستند.

قالب عدد صحیح:

**INT ::=** IntegralLiteralValue (ε | IntegralTypeCharacter )

IntegralLiteralValue ::= IntLiteral | HexLiteral | OctalLiteral

IntegralTypeCharacter ::= ShortCharacter | IntegerCharacter | LongCharacter |  
 IntegerTypeCharacter | LongTypeCharacter  
 ShortCharacter ::= **S**  
 IntegerCharacter ::= **I**  
 LongCharacter ::= **L**  
 IntLiteral ::= Digit<sup>+</sup>  
 HexLiteral ::= **& H** HexDigit<sup>+</sup>  
 OctalLiteral ::= **& O** OctalDigit<sup>+</sup>  
 Digit ::= **0..9**  
 HexDigit ::= **0..9 | A..F**  
 OctalDigit ::= **0..7**

مثال:

0026L    &h1A    &o32S    →    همگی معادل با 26

قالب عدد اعشاری:

**FLT ::=**

FloatingPointLiteralValue (ε | FloatingPointTypeCharacter) |  
 IntLiteral FloatingPointTypeCharacter  
 FloatingPointTypeCharacter ::=  
 SingleCharacter | DoubleCharacter | DecimalCharacter |  
 SingleTypeCharacter | DoubleTypeCharacter | DecimalTypeCharacter  
 SingleCharacter ::= **F**  
 DoubleCharacter ::= **R**  
 DecimalCharacter ::= **D**  
 FloatingPointLiteralValue ::=  
 IntLiteral . IntLiteral (ε | Exponent) | . IntLiteral (ε | Exponent) | IntLiteral Exponent  
 Exponent ::= **E** (ε | Sign) IntLiteral  
 Sign ::= + | -

مثال:

.4    4.e-1    0.004e2d    00.400R    →    0.40 همگی معادل با

قالب رشته ها:

**STR ::=** DoubleQuote StringCharacter\* DoubleQuote  
 StringCharacter ::= ~(DoubleQuote , Enter) | DoubleQuote DoubleQuote

توجه: طول رشته نباید بیشتر از ۲۵۵ حرف باشد.

مثال:

رشته	معنای رشته در موقع چاپ شدن
"abc""d""e'f"	abc"d""e'f
""	

قالب علائم کاراکتری (تک حرفی):

**CHR ::=** DoubleQuote StringCharacter DoubleQuote **C**

مثال:

کاراکتر	معنای کاراکتر در موقع چاپ
""c	"
"a"C	a

**DTM ::= # SpaceTab\* DateOrTime SpaceTab\* #**  
 DateOrTime ::=  
 DateValue SpaceTab+ TimeValue | DateValue | TimeValue  
 DateValue ::= IntLiteral / IntLiteral / IntLiteral | IntLiteral – IntLiteral – IntLiteral  
 TimeValue ::= IntLiteral : IntLiteral (ε | : IntLiteral) SpaceTab\* (ε | AMPM )  
 SpaceTab ::= Space | Tab  
 AMPM ::= AM | PM

مثال:

Dim d as Date  
 d = # 8/23/1970 3:45:39AM #  
 d = # 3:45:39AM # ' Date value: 1/1/1 3:45:39AM.  
 d = #3:45:39 # ' Date value: 1/1/1 3:45:39AM.  
 d = # 13:45:39 # ' Date value: 1/1/1 1:45:39PM.  
 d = #13:45:39PM # ' خطای منطقی دارد اما شما آن را خطا در نظر نگیرید!!  
 d = #1AM # → ERROR: Line 7, Column 5

علامت ویژه:

( ) { } # , . :  
 & \* + - / \ ^ < = >  
 <> <= >= << >>

کلمات کلیدی:

AddHandler	AddressOf	Alias	And	AndAlso	Ansi	As	Assembly
Auto	Boolean	ByRef	Byte	ByVal	Call	Case	Catch CBool
CByte	CChar	CDate	CDbl	CDec	Char	CI nt	Class CLng
CObj	Const	CShort	CSng	CStr	CType	Date	Decimal
Declare	Default	Delegate	Dim	DirectCast	Do	Double	Each
Else	Elseif	End	EndIf	Enum	Erase	Error	Event Exit
ExternalSource	False	Finally	For	Friend	Function	Get	
GetType	GoSub	GoTo	Handles	If	Implements	Imports	In
Inherits	Integer	Interface	Is	Let	Lib	Like	Long Loop
Me	Mod	Module	MustInherit	MustOverride	MyBase	MyClass	
Namespace	New	Next	Not	Nothing	NotInheritable		
NotOverridable	Object	On	Option	Optional	Or	OrElse	
Overloads	Overridable	Overrides	ParamArray	Preserve	Private		
Property	Protected	Public	RaiseEvent	ReadOnly	ReDim	Region	
RemoveHandler	Resume	Return	Select	Set	Shadows	Shared	
Short	Single	Static	Step	Stop	String	Structure	Sub
SyncLock	Then	Throw	To	True	Try	TypeOf	Unicode Until
Variant	Wend	When	While	With	WithEvents	WriteOnly	Xor

## نکات لازم در پیاده سازی:

در این قسمت پروژه یک روال Initialize جهت کارهای اولیه و یک تابع Scanner جهت تشخیص و برگرداندن کلمات باید نوشته شود و برای تست آنها، یک تابع Write جهت چاپ یک کلمه باید نوشته شده و در داخل روال main در یک حلقه، کلمه تشخیص داده شده و چاپ می شوند.

```
int main(int argc, char* argv[]) {
    Initialize(argc, argv);
    while (1) {
        Token t = Scanner( );
        Write( t);
        if (t == _EOF) break;
    } }
```

برنامه باید به صورت command line یک فایل را از کاربر باید بگیرد و آن را اجرا کند. مثلاً:

```
Scanner.exe c:\temp\input.txt
```

در اجرا باید کلمه کلمات مفید، در یک خط نمایش داده شوند. در صورتی که کلمه ID و یا INT و یا FLT و یا STR و یا CHR و یا DTM باشد، معنای آن جلو آن نمایش داده شود. برای اعداد اعشاری باید مقدار عددی به صورت اعشاری با ۲ رقم اعشار نمایش داده شود. اعداد صحیح هگزادسیمال و یا اکتال نیز باید به فرم دهدهی تبدیل شوند. برای رشته ها و کاراکترها نیز علامت گیومه اول و آخر و گیومه های زاید داخل آن باید حذف شوند. برای سایر موارد نیازی به توضیح اضافه نیست و فقط خود کلمه باید چاپ شود. (برای ID و یا کلمات کلیدی کل کلمه باید با حروف بزرگ نشان داده شود.) برای تاریخ نیازی به حذف فاصله های زائد نیست اما حذف آنها نیز بلا اشکال است.

بعد از چاپ هر کلمه کاربر یک Enter می زند و کلمات بعدی به ترتیب نمایش داده می شوند.

در صورت برخورد به خطای لغوی، باید شماره خط و شماره ستون (حرف چندم آن خط) دارای مشکل نمایش داده شود

```
ERROR: Line 1, Column 7
```

و برنامه خاتمه پیدا کند. مثلاً:

در عمده موارد تعریف خطای لغوی آن است که اتوماتای طراحی شده دچار Fail شود و یا این که کلمه از لحاظ طول و قالب مشکل داشته باشد.

## مثال ۱:

```
If_&h1g[k2]"" "a"Char.4e-1do"" "const<>>ReM &o4
# const## 13:45 pM #13:45ij%'_abc_
4.g&gRem if
➔
ID IF_
INT 1
ID G
ID [K2]
STR "a
CHAR
FLT 0.04
ID O
CHR "
```

```
ID ONST
<>
>
#
ID CONST#
DTM # 13:45 pM #
INT 13
:
INT 45
ID J%
FLT 4.00
ID G
&
ID GREM
IF
EOF
```

مثال ٢:

```
13Rem>>
While 12_%^5 Rem----
➔
FLT 13.00
ID EM
>>
WHILE
INT 12
ERROR: Line 2, Column 10
```

مثال ٣:

```
&&hg4"abc"
➔
&
ERROR: Line 1, Column 4
```

مثال ٤:

```
"c" c & "ab" const
def "&x"
➔
CHR c
&
ERROR: Line 1, Column 16      ERROR: Line 2, Column 1
```