

Строковые величины

Повторимся, что **строка** - это ограниченная апострофами последовательность любых **символов**, например: 'Школа', '2009', 'ГАЗ-21', 'Мама мыла раму'. Для того чтобы апостроф входил в состав строки его необходимо продублировать, например: 'Ab"java', 'don"t'.

```
var  
s: String;  
s1: String[40]; {Для экономии памяти строка из 40 символов}
```

Переменная типа String, содержит любое количество символов. В данном случае в первой строке может содержаться максимум 40 символов.

Переменной строкового типа можно присвоить значение с помощью команды присваивания. Значение строковой величины записывается в апострофах.

Пустая строка задается следующим образом: `s := '';`

Запись поясняющего текста при выводе в команде `write` является строковой константой.

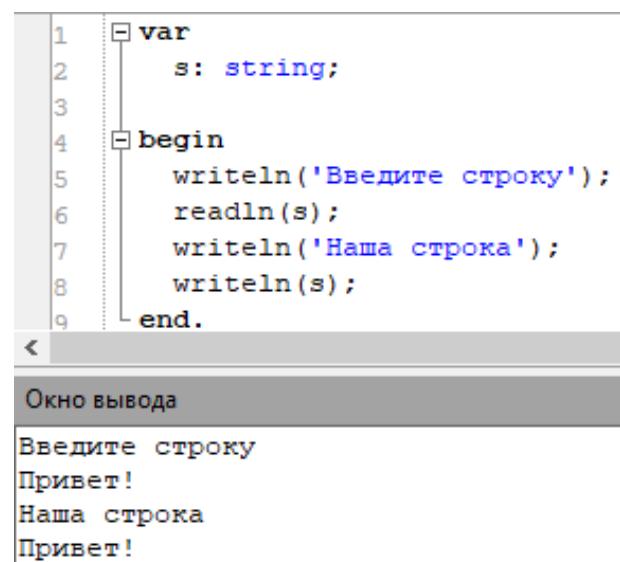
Переменные типа **String** выводятся на экран посредством стандартных процедур **Write** и **WriteLn** и вводятся с помощью стандартной процедуры **ReadLn**. То есть, вводятся не поэлементно, а целиком.

Примечание. Если при вводе задать символов больше, чем максимально допустимо, то лишние символы будут проигнорированы. Обратим внимание, что при вводе строки открывающиеся и закрывающиеся апострофы не набираются.

Пример 1.

Ведите строку `s` и выведите её значение на экран.

```
var  
s: string;  
  
begin  
writeln('Введите строку');  
readln(s);  
writeln('Наша строка');  
writeln(s);  
end.
```



The screenshot shows a Delphi IDE interface. On the left, there is a code editor window with the following code:

```
1 var  
2   s: string;  
3  
4 begin  
5   writeln('Введите строку');  
6   readln(s);  
7   writeln('Наша строка');  
8   writeln(s);  
9 end.
```

On the right, there is an 'Окно вывода' (Output Window) titled 'Введите строку'. It contains the following text:
Привет!
Наша строка
Привет!

К символам в строке можно обращаться, используя индекс.

Нумерация символов начинается с единицы, `s[i]` соответствует i-му символу в строке `s`.

Самый первый байт в строке имеет индекс равный нулю и содержит текущую длину строки.

S =	1	2	3	4	5	6	7	8	9	10	11
	И	Н	Ф	О	Р	М	А	Т	И	К	А

`s[5] = 'Р'`
`s[10] = 'К'`

Пример 2.

Вывести пятый и десятый символ введенной строки (количество символов в строке больше 10).

```
var
  s: string;
begin
  Readln(s);
  writeln(s[5]);
  writeln(s[10]);
end.
```

```
1 var
2   s: string;
3 begin
4   Readln(s);
5   writeln(s[5]);
6   writeln(s[10]);
7 end.
```

Окно вывода

Информатика

р
к

Пример 3.

Дана строка s. Вывести на экран символы строки, стоящие под номерами n, m, k. n, m, k вводятся с клавиатуры. (Значения n, m, k не превосходят длину строки).

```
var
  s:string;
  n,m,k:integer;
begin
  Readln(s);
  Readln(n,m,k);
  writeln(s[n]);
  writeln(s[m]);
  writeln(s[k]);
end.
```

```
1 var
2   s:string;
3   n,m,k:integer;
4 begin
5   Readln(s);
6   Readln(n,m,k);
7   writeln(s[n]);
8   writeln(s[m]);
9   writeln(s[k]);
10 end.
```

Окно вывода

Информатика
6 9 5
м
и
р

Операции со строками.

В Паскале имеется два основных способа обработки переменных типа **String**. Первый способ предполагает обработку всей строки как единого целого, т.е. единого объекта. Кроме того (это второй способ), можно рассматривать строку как составной объект, состоящий из отдельных символов, то есть элементов типа **Char** (символ), которые при обработке доступны каждый в отдельности.

1. **Склейивание (конкатенация).** Под склейиванием (сцеплением) понимается последовательное объединение нескольких строк. Склейивание осуществляется с помощью оператора «+». Склейивать (сцеплять) можно как строковые константы, так и переменные. Например:

```
var
  S1, S2, S3: String[20];

begin
  S1 := 'Я';
  S2 := 'учусь';
  S3 := 'в 9 классе';
  S3 := S1 + ' ' + S2 + ' ' +
S3;
  Write(S3)
end.
```

The screenshot shows a Pascal IDE interface. On the left is the code editor with the following text:

```
1  var
2    S1, S2, S3: String[20];
3
4  begin
5    S1 := 'Я';
6    S2 := 'учусь';
7    S3 := 'в 9 классе';
8    S3 := S1 + ' ' + S2 + ' ' +
9      S3;
10   end.
```

On the right is the 'Окно вывода' (Output window) containing the text:

Я учусь в 9 классе

Замечания.

«Склейте» строки можно также при помощи функции **Concat** (**S1, S2, ..., SN**).
От переменны мест слагаемых сумма в операции склейивания **МЕНЯЕТСЯ**.

2. **Сравнение.** Строки сравниваются посимвольно. Сначала сравниваются первые символы двух строк. Если символы различны, то больше та строка, символ которой имеет больший номер. символы одинаковые, то переходят к сравнению следующих символов. Сравнение заканчивается, когда найдены различные символы или в одной из строк закончились символы — в этом случае больше та строка, в которой символы остались. Если при сравнении символов различия не найдены и строки закончились одновременно, то они равны.

Порядок, в котором меньшая строка предшествует большей, называют **лексикографическим**. Это название он получил по аналогии с размещением по алфавиту в словаре. При сравнении строк следует помнить, что заглавные и строчные буквы — это разные буквы, поскольку имеют различные номера в таблице символов.

При работе со строками можно использовать любые операции отношения (**>**, **<**, **=**, **<>**, **>=**, **<=**) и их комбинации в условных операторах. Их результат — это одно из двух значений: **True** или **False**.

Кодовая таблица символов ASCII (32—127)

Код	Символ	Код	Символ	Код	Символ	Код	Символ
32		56	8	80	P	104	h
33	!	57	9	81	Q	105	i
34	"	58	:	82	R	106	j
35	#	59	;	83	S	107	k
36	\$	60	<	84	T	108	l
37	%	61	=	85	U	109	m
38	&	62	>	86	V	110	n
39	'	63	?	87	W	111	o
40	(64	@	88	X	112	p
41)	65	A	89	Y	113	q
42	*	66	B	90	Z	114	г
43	+	67	C	91	І	115	s
44	,	68	D	92	\	116	t
45	-	69	E	93	І	117	u
46	.	70	F	94	^	118	v
47	/	71	G	95		119	w
48	0	72	H	96	'	120	x
49	1	73	I	97	a	121	y
50	2	74	J	98	b	122	z
51	3	75	K	99	с	123	{
52	4	76	L	100	d	124	
53	5	77	M	101	e	125	}
54	6	78	N	102	f	126	~
55	7	79	O	103	g	127	□

Кодовая таблица символов CP-1251 (128—255)

128	Ђ	160		192	А	224	а
129	Ѓ	161	Ӵ	193	Б	225	б
130	,	162	ӵ	194	В	226	в
131	Ӷ	163	҂	195	Г	227	г
132	ӷ	164	҃	196	҄	228	д
133	...	165	҅	197	҆	229	е
134	†	166	҇	198	҈	230	ж
135	‡	167	҉	199	Ҋ	231	з
136	€	168	ҋ	200	Ҍ	232	и
137	%o	169	Ҏ	201	ҏ	233	й
138	Љ	170	Ҏ	202	ҏ	234	к
139	‹	171	ҏ	203	ҏ	235	л
140	Њ	172	—	204	ҏ	236	м
141	Ќ	173		205	Ҥ	237	н
142	Ћ	174	®	206	Ѻ	238	о
143	Џ	175	҈	207	Ҍ	239	п
144	Ѡ	176	°	208	Ҍ	240	р
145	‘	177	±	209	Ҍ	241	с
146	’	178	Ҍ	210	Ҍ	242	т
147	“	179	и	211	Ҍ	243	у
148	”	180	Ҍ	212	Ҍ	244	ф
149	•	181	µ	213	Ҍ	245	х
150	—	182	Ҍ	214	Ҍ	246	Ҍ

Примеры сравнения строк:

S1	S2	Результат
'string'	'char'	S1 > S2
'строка'	'Строка'	S1 > S2
'книга'	'Журнал'	S1 > S2
'мебель'	'стол'	S1 < S2
'липа'	'лист'	S1 < S2
'леска'	'лес'	S1 > S2
'112'	'7'	S1 < S2
'123'	'11111111'	S1 > S2
'123'	'123'	S1 = S2

Пример 4.

Даны строки s1 и s2. Вывести на экран большую (лексикографически) из них.

```
var
  S1, S2: string;

begin
  readln(S1);
  readln(S2);
  if S1 > S2
    then
      writeln(S1)
    else
      writeln(S2);
end.
```

```
1  var
2    S1, S2: string;
3
4  begin
5    readln(S1);
6    readln(S2);
7    if S1 > S2
8      then
9        writeln(S1)
10   else
11     writeln(S2);
12 end.
```

Окно вывода

```
Привет
Пока
Привет
```