

Advanced SAS® Programming Techniques

This tip sheet is associated with the SAS® Certified Professional Prep Guide Advanced Programming Using SAS® 9.4. For more information, visit www.sas.com/certify

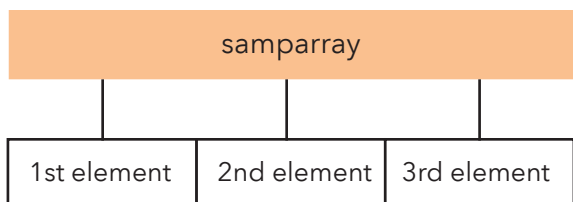
Arrays

Defining an array

```
ARRAY array-name<[number-of-array-elements]>  
<$> <length> <array-elements>  
<_TEMPORARY_> <(initial-values)>;
```

Referencing an array

```
array-name[element-number];
```



PDV	samparray[1]	samparray[2]	samparray[3]
EmpID	Salary	Bonus	Raise_Percent

The number of elements must be enclosed in either parentheses (), braces { }, or brackets [].

Unknown Number of Elements

Use an asterisk (*) within your brackets when defining an array.

Use the DIM function to return the number of elements in an array.

```
DIM(array-name);
```

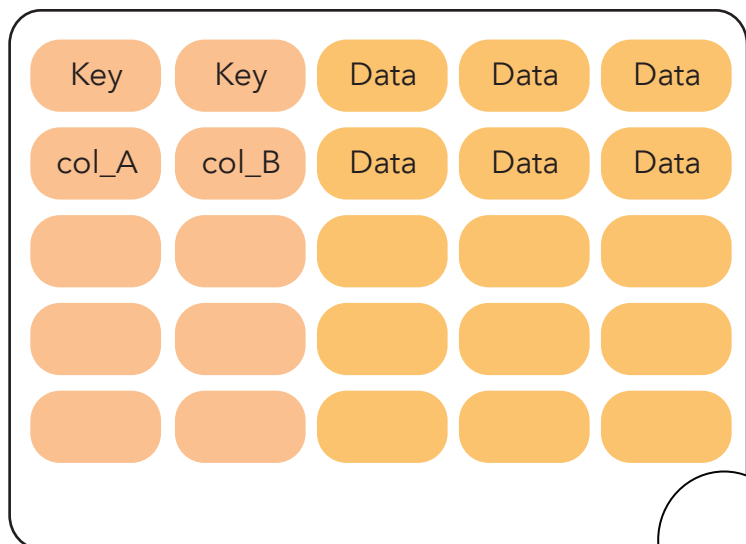
Two-Dimensional Arrays

```
ARRAY array-name  
[number-of-rows,number-of-columns];
```

```
array samplearray[3,2];
```

The example above creates an array named SampleArray which has 3 rows and 2 columns.

Hash Objects



A hash object is an in-memory table that contains key and data components.

Hash Object and Iterator Process

Declaring hash object or hash iterator object:

```
DECLARE hash object-name  
(<argument_tag-1:value-1, ...>);
```

```
DECLARE hiter object-name  
('hash-object-name');
```

Defining a hash object:

```
object-name.ADD( );  
object-name.DEFINEKEY('key-1' <, ...'key-n'>);  
object-name.DEFINEDATA('data-1' <, ...'data-n'>);  
object-name.DEFINEDONE( );  
object-name.OUTPUT( );
```

Using a hash object:

```
object-name.FIND( );
```

Retrieving a hash object with a hash iterator object:

```
object-name.FIRST( );  
object-name.LAST();  
object-name.NEXT();  
object-name.PREV( );
```

Advanced SAS® Programming Techniques

This tip sheet is associated with the SAS® Certified Professional Prep Guide Advanced Programming Using SAS® 9.4. For more information, visit www.sas.com/certify

Picture Formats

```
PROC FORMAT;  
  PICTURE format-name <(format-options)>  
  <value-range-set-1= 'template-value' (template-options)>  
  <value-range-set-n= 'template-value' (template-options)>;  
RUN;
```

Options

Creating Custom Date, Time, Datetime Formats

DATATYPE=DATE | TIME | DATETIME
enables the use of directives in the picture as a template to format date, time, or datetime values.

DEFAULT=length
specifies the default length of the picture.

Creating Custom Numeric Formats

MULT|MULTIPLIER=*n*
specifies a number to multiply the value by.

PREFIX='prefix'
specifies a character prefix to place in front of the formatted value.

ROUND
rounds the value to the nearest integer before formatting.

Creating Functions

```
PROC FCMP OUTLIB=libref.table.package;  
  FUNCTION function-name(arguments) <$> <length>;  
  ... programming statements ...  
  RETURN(expression);  
ENDSUB;  
QUIT;
```

Using Custom Functions

OPTIONS CMPLIB=*libref.table* | (*libref.table-1...libref.table-n*)

Advanced Functions

```
LAG<n>(column);  
COUNT(string, substring<,modifiers>);  
COUNTC(string, character-list<,modifiers>);  
COUNTW(string, <,delimiters><,modifiers>);  
FIND(string, substring<,modifiers><,start-position>);  
FINDC(string, character-list<,modifiers> <,start-position>);  
FINDW(string, word<,delimiters><,modifiers><,start-position>);
```

Perl Regular Expressions

Perl Regular Expressions Metacharacters

Metacharacter	Description
/.../	Starting and ending delimiter
(...)	Enables grouping
	Denotes the OR situation
\d	Matches a digit (0-9)
\D	Matches a non-digit such as letter
\s	Matches a whitespace character
\w	Matches a group of characters
.	Matches any character
[...]	Matches a character in brackets
[^...]	Matches a character not in brackets
^	Matches the beginning of the string
\$	Matches the end of the string
\b	Matches a word boundary
\B	Matches a non-word boundary
*	Matches the preceding character 0 or more times
+	Matches the preceding character 1 or more times
?	Matches the preceding character 0 or 1 times
{ <i>n</i> }	Matches exactly <i>n</i> times
\	Overrides the next metacharacter such as a (or ?)

PRXMATCH Function

PRXMATCH function searches for a pattern match and returns the position at which the pattern is found.

PRXMATCH (*Perl-regular-expression*, *source*);

PRXPARSE Function

PRXPARSE function returns a pattern identifier number that is used by other PRX functions and call routines.

pattern-ID-number=PRXPARSE (*Perl-regular-expression*);

PRXCHANGE Function

PRXCHANGE function performs a substitution for a pattern match

PRXCHANGE (*Perl-regular-expression*, *times*, *source*)