Arrays
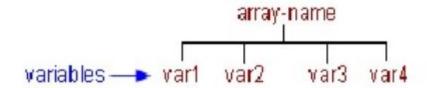
▸ A SAS **array** is a temporary grouping of SAS variables under a single name. An array exists only for the duration of the DATA step.

Let's Consider This Problem

```
net_sav1 = inc1- exp1;
net_sav2 = inc2- exp2; . . .
.
.
.
.
net_sav3 = inc3- exp3;
net_sav11 = inc11-exp11;
net_sav12 = inc12- exp12;
```
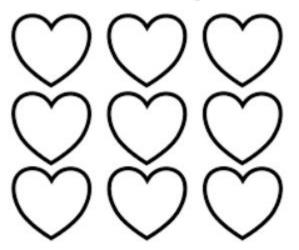
What is An Array

Questions????

Which is the repeated addition sentence for this array?
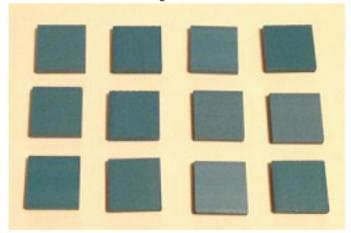

○ 9 + 9 + 9 = 27

○ 3 + 3 + 3 = 9

○ 2 + 2 + 2 = 6

○ 1 + 1 + 1 = 3

How many rows and columns does the array have?

- ○ 4 columns and 3 rows

- ○ 3 columns and 3 rows

- ○ 3 columns and 4 rows

- ○ 4 columns and 4 rows
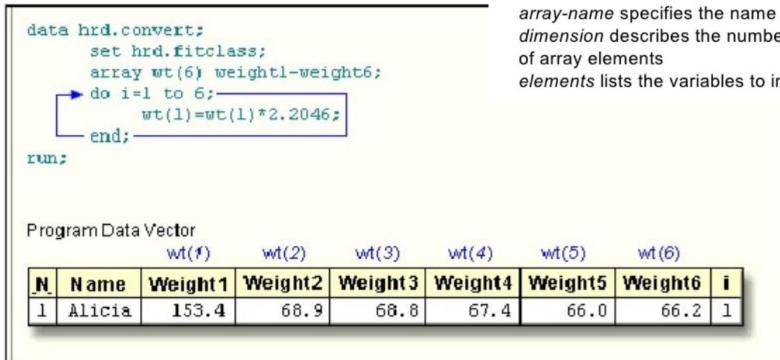
Name	Weight1	Weight2	Weight3	Weight4	Weight5	Weight6
Alicia	69.6	68.9	68.8	52.6	52.6	51.7
Brenda	68.6	67.6	67.0	67.6	66.6	66.0

ARRAY *array-name*{*dimension*} *elements*;
where
array-name specifies the name of the array
dimension describes the number and arrangement of array elements
elements lists the variables to include in the array.

```
data hrd.convert;
    set hrd.fitclass;
    array wt(6) weight1-weight6;
    do i=1 to 6;
        wt(1)=wt(1)*2.2046;
    end;
run;
```

Program Data Vector

		wt(1)	wt(2)	wt(3)	wt(4)	wt(5)	wt(6)	
N	Name	Weight1	Weight2	Weight3	Weight4	Weight5	Weight6	i
1	Alicia	153.4	68.9	68.8	67.4	66.0	66.2	1

```
# Declare an array of length 4 named age with values.
ARRAY age[4] (11 1 2 62);

# Declare an array of length 8 named colors with values starting at index 0.
ARRAY colors (0:5) A B C D E F G H I;

# Declare an array of length 5 named books which contain character values.
ARRAY books(1:5) $ b1-b5;

# Declare an array of required length depending on the number of values supplied.
ARRAY ANS(*) A1-A10;
```
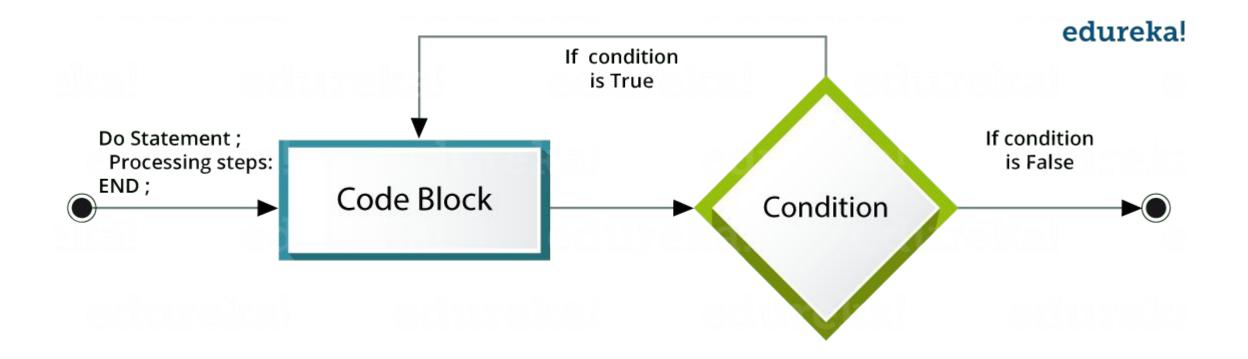
```
DATA avgfahrenheit;
    set avgcelsius;
    janf = 1.8*jan + 32;
    febf = 1.8*feb + 32;
    marf = 1.8*mar + 32;
    aprf = 1.8*apr + 32;
    mayf = 1.8*may + 32;
    junf = 1.8*jun + 32;
    julf = 1.8*jul + 32;
    augf = 1.8*aug + 32;
    sepf = 1.8*sep + 32;
    octf = 1.8*oct + 32;
    novf = 1.8*nov + 32;
    decf = 1.8*dec + 32;
    drop jan feb mar apr may jun
        jul aug sep oct nov dec;
RUN;


PROC PRINT data = avgfahrenheit;
    title 'Average Monthly Temperatures in Fahrenheit';
    id City;
    var janf febf marf aprf mayf junf
        julf augf sepf octf novf decf;
RUN;
```

```
array month{4} jan feb mar apr;
    do i=1 to 4;
    New_variable=month{i}*2;
    end;
```
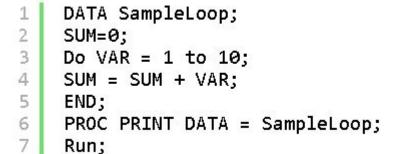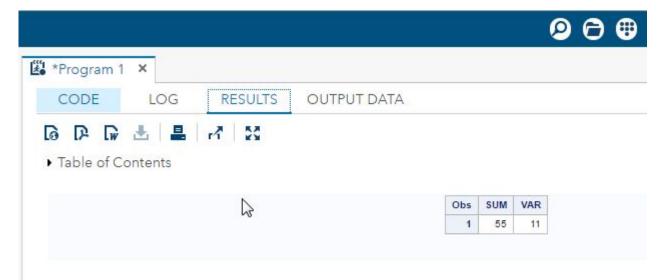

edureka!

If condition
is True

Do Statement ;
 Processing steps:
END ;

Code Block

Condition

If condition
is False

Do Index loop

Syntax:

```
1   Do indexvariable = initialvalue to finalvalue;
2   SAS statements;
3   End;
```

```
1   DATA SampleLoop;
2   SUM=0;
3   Do VAR = 1 to 10;
4   SUM = SUM + VAR;
5   END;
6   PROC PRINT DATA = SampleLoop;
7   Run;
```

Program 1 ✕

| CODE | LOG | RESULTS | OUTPUT DATA |

▸ Table of Contents

Obs	SUM	VAR
1	55	11

Do While Loop

Syntax:

```
1   Do While (condition);
2   SAS statements;
3   End;
```

```
1   DATA SampleLoop;
2   SUM=0;
3   VAR=1;
4   Do While(VAR<15);
5   SUM = SUM + VAR;
6   VAR+1;
7   END;
8   PROC PRINT DATA = SampleLoop;
9   Run;
```

Program 1 ×

CODE   LOG   RESULTS   OUTPUT DATA

▸ Table of Contents

Obs	SUM	VAR
1	105	15

Do Until Loop

```
1  Do Until (condition);
2  SAS statements;
3  END;
```

```
1  DATA SampleLoop;
2  SUM=0;
3  VAR=1;
4  Do Until(VAR>15);
5  SUM=SUM+VAR;
6  VAR+1;
7  END;
8  PROC PRINT;
9  Run;
```


Obs	SUM	VAR
1	120	16

Questions????

1. An iterative DO loop must have a stop value.

- a. True
- b. False

---

2. Which statement is **false** regarding nested DO loops?

- a. Each DO statement must have a corresponding END statement.
- b. Each DO loop must have its own index variable.
- c. Each DO loop must use the same increment value.
- d. Each DO loop can contain iterated □SAS statements.

During each execution of the following DO loop, the value of **Earned** is calculated and is added to its previous value. How many times does this DO loop execute?

```
data finance.earnings;
    Amount=1000;
    Rate=.075/12;
    do Month=1 to 12;
        Earned+(Amount+Earned)*Rate;
    end;
run;
```

- *a.* 0
- *b.* 1
- *c.* 12
- *d.* 13

```sas
DATA avgfahrenheit;
    set avgcelsius;
    janf = 1.8*jan + 32;
    febf = 1.8*feb + 32;
    marf = 1.8*mar + 32;
    aprf = 1.8*apr + 32;
    mayf = 1.8*may + 32;
    junf = 1.8*jun + 32;
    julf = 1.8*jul + 32;
    augf = 1.8*aug + 32;
    sepf = 1.8*sep + 32;
    octf = 1.8*oct + 32;
    novf = 1.8*nov + 32;
    decf = 1.8*dec + 32;
    drop jan feb mar apr may jun
        jul aug sep oct nov dec;
RUN;


PROC PRINT data = avgfahrenheit;
    title 'Average Monthly Temperatures in Fahrenheit';
    id City;
    var janf febf marf aprf mayf junf
        julf augf sepf octf novf decf;
RUN;
```

```sas
DATA avgfahrenheit;
    set avgcelsius;
    array fahr(12) jan feb mar apr may jun
                   jul aug sep oct nov dec;
    do i = 1 to 12;
        fahr(i) = 1.8*fahr(i) + 32;
    end;
RUN;

PROC PRINT data = avgfahrenheit;
    title 'Average Monthly Temperatures in Fahrenheit';
    id City;
    var jan feb mar apr may jun
        jul aug sep oct nov dec;
RUN;
```

Average Monthly Temperatures in Fahrenheit

City	jan	feb	mar	apr	may	jun
State College, PA	28.4	28.4	35.6	46.4	57.2	66.2
Miami, FL	68.0	68.0	71.6	73.4	78.8	80.6
St. Louis, MO	30.2	33.8	42.8	55.4	64.4	73.4
New Orleans, LA	51.8	55.4	60.8	68.0	73.4	80.6
Madison, WI	17.6	23.0	32.0	44.6	57.2	66.2
Houston, TX	50.0	53.6	60.8	68.0	73.4	80.6
Phoenix, AZ	53.6	57.2	60.8	69.8	78.8	87.8
Seattle, WA	41.0	42.8	44.6	50.0	55.4	60.8
San Francisco, CA	50.0	53.6	53.6	55.4	57.2	59.0
San Diego, CA	55.4	57.2	59.0	60.8	62.6	66.2

| City	jul	aug	sep	oct	nov	dec
State College, PA	69.8	68.0	60.8	50.0	39.2	30.2
Miami, FL	82.4	82.4	80.6	78.8	73.4	68.0
St. Louis, MO	78.8	77.0	69.8	59.0	44.6	33.8
New Orleans, LA	80.6	80.6	78.8	69.8	60.8	53.6
Madison, WI	71.6	68.0	60.8	50.0	35.6	23.0
Houston, TX	82.4	82.4	78.8	69.8	60.8	53.6
Phoenix, AZ	91.4	89.6	86.0	73.4	60.8	53.6
Seattle, WA	64.4	64.4	60.8	53.6	46.4	42.8
San Francisco, CA	59.0	60.8	62.6	60.8	57.2	51.8
San Diego, CA	69.8	71.6	69.8	66.2	60.8	57.2
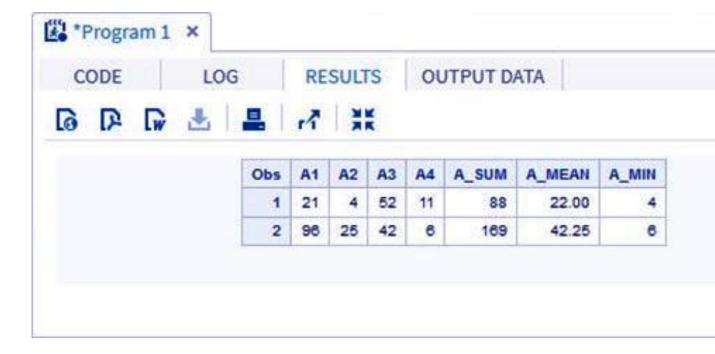
```sas
DATA avgtemps;
    set avgcelsius;
    array celsius(12) jan feb mar apr may jun
                      jul aug sep oct nov dec;
    array fahr(12) janf febf marf aprf mayf junf
                   julf augf sepf octf novf decf;
    do i = 1 to 12;
        fahr(i) = 1.8*celsius(i) + 32;
    end;
RUN;


PROC PRINT data = avgtemps;
    title 'Average Monthly Temperatures';
    id City;
    var jan janf feb febf mar marf;
    var apr aprf may mayf jun junf;
    var jul julf aug augf sep sepf;
    var oct octf nov novf dec decf;
RUN;
```

```
DATA example_OF;
INPUT A1 A2 A3 A4;
ARRAY A(4) A1-A4;
A_SUM=SUM(OF A(*));
A_MEAN=MEAN(OF A(*));
A_MIN=MIN(OF A(*));
DATALINES;
21 4 52 11
96 25 42 6;
RUN;
PROC PRINT DATA=example_OF;
RUN;
```

CODE | LOG | RESULTS | OUTPUT DATA

Obs	A1	A2	A3	A4	A_SUM	A_MEAN	A_MIN
1	21	4	52	11	88	22.00	4
2	96	25	42	6	169	42.25	6

```
DATA in_example;
INPUT A1 $ A2 $ A3 $ A4 $;
ARRAY COLOURS(4) A1-A4;
IF 'yellow' IN COLOURS THEN available='Yes'; ELSE
available='No';
DATALINES;
Orange-pink violet yellow;
RUN;
PROC PRINT DATA=in_example;
RUN;
```


Obs	A1	A2	A3	A4	available
1	Orange	pink	violet	yellow	Yes

Referring to Values in an Array

NEWVAL=(TIME[5]/365);

Questions????

Which statement is *false* regarding an ARRAY statement?

- ○ It is an executable statement

- ○ It can be used to create variables

- ○ It must contain either all numeric or all character elements

- ○ It must be used to define an array before the array name can be referenced.

For the program below, select an iterative DO statement to process all elements in the contrib array. data work.contrib array contrib{4} qtr1-qtr4;   ...   contrib{i}=contrib{i}*1.25; end; run;

○ Do i=4;

○ Do i=1 to 4;

○ Do until i=4;

○ Do while i le 4;