

```
/* LOOPS IN SAS: */

/* LOOPS: */
/* IT IS A PART OF A PROGRAM WHICH HELPS US TO
REPEAT CALCULATIONS MULTIPLE NUMBERS OF TIMES FOR A GIVE SET OF PARAMETERS. */

/* CREATE A TABLE */

DATA COUNTER_DATA;
DO COUNTER = 1 TO 10;
    OUTPUT;
END;
RUN;

/* JOIN THE CREATED TABLE WITH THE MAIN TABLE */

DATA COM;
SET WORK.lop1;
SET WORK.counter_data;
RUN;

/* EG-2: */

DATA TEST;
DO TABLE = 16 TO 160 BY 16;
    OUTPUT;
END;
RUN;

PROC PRINT DATA=TEST;

/* TYPES OF LOOPS: */
/* 1. ITERATIVE DO LOOPS */
/* 2. DO WHILE LOOPS */
/* 3. DO UNTIL LOOPS */

/* 1. ITERATIVE DO LOOPS: */
/* THE ITERATIVE DO LOOPS EXECUTES THE STATEMENTS BETWEEN DO AND
END REPETITIVELY BASED ON THE VALUE OF AN INDEX VARIABLE. */

/* SYNTAX--> */

/* DO INDEX-VARIABLE = 'START' TO 'STOP' <BY INCREMENT>; */
/* SAS STATEMENTS */
/* END; */

/* EG-1: */

DATA TEST1;
DO TABLE = 2 TO 20 BY 2;
    OUTPUT;
END;
RUN;

PROC PRINT DATA=TEST1;

/* EG-2: */
/* USING DO LOOP FOR MULTIPLE STATEMENTS */

DATA CLASS1;
SET SASHELP.CLASS;
FORMAT STAY CATEGORY $30.;

IF AGE <= 12 THEN DO;
    FEE="15K";
    STAY="ALLOWED";
    CATEGORY="KID";
END;
ELSE DO
    FEE="25K";
    STAY="NOT ALLOWED";
    CATEGORY="TEENAGE";
END;
RUN;

PROC PRINT DATA=CLASS1;
```

```

/* 2. DO WHILE LOOPS: */
/* IT EXECUTES STATEMENTS REPETITIVELY TILL THE TIME GIVEN CONDITION IS TRUE. */
/* SYNTAX--> */

/* DO WHILE (EXPRESSION); */
/* ...MORE SAS STATEMENTS... */
/* END; */

/* NOTES:--> */
/* 1. THE EXPRESSION IS EVALUATED AT THE TOP OF THE LOOP BEFORE THE STATEMENTS IN THE DO LOOP ARE EXECUTED. */
/* 2. IF THE EXPRESSION IS FALSE THE FIRST TIME IT IS EVALUATED, THE DO LOOP DOES NOT ITERATE EVEN ONCE. */

/* EG-1: */

```

```

DATA TEST2;
X=1;
DO WHILE (X<=3);
    OUTPUT;
    X+1;
END;
RUN;

```

```

/* 3. DO UNTIL LOOPS: */
/* IT EXECUTES STATEMENTS REPETITIVELY TILL THE TIME GIVEN CONDITION BECOMES TRUE. */

/* SYNTAX:--> */

/* DO UNTIL (EXPRESSION); */
/* ...MORE SAS STATEMENTS.. */
/* END; */

/* NOTES:--> */
/* 1. THE EXPRESSION IS EVALUATED AT THE BOTTOM OF THE LOOP
AFTER THE STATEMENTS IN THE DO LOOP HAVE BEEN EXECUTED. */
/* 2. IF THE EXPRESSION IS TRUE, THE DO LOOP DOES NOT ITERATE AGAIN. */
/* 3. THE DO LOOP ALWAYS ITERATES AT LEAST ONCE. */

/* EG-1: */

```

```

DATA TEST3;
X=1;
DO UNTIL (X>3);
    OUTPUT;
    X+1;
END;
RUN;

```

```

/* SUMMARY & DIFFERENCE BETWEEN 'DO WHILE' & 'DO UNTIL': */

```

```

/* DO WHILE

```

```

/* 1. IT ITERATES TILL THE TIME CONDITION IS TRUE
/*

```

```

/* 2.

```

```

DATA TEST2;
X=1;
DO UNTIL (X>3);
    OUTPUT;
    X+1;
END;
RUN;

```

```

/* 3. THE EXPRESSION IS EVALUATED AT THE TOP OF THE LOOP
/* BEFORE THE STATEMENTS IN THE DO LOOP ARE EXECUTED.

/* 4. IF THE EXPRESSION IS FALSE THE FIRST TIME IT IS EVALUATED,
/* THE DO LOOP DOES NOT ITERATE EVEN ONCE. */

/*

```

```

DO UNTIL */

```

```

1. ITERATES TILL THE TIME CONDITION BECOMES TRUE *
(i.e. ITERATES WHEN GIVEN CONDITION FALSE) */

```

```

2. */

```

```

DATA TEST3;
X=1;
DO WHILE (X<=3);
    OUTPUT;
    X+1;
END;
RUN;

```

```

3. THE EXPRESSION IS EVALUATED AT THE BOTTOM OF THE
AFTER THE STATEMENTS IN THE DO LOOP HAVE BEEN

4. IF THE EXPRESSION IS TRUE, THE DO LOOP DOES NOT

5. THE DO LOOP ALWAYS ITERATES AT LEAST ONCE. */

```

