```
/**********************************************************************
 * Program: example.sas
 * Purpose: SAS Example for HighlightJS Plug-in
 *********************************************************************/

%put Started at %sysfunc(putn(%sysfunc(datetime()), datetime.));
options
    errors = 20  /* Maximum number of prints of repeat errors */
    fullstimer   /* Detailed timer after each step execution  */
;

%let maindir = /path/to/maindir;
%let outdir  = &maindir/out.;
systask command "mkdir -p &outdir." wait;
libname main "&maindir." access = readonly;

data testing;
    input name $ number delimiter = ",";
    datalines;
    John,1
    Mary,2
    Jane,3
    ;
    if number > 1 then final = 0;
    else do;
        final = 1;
    end;
run;

proc sql &sqlopts;
create table waffles as
    select * from testing;
quit;

%put NOTE: Hello;
%put NOTE- Hello;
%put WARNING: Hello;
%put ERROR: Hello;
%put Something ERROR- Hello;

%macro testMacro(positional, named = value);
    %put positional = &positional.;
    %put named      = log(&named.);
%mend testMacro;
%testMacro(positional, named = value);

dm 'clear log output odsresults';

proc datasets lib = work kill noprint; quit;
libname _all_ clear;
```

```
/**********************************************************************
 * Program: example.sas
 * Purpose: SAS Example for HighlightJS Plug-in
 **********************************************************************/

%put Started at %sysfunc(putn(%sysfunc(datetime()), datetime.));
options
    errors = 20  /* Maximum number of prints of repeat errors */
    fullstimer   /* Detailed timer after each step execution  */
;

%let maindir = /path/to/maindir;
%let outdir  = &maindir/out.;
systask command "mkdir -p &outdir." wait;
libname main "&maindir." access = readonly;

data testing;
    input name $ number delimiter = ",";
    datalines;
    John,1
    Mary,2
    Jane,3
    ;
    if number > 1 then final = 0;
    else do;
        final = 1;
    end;
run;

proc sql &sqlopts;
create table waffles as
    select * from testing;
quit;

%put NOTE: Hello;
%put NOTE- Hello;
%put WARNING: Hello;
%put ERROR: Hello;

%macro testMacro(positional, named = value);
    %put positional = &positional.;
    %put named      = log(&named.);
%mend testMacro;
%testMacro(positional, named = value);

dm 'clear log output odsresults';

proc datasets lib = work kill noprint; quit;
libname _all_ clear;
```

```
program define excellentProgram
version 14.0

local hi  = `1'
local bye = `2'
local yes = ln(`hi')

* This is a comment
set obs `= _N + 1'
gen neg = 1 - 1 / (1 + exp(score))

/*
 * Multi line comments are pretty
 * because they span many lines
 */

reg y x
xi: reg y2 x i.dummy // This is another comment type

di "This is a normal string with a `local' $global ${global}"
di `"This is a "super string" that takes on anything"'
di "string`1'two${three}" bad `"string " "' good `"string " "'

// This also works at line starts
adopath ++ "${lib}/code/ado/"
cap adopath - SITE
cap adopath - PLUS
/*cap adopath - PERSONAL
cap adopath - OLDPLACE*/

forval i = 1 / 4{
  cap reg y x`i', robust
  if `i' == 2 {
    local c = _b[_cons]
    local b = _b[x`i']
    local x = ln(`i')
  }
}

* Something about how mata is really a second language within Stata
mata: mata mlib index
end
```

```
program define excellentProgram
version 14.0

local hi  = `1'
local bye = `2'
local yes = ln(`hi')

* This is a comment
set obs `= _N + 1'
gen neg = 1 - 1 / (1 + exp(score))

/*
 * Multi line comments are pretty
 * because they span many lines
 */

reg y x
xi: reg y2 x i.dummy // This is another comment type

di "This is a normal string with a `local' $global ${global}"
di `"This is a "super string" that takes on anything"'
di "string`1'two${three}" bad `"string " "' good `"string " "'

// This also works at line starts
adopath ++ "${lib}/code/ado/"
cap adopath - SITE
cap adopath - PLUS
/*cap adopath - PERSONAL
cap adopath - OLDPLACE*/

forval i = 1 / 4{
  cap reg y x`i', robust
  if `i' == 2 {
    local c = _b[_cons]
    local b = _b[x`i']
    local x = ln(`i')
  }
}

* Something about how mata is really a second language within Stata
mata: mata mlib index
end
```