IDL Tutorials

Day 5

Henry (Trae) Winter
winter@physics.montana.edu
http://solar.physics.montana.edu/winter/idl_tutorial/2003/index.html

Today's Topics

- Relational operators
- Conditional statements and
- Loops
- Boolean operators
- The End

Supplements

- When good programs go bad, and how to handle crashes
- More example programs

Relational Operators

- Relation operators test the relationships between two arguments (i.e. Is x greater than y?)
- Relation operators return 1 if they are true, 0 if false
- Can be used on strings with caveats
- Common R.O.'s
  - eq ;Equal to
  - ne ;Not equal to
  - lt ;less than
  - gt ;greater than
  - ge ;greater than or equal to
  - le ;less than or equal to

  Syntax:

  Arg1 R.O. Arg2

  Examples:

  >print, 1 gt –1

  >print, array_1 le array_2

If Statements

- Programs are only useful if they can make some basic decisions. This is often done with if statements.
- If statements use R.O.'s as a condition. If the R.O. returns true (1) then the if statement executes.
- An optional else statement can be put in case R.O. returns false. If there is not an else, IDL just ignores the line on R.O. false
- Best way to show if statements is to show their syntax

  Syntax:

  if condition then something

  if condition then something else something_else
- Examples:

  ```
  >if x ge y then print, x    ;
  >if x gt y then biggest=x  else biggest =y
  >if keyword_set(KEY) then KEY=KEY else key=0.5
  >if n_params() lt 1 then print, 'You need a parameter!'
  ```
- See also Case statements

Loops

- Loops allow for actions to occur repeatedly
- Usual syntax is:

  conditional_statement begin

  do_something

  do_something_else

  endstatement ; or just end
- Many types of loops
  - If then loops
  - For loops
  - While loops
  - Repeat loops

If Then Loop Syntax & Examples

Syntax:

```
>if true then begin
>                    ……
>                    ……
>endif
```

Example:

```
>if keyword_set(KEY) then begin
>        print,"Key set by user to be: "+ strcompress(string( KEY), $
>                                /REMOVE_ALL)
>        beta=cos(KEY)
>endif else begin
>        key=!pi/4
>        print, "Choose a key of: " +strcompress(string(key), $
>                                /REMOVE_ALL)
>        beta=cos(key)
>endelse
```

For Loop Syntax & Examples

- For statements repeat a statement until a counter reaches an assigned value.
- The statement automatically advances the counter by one after each step

Syntax:

>for *counter=value*, *number* do *statement*

or

>for *counter=value*, *number* do begin

> …

> …

>endfor

Examples:

```
>for counter=0, 10 do print, cos(((2*!pi)/10)*counter)
>for index=0, n_elements(array)-1 do begin
>    array[*,index]=cos(array_3[index, *])
>endfor ;index loop
```

While Loop Syntax & Examples

- While loops repeat until some conditional statement is met

Syntax:

>while *condition* do *statement*

or

>while *condition* do begin
> *statement*
> ….
>endwhile

Examples:

>while x ge y do x=x-5

>while x lt y do begin
> x=x+5
> print,x
>endwhile; x lt y

Repeat Loop Syntax & Examples

- Repeat loops repeat until some conditional statement is met
- The difference between repeat and while loops is the location of the conditional statement

Syntax:

>repeat *statement* until *condition*

or

> repeat begin
> *statement*
> …
>endrep until *condition*

Examples:

>repeat x=x-5 until x lt y

>repeat begin
> x=x+5
> print,x
>endrep until x gt y

Boolean Operators

- Boolean operators can be used to make complex groupings of conditional statements

  Ex: if (x lt y) and (x lt z) then print,"X is small
- There are four basic boolean operators
  - and; Returns true when both of its operands are true

    >print, (1 lt 2) and (1 lt 3) ;IDL prints 1

    >print, (1 lt 2) and (1 gt 3) ; IDL prints 0
  - or ; Returns true when either of its operands are true

    >print, (1 lt 2) or (1 gt 3) ;IDL prints 1

    >print, (1 gt 2) or (1 gt 3) ; IDL prints 0
  - not; Boolean inverse operator
  - xor; Boolean exclusive or function. Only good for certain data types
- Just about any logic making steps you need can be made with if's or's and and's.

THE END!