7.13. Glossary

System Message: WARNING/2
(/opt/web2py/applications/runestone/books/fopp/_sources/Iteration/Glossary.rst, line 18)

glossary seems to be misformatted, check indentation

System Message: WARNING/2
(/opt/web2py/applications/runestone/books/fopp/_sources/Iteration/Glossary.rst, line 19)

glossary seems to be misformatted, check indentation

System Message: WARNING/2
(/opt/web2py/applications/runestone/books/fopp/_sources/Iteration/Glossary.rst, line 20)

glossary seems to be misformatted, check indentation

Control Flow

Also known as the **flow of execution**, the order in which a program executes. By default, the control flow is *sequential*.

for loop traversal (`for`)

Traversing a string or a list means accessing each character in the string or item in the list, one at a time. For example, the following for loop:

```
for ix in 'Example':
    ...
```

executes the body of the loop 7 times with different values of `ix` each time.

ex A variable or value used to select a member of an ordered collection, such as a character from a string, or an element from a list.

p body The loop body contains the statements of the program that will be iterate through upon each loop. The loop body is always indented.

pattern

A sequence of statements, or a style of coding something that has general applicability in a number of different situations. Part of becoming a mature programmer is to learn and establish the patterns and algorithms that form your toolkit.

range

A function that produces a list of numbers. For example, `range(5)` , produces a list of five numbers, starting with 0, `[0, 1, 2, 3, 4]` .

sequential flow

The execution of a program from top to bottom, one statement at a time

terminating condition

A condition which stops an interation from continuing

traverse

To iterate through the elements of a collection, performing a similar operation on each.

✔ Completed. Well Done!

| Back to top