GEOS 436 / 636
Programming and Automation for Geoscientists

– Lecture 03: Flow Control –

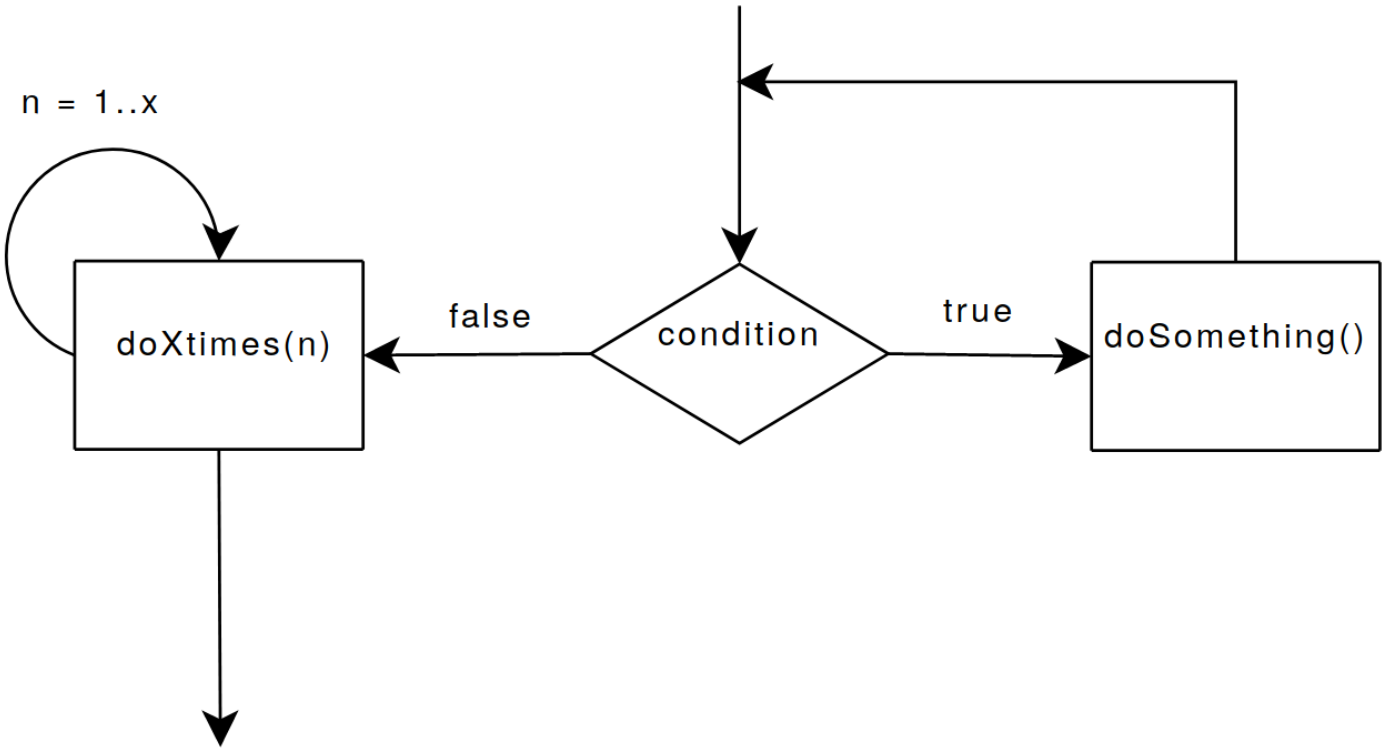
Ronni Grapenthin
rgrapenthin@alaska.edu
Elvey 413C

Flow Control: Redirect the Stream

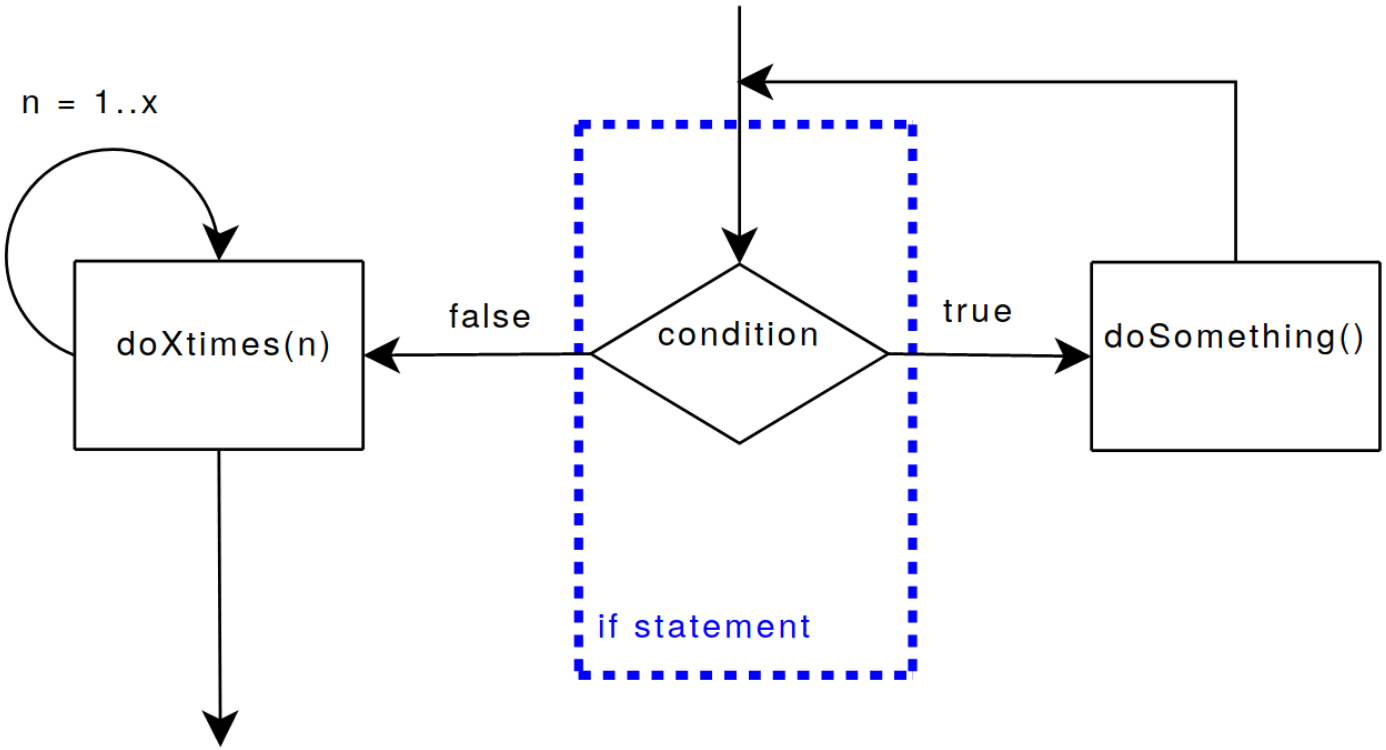
Flow control turns batch processing into programming

- (high level) programming languages allow different behavior based on **conditions** you define
- a condition can be `True` (1) or `False` (0)
- you test a condition using the operators: `<`, `<=`, `>`, `>=`, `==`, `!=` (actual symbols may differ in respective language.)
- some (built-in) functions implement tests of conditions, return truth value or `0,1` for instance `numpy.isfinite(A)` will return `True,False` depending on whether `A` is a finite number.
- the logic tables from last lecture will now be useful!

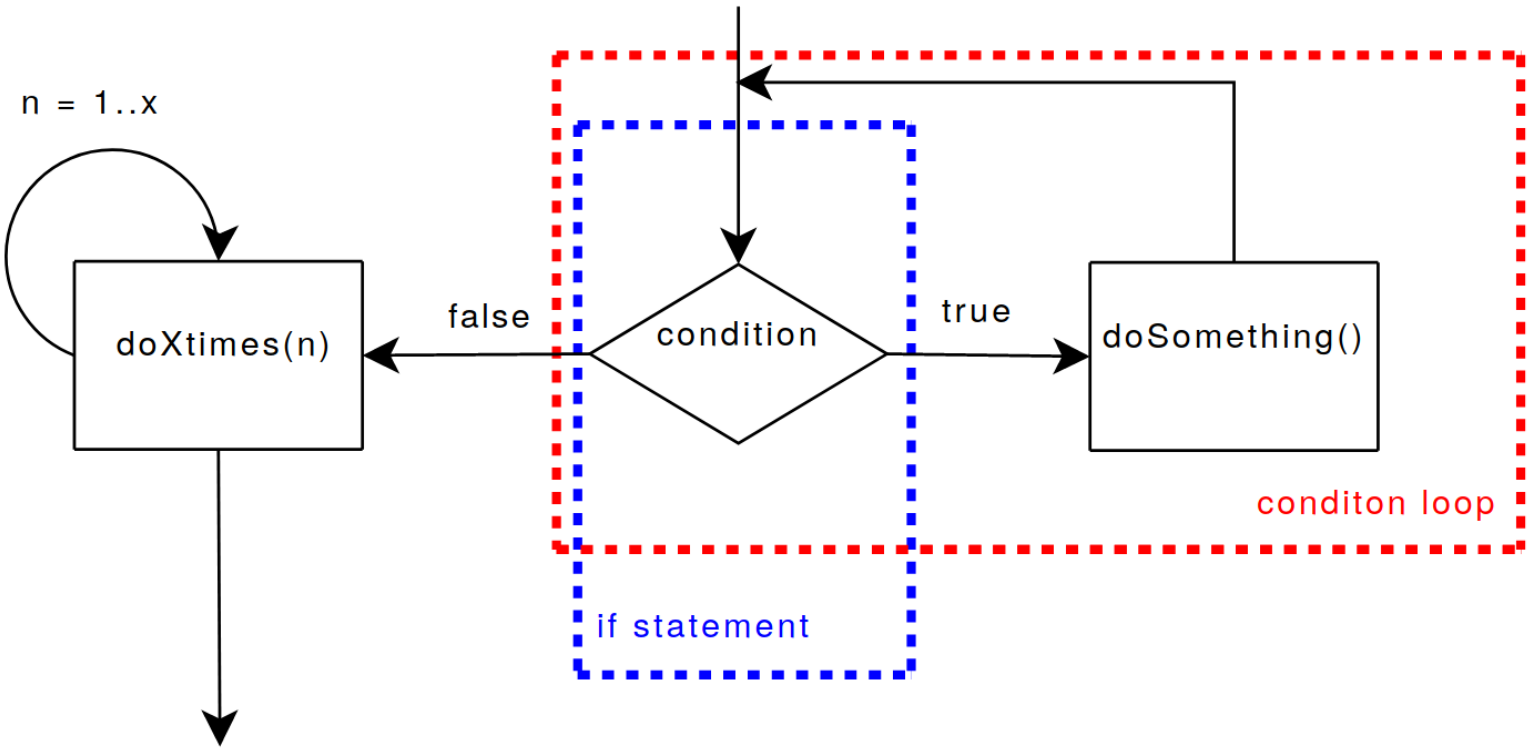
Flow Control



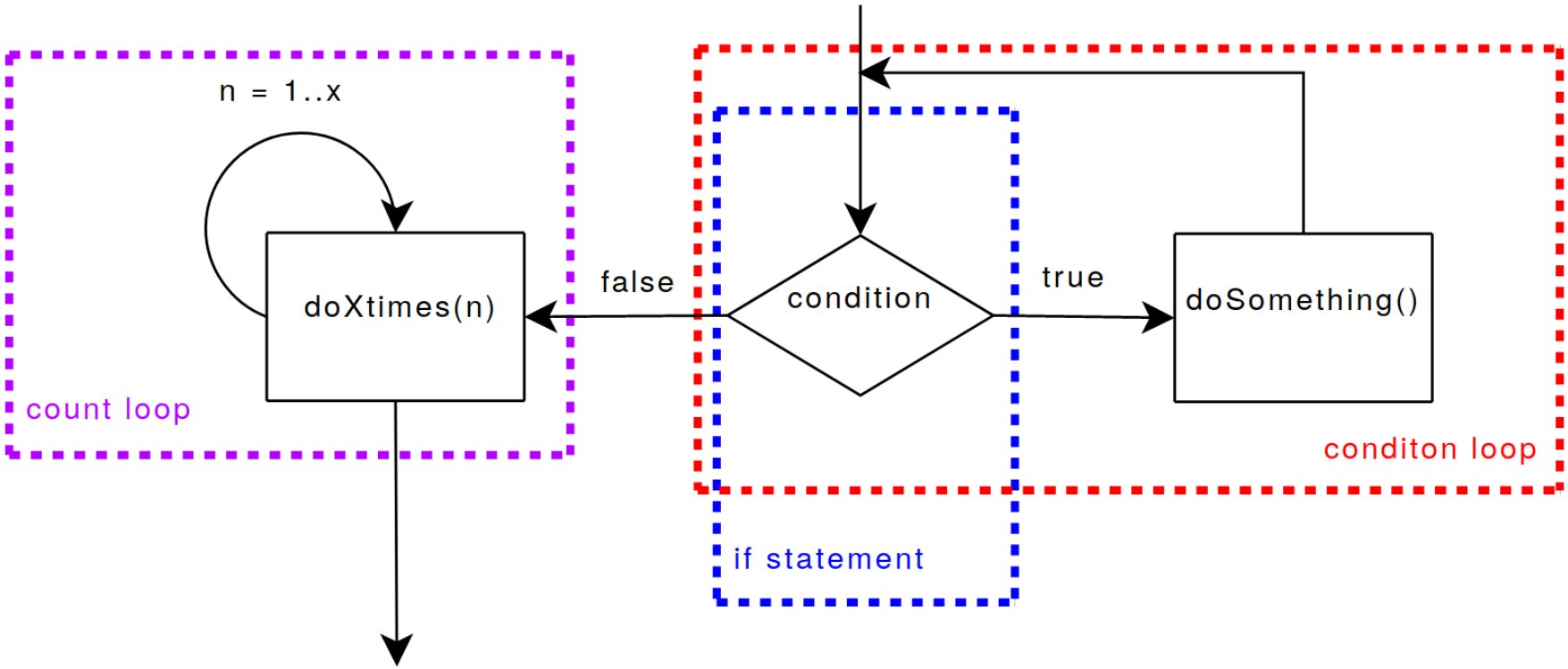
Flow Control



Flow Control



Flow Control



Statements

We need some formal background to keep this fairly general:

Formal language definitions

```
<block> ::= { <statement list> }.

<statement list> ::=
          <statement>
      |   <statement list> <statement>.

<statement> ::=
          <block>
      |   <assignment statement>
      |   <if statement>
      |   <for loop>
      |   <while loop>
      |   <do statement>
      |   . . .
```

'[' and ']' enclose optional statements

Conditions: `if ...then ...else ...`

Formal

```
1 <if statement> ::= if (<condition>) <statement> [else <statement>].
```

Conditions: `if ...then ...else ...`

Formal

```
1 <if statement> ::= if (<condition>) <statement> [else <statement>].
```

Python

```
  # if CONDITION:
2 #    STATEMENT
  # [elif CONDITION:
4 #    STATEMENT ]
  # [elif...]
6 # [else:
  #    STATEMENT ]
8 #
  import datetime
10
  #Get day of the week as an integer,
12 #Monday is 0, Sunday is 6.
  day = datetime.datetime.today().weekday()
14
  if day == 2:
16    print('New␣PAG␣Package')
  elif day == 5 or day == 6:
18    print('weekend')
  else:
20    print('Deal␣with␣other␣stuff.')
```

Conditions: `if ...then ...else ...`

Formal

```
1 <if statement> ::= if (<condition>) <statement> [else <statement>].
```

Python

```
  # if CONDITION:
2 #     STATEMENT
  # [elif CONDITION:
4 #     STATEMENT ]
  # [elif...]
6 # [else:
  #     STATEMENT ]
8 #
  import datetime
10
  #Get day of the week as an integer,
12 #Monday is 0, Sunday is 6.
  day = datetime.datetime.today().weekday()
14
  if day == 2:
16    print('New PAG Package')
  elif day == 5 or day == 6:
18    print('weekend')
  else:
20    print('Deal with other stuff.')
```

Shell

```
1 #!/bin/tcsh
  # if ( <condition> ) then <statement>
3 # [else <statement> ]
  # endif
5 #
  # Example: What are we gonna do today?
7
  set day = `date | awk '{print $1}'`
9
  if ($day == 'Wed' ) then
11    echo 'New PAG Package'
  else
13    if ($day == 'Sat' || \
          $day == 'Sun') then
15       echo 'weekend'
      else
17       echo 'Deal with other stuff'
      endif
19 endif
```

Condition Controlled Loop: `while`

Formal

```
1 <while loop> ::= while (<condition>) <block>.
```

Condition Controlled Loop: `while`

Formal

```
1 <while loop> ::= while (<condition>) <block>.
```

Python

```
  # while CONDITION:
2 #   STATEMENT
  #   [if CONDITION:
4 #       break]
  #
6 # EXAMPLE: Run until a new minute starts

8 import time, datetime

10 #infinite loop, unless there's a break
  #somewhere
12 while True:
      #use datetime module to get second
14    sec = datetime.datetime.now().second

16    print(sec)
      time.sleep(1)
18
      #not every while needs a break
20    if sec == 59:
          break
```

Condition Controlled Loop: `while`

Formal

```
1 <while loop> ::= while (<condition>) <block>.
```

Python

```
  # while CONDITION:
2 #   STATEMENT
  #   [if CONDITION:
4 #       break]
  #
6 # EXAMPLE: Run until a new minute starts

8 import time, datetime

10 #infinite loop, unless there's a break
  #somewhere
12 while True:
       #use datetime module to get second
14     sec = datetime.datetime.now().second

16     print(sec)
       time.sleep(1)
18
       #not every while needs a break
20     if sec == 59:
           break
```

Shell

```
1  #!/bin/tcsh
   # while ( <condition> ) <block>
3  #
   # Example: Run until a new minute starts
5
   #infinite loop unless there's a
7  #break somewhere
   while ( 1 )
9     #use unix date command
      #to get a the current second
11    set sec = `date +"%S"`

13    echo $sec
      sleep 1
15
      if($sec == 59) then
17       break
   end
```

Count Controlled Loop: `for`

Formal

```
1 <for loop> ::= for (<assignment>; <condition>; <assignment>) <block>.
```

Count Controlled Loop: `for`

Formal

```
<for loop> ::= for (<assignment>; <condition>; <assignment>) <block>.
```

Python

```
# for variable in sequence
#   STATEMENT
#
# EXAMPLE: countingm adding
#

import numpy as np

#from 2 to 10 in steps of 2
for x in range(2,11,2):
    print(x)
print("first 'for' done\n")

#sum of vector elements
s = 0
for x in np.array([23,1,5]):
   s = s + x

print(s)
```

Count Controlled Loop: `for`

Formal

```
<for loop> ::= for (<assignment>; <condition>; <assignment>) <block>.
```

Python

```
# for variable in sequence
#   STATEMENT
#
# EXAMPLE: countingm adding
#

import numpy as np

#from 2 to 10 in steps of 2
for x in range(2,11,2):
    print(x)
print("first 'for' done\n")

#sum of vector elements
s = 0
for x in np.array([23,1,5]):
   s = s + x

print(s)
```

Shell

```
#!/bin/tcsh
# foreach variable ( <list> ) <block>
#
# Example: counting, adding

#from 2 to 10 in steps of 2
foreach x (`seq 2 2 10`)
        echo $x
end
echo 'first `for` done'

#sum of vector elements
set s = 0
foreach x (23 1 5)
   @ s += $x
end

echo $s
```

Breaking out and Continuing Loops: `break`, `continue`

Python

```
# for variable in sequence
#   STATEMENT
#
# EXAMPLE: countingm adding
#

import numpy as np

for x in range(11):
    if x == 2:
        print("Two␣is␣prime!")
        continue

    if x == 5:
        print("I'll␣stop")
        break

    print("x␣=␣%d" % x)
```

Breaking out and Continuing Loops: `break`, `continue`

Python

```python
# for variable in sequence
#    STATEMENT
#
# EXAMPLE: countingm adding
#

import numpy as np

for x in range(11):
    if x == 2:
        print("Two is prime!")
        continue

    if x == 5:
        print("I'll stop")
        break

    print("x = %d" % x)
```

Shell

```
#!/bin/tcsh
# foreach variable ( <list> ) <block>
#
# Example: counting, adding

set cnt = 0

foreach x (`ls ./`)
  if ($x == foreach_break_example.csh) then
      echo "That's me: $x"
      continue # <-- omit everything below
  endif

  echo $x

  if ($cnt >= 10) then
      echo "I've got $cnt files, I am done."
      break #<-- exit the foreach NOW
  endif

  @ cnt += 1
end
```

Error Control: `try - except`

Formal

```
<try_catch> ::= try <block> catch <block>.
```

Error Control: `try - except`

Formal

```
<try_catch> ::= try <block> catch <block>.
```

Python

```
# try:
#   STATEMENT
# except [ErrorCode]:
#   STATEMENT
# [except [ErrorCode]:
#   STATEMENT]
# [finally:
#   STATEMENT]
#
# EXAMPLE: iterating and dividing
#

values = [1, 'x', 3]

for x in values:
  try:
    r = 1/x
    print("reciprocal of %d is %f" % (x, r))
  except Exception as e:
    print(e)
```