Repetition

Examples

- When is repetition necessary/useful?

Types of Loops

- Counting loop
  - Know how many times to loop
- Sentinel-controlled loop
  - Expect specific input value to end loop
- Endfile-controlled loop
  - End of data file is end of loop
- Input validation loop
  - Valid input ends loop
- General conditional loop
  - Repeat until condition is met

while

```
while condition:
   statements

x=1
while x < 10:
   print x
   x = x + 1
```

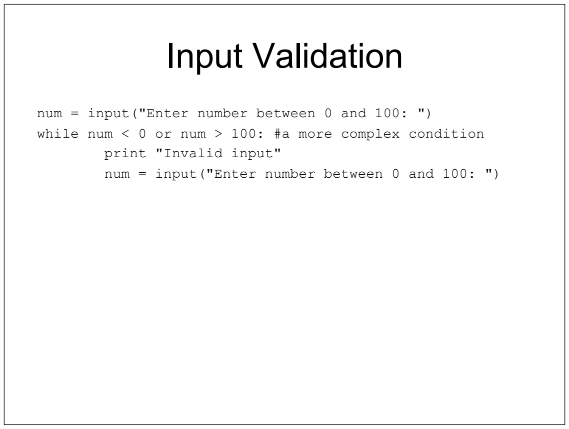
while

```
x=1 #initialization of control variable
while x < 10: #condition
   print x #task to be repeated
   x = x + 1 #update - VERY VERY IMPORTANT
```

Sentinel-controlled

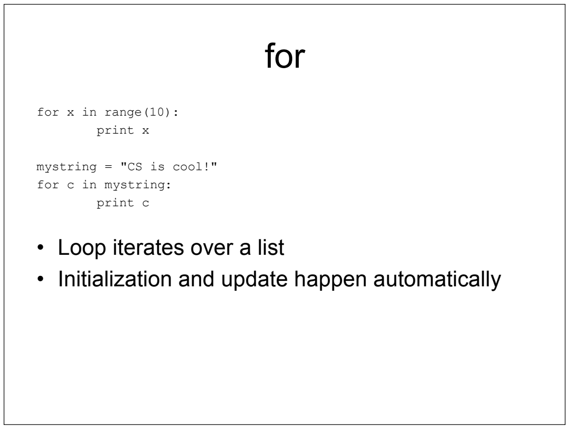
```
num = input("Enter number - 0 to quit: ")
while num != 0:
        print “You entered “, num
        num = input("Enter number - 0 to quit: ")
```

- Which is the control variable?



Input Validation

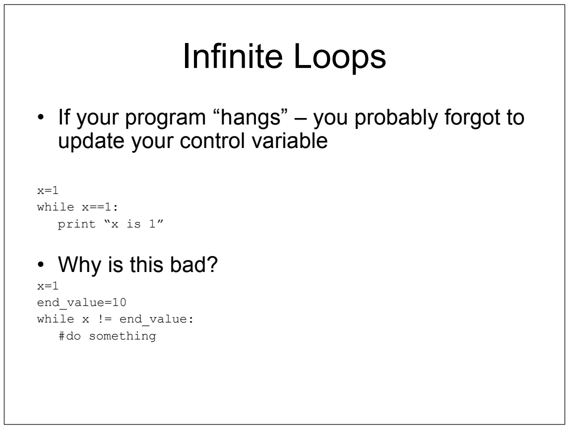
```
num = input("Enter number between 0 and 100: ")
while num < 0 or num > 100: #a more complex condition
        print "Invalid input"
        num = input("Enter number between 0 and 100: ")
```



for

```
for x in range(10):
        print x

mystring = "CS is cool!"
for c in mystring:
        print c
```

- Loop iterates over a list
- Initialization and update happen automatically



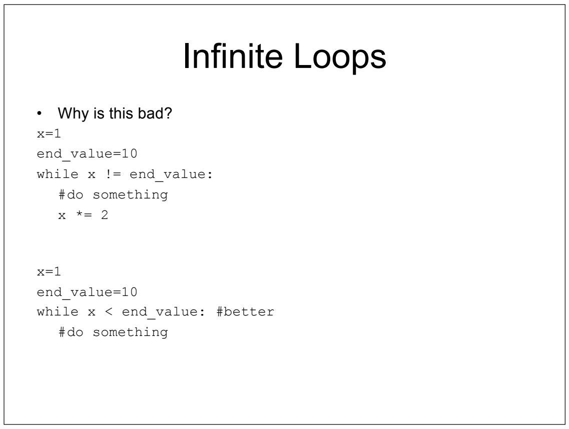
Infinite Loops

- If your program “hangs” – you probably forgot to update your control variable

```
x=1
while x==1:
   print “x is 1”
```

- Why is this bad?

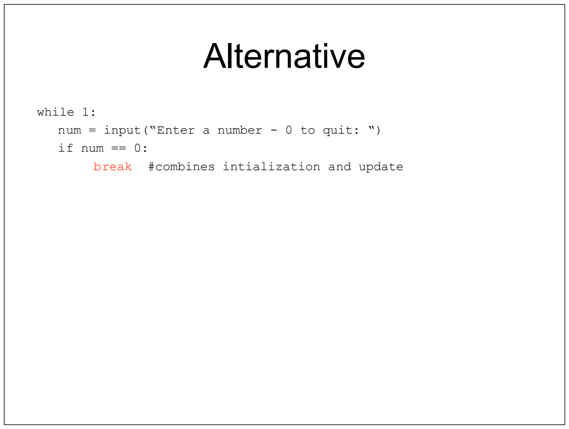
```
x=1
end_value=10
while x != end_value:
   #do something
```



Infinite Loops

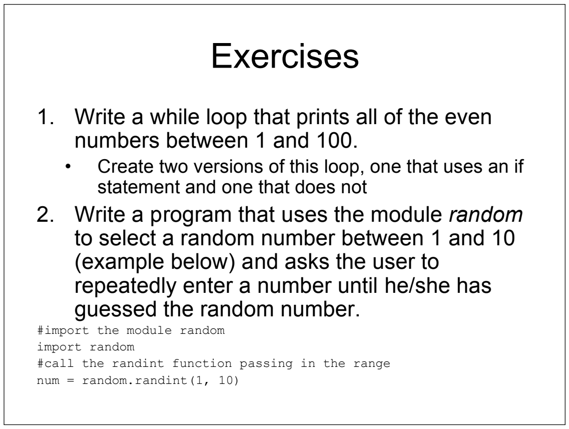
- Why is this bad?

```
x=1
end_value=10
while x != end_value:
   #do something
   x *= 2


x=1
end_value=10
while x < end_value: #better
   #do something
```



Alternative

```
while 1:
   num = input(“Enter a number - 0 to quit: “)
   if num == 0:
        break  #combines intialization and update
```



Exercises

1. Write a while loop that prints all of the even numbers between 1 and 100.
   - Create two versions of this loop, one that uses an if statement and one that does not
2. Write a program that uses the module *random* to select a random number between 1 and 10 (example below) and asks the user to repeatedly enter a number until he/she has guessed the random number.

```
#import the module random
import random
#call the randint function passing in the range
num = random.randint(1, 10)
```

Problem

- Print
```
*****
*****
*****
```
- The only print statements you can use are the following:
  - – print “*”, #the comma prevents the \n
  - – print

Nested Loops

#print a rectangle of stars

#3 times
  #print a line of stars

Nested Loops

```
#print a rectangle of stars
x=1
while x <= 3:
  #print a line of stars

#print a line of stars
y=1
while y<=3:
  print “*”,
```

Nested Loops

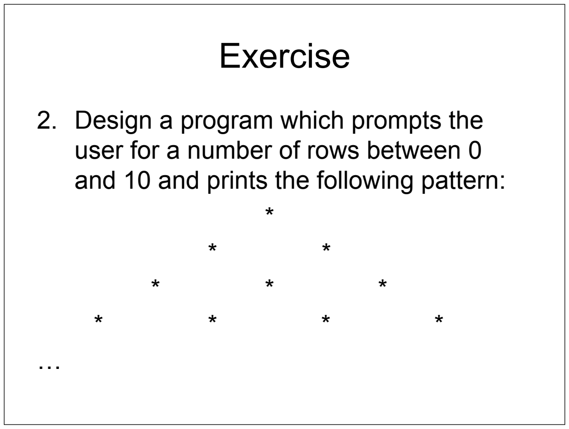
```
#print a rectangle of stars
x=1
while x <= 3:
  #print a line of stars
  y=1
  while y<=3:
     print “*”,

#DONE?
```

Nested Loops

```
#print a rectangle of stars
x=1
while x <= 3:
  #print a line of stars
  y=1
  while y<=3:
     print “*”,
     y+=1
  print
  x+=1
```

Exercise

1. Design a program which prompts the user for a number of rows between 0 and 10 and prints the following pattern:
```
****
***
**
*
```



Exercise

2. Design a program which prompts the user for a number of rows between 0 and 10 and prints the following pattern:

```
            *
        *       *
    *       *       *
*       *       *       *
```

…