More on Python Programming

J.K. Denny

Department of Mathematics, Mercer University

Review of `for` loops

```
# sum for ln(1+x) -- converges when -1<=x<1
x = 0.5
N = 10
sum = 0
for i in range(1,N):
    sum = sum + float((-1)**(i+1))*float(x)**i/float(i
print  (sum)
```

Using a loop: Truth tables

To define a Boolean variable in python, just assign a variable the value “True” or “False” with this exact capitalization. Then, we can operate on these values using “and,” “or,” and ”not.” For example, try typing the following in the python command window.

```
p = True
q = False
p and q
p or q
not(p)
```

Using a loop: Truth tables

How would you create a truth table for $\neg p \lor q$ with `for` loops?

```
for p in [True,False]:
  for q in [True,False]:
    print (p,q,not(p) or q)
```

Using a loop: Truth tables

How would you create a truth table for $\neg p \vee q$ with `for` loops?

```
for p in [True,False]:
  for q in [True,False]:
    print (p,q,not(p) or q)
```

Using a loop

On your own, create a truth table for

$$\sim p \wedge (q \vee \sim r)$$

using `for` loops.

Functions

Functions allow programmers to re-use bits of code after writing the code once. This is the basis of “modular design” in programming.

```
def f(n):
    sum=0
    for i in range(1,n+1):
        sum+=i
    return(sum)
```

Functions

Create a function to return the truth value of $p \wedge (q \vee \neg r)$.

```
def truth(p,q,r)
  return p and (q or not(r))
```

Functions

Create a function to return the truth value of $p \wedge (q \vee \neg r)$.

```
def truth(p,q,r)
  return p and (q or not(r))
```

Functions

Now, we combine `for` loops and this function to create a truth table.

```
for p in [True,False]:
  for q in [True,False]:
    for r in [True,False]
      print (p,q,r,truth(p,q,r))
```