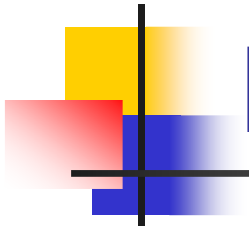
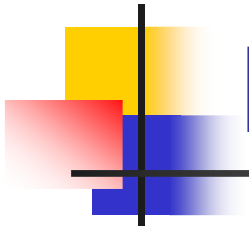
An Introduction to Python Programming:

Python Statements



Definite Loops

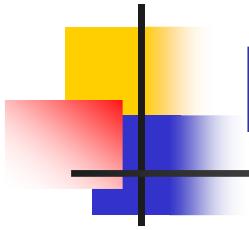
- A *definite* loop executes a definite number of times, i.e., at the time Python starts the loop it knows exactly how many *iterations* to do.
- for <var> in <sequence>:
  <body>
- The beginning and end of the body are indicated by indentation.



Definite Loops

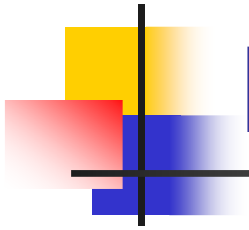
for <var> in <sequence>:
    <body>

- The variable after the *for* is called the *loop index*. It takes on each successive value in *sequence*.



Definite Loops

```
>>> for i in [0,1,2,3]:
        print i


0
1
2
3
>>> for odd in [1, 3, 5, 7]:
        print odd*odd


1
9
25
49
>>>
```



Definite Loops

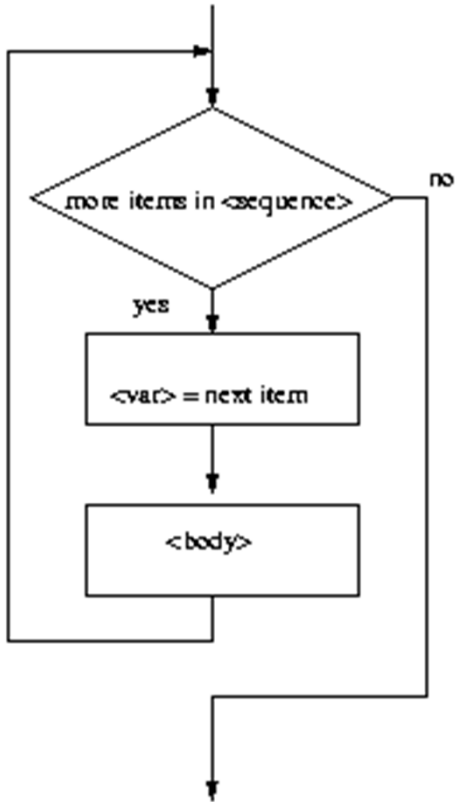
- In chaos.py, what did *range(10)* do?

```
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

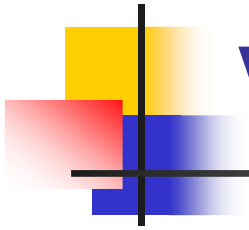
- Range is a built-in Python function that returns a list of numbers, starting with 0.
- Range(10) will make the body of the loop execute 10 times.

Definite Loops

- **for** loops alter the flow of program execution, so they are referred to as *control structures*.

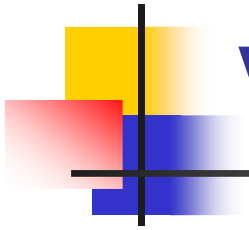


Example Program: Future Value



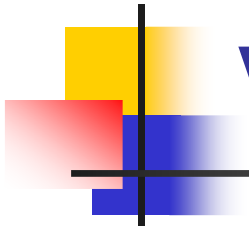
- Analysis
  - Money deposited in a bank account earns interest.
  - How much will the account be worth 10 years from now?
  - Inputs: principal, interest rate
  - Output: value of the investment in 10 years

Example Program: Future Value



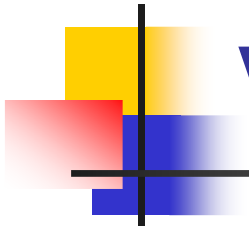
- Specification
  - User enters the initial amount to invest, the principal
  - User enters an annual percentage rate, the interest
  - The specifications can be represent like this …

Example Program: Future Value



- **Program** Future Value
- **Inputs**
  principal The amount of money being invested, in dollars
  apr The annual percentage rate expressed as a decimal number.
- **Output** The value of the investment 10 years in the future
- **Relatonship** Value after one year is given by *principal* * (1 + *apr*). This needs to be done 10 times.

Example Program: Future Value



- Design

Print an introduction

Input the amount of the principal (principal)

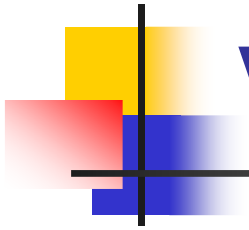
Input the annual percentage rate (apr)

Repeat 10 times:

    principal = principal * (1 + apr)

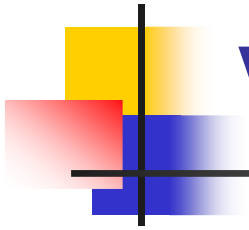
Output the value of principal

Example Program: Future Value



- Implementation
  - Each line translates to one line of Python (in this case)
  - Print an introduction
    print "This program calculates the future"
    print "value of a 10-year investment."
  - Input the amount of the principal
    principal = input("Enter the initial principal: ")

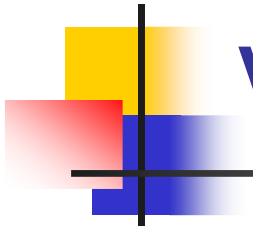
Example Program: Future Value



- Input the annual percentage rate
  apr = input(“Enter the annual interest rate: ”)
- Repeat 10 times:
  for i in range(10):
- Calculate principal = principal * (1 + apr)

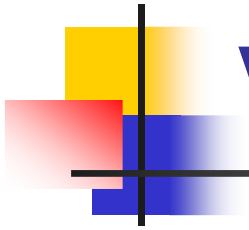
  principal = principal * (1 + apr)
- Output the value of the principal at the end of 10 years
  print “The value in 10 years is:”, principal

Example Program: Future Value

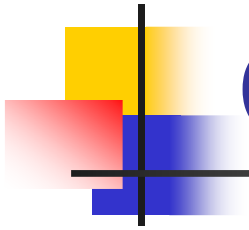


```
# futval.py
#    A program to compute the value of an investment
#    carried 10 years into the future

def main():
    print "This program calculates the future value of a 10-year investment."

    principal = input("Enter the initial principal: ")
    apr = input("Enter the annual interest rate: ")

    for i in range(10):
        principal = principal * (1 + apr)

    print "The value in 10 years is:", principal

main()
```

Example Program: Future Value



```
>>> main()
This program calculates the future value of a 10-year investment.
Enter the initial principal: 100
Enter the annual interest rate: .03
The value in 10 years is: 134.391637934
>>> main()
This program calculates the future value of a 10-year investment.
Enter the initial principal: 100
Enter the annual interest rate: .10
The value in 10 years is: 259.37424601
```



Conditional Statements

- Making Decisions in Python

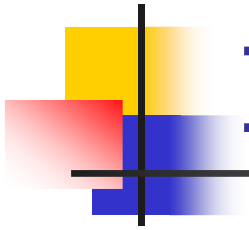
If, Eles, Elif

- The if statement of Python is similar to that of other languages.
- The if statement contains a logical expression using which data is compared, and a decision is made based on the result of the comparison.
- The syntax of the if statement is:

if expression:

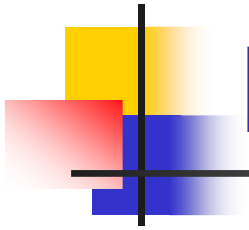
statement(s)

Note: In Python, all the statements indented by the same number of character spaces after a programming construct are considered to be part of a single block of code. Python uses indentation as its method of grouping statements.



If, Else, Elif

- An else statement can be combined with an if statement. An else statement contains the block of code that executes if the conditional expression in the if statement resolves to 0 or a false value.The *else statement is an optional statement and there could be at most only one else statement following an if .*
- The elif statement allows you to check multiple expressions for truth value and execute a block of code as soon as one of the conditions evaluates to true. Like the else, the elif statement is optional. However, unlike else, for which there can be at most one statement, there can be an arbitrary number of elif statements following an if.



Example

```
if expression1:
        statement(s)
elif expression2:
        statement(s)
elif expression3:
        statement(s)
else:
        statement(s)
```

While loop

- The while loop is one of the looping constructs available in Python. The while loop continues until the expression becomes false. The expression has to be a logical expression and must return either a *true or a false value*
- *The syntax of the while look is:*

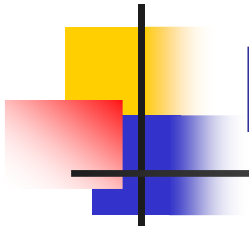
while expression:
    statement(s)
Example:

```
#!/usr/bin/python
count = 0
while (count < 9):
        print 'The count is:', count
        count = count + 1
print "Good bye!"
```

Infinite loop!

- Following loop will continue till you enter CNTL+C at the command prompt:

```
var = 1
while var == 1 :  # This constructs an infinite
   loop
        num = raw_input("Enter a number  :")
        print "You entered: ", num
print "Good bye!"
```



For Loop

- The for loop in Python has the ability to iterate over the items of any sequence, such as a list or a string.
- The syntax of the loop look is:

```
for iterating_var in sequence:
    statements(s)
```

- For a conventional loop through index system (ie., same as Fortran do I = 1, 10 or C for (i=1; I =<10;++) { use for I in range(1,n+1):
- NOTE the need to go 1 more

For Loop example

```
for letter in 'Python':     # First Example
    print 'Current Letter :', letter
fruits = ['banana', 'apple',  'mango']
for fruit in fruits:        # Second Example
        print 'Current fruit :', fruit
print "Good bye!”
```

This will produce following output:

```
Current Letter : P
Current Letter : y
Current Letter : t
Current Letter : h
Current Letter : o
Current Letter : n
Current fruit : banana
Current fruit : apple
Current fruit : mango
Good bye!
```