Python

Control Structures

Control Structures

- sequence
- conditional
- iteration / repetition / loop

First things first: Programming Style

- good programming style is paramount to software development
- improve code readability and thus maintainability
- Python's advantage: indentation-sensitive language
- other ways you can cultivate good programming style
  - annotate your code (appropriate comments)
  - descriptive/meaningful identifier names
  - effective use of white space (blank lines, space, indentation)
  - fixed width font (easy alignment)

Conditionals

- if some condition is fulfilled, take some action
- eg if it rains, carry an umbrella

if statement

- format

```
if <condition1>:
    <action1>
[elif <condition2>:
    <action2>]
...
[else:
    <actionn>]
```

eg

```
a = 3
b = 2
if a > b:
    print("greater")
```

if statement (contd)

- else

```
a = 3
b = 2
if a > b:
    print("greater")
else:
    print("less than")
```

- nested if-else

```
a = 3
b = 2
if a > b:
    print("greater")
elif a < b:
    print("less than")
else:
    print("equal")
```

- elif is shorthand for else if

Iteration / Repetition / Loop

- repeat a fixed pattern many times
- 2 types of loops in Python
  - for (fixed number of times)
  - while (variable number of times, 0 or more)

for statement

```
# print 1 to 10

for i in range(1,11):
    print(i)
```

for statement (contd)

```
# iterate over string

for letter in "Python":
    print(letter)
```

for statement (contd)

```
# compute sum of 1 to 10

sum = 0

for i in range(1, 11):
    print(i)
    sum = sum + i

print(sum)
```

Lists

- one of the most powerful and flexible data structure in Python to store a collection of (unlike) items
- say if you want to store 10 scores, you could use

```
s1, s2, s3, ... s10
```

- now what if you want to store 100 scores?
- or 1000000 scores?
- there must be a better way

Lists (contd)

```
scores = [] # empty list

scores = [89, 73, 100, 65, 54]

print(scores[0]) # 1st item
print(scores[1]) # 2nd item
print(scores[4]) # last item
print(scores[-1]) # last item

for item in scores:
    print(item)
```

Lists (contd)

- lists can comprise items of same data type (array)

```
fruits = ['apple', 'banana', 'mango']

for fruit in fruits:
    print(fruit)
```

Lists (contd)

- lists can comprise items of different/mixed data types

```
info = [123, 'Lim Ah Seng', 67.5]

for item in info:
    print(item)
```

while statement

```
# print from 1 to 10

i = 1
while (i <= 10):
    print(i)
    i = i + 1
```

while statement (contd)

```
# infinite loop, terminate with ctrl-c

i = 1
while i == 1:   # never ending
    num = input("Enter a number:")
    print("You entered:", num)
```

while statement

```
number = 23
running = True

while running:
    guess = int(input("Guess an integer: "))
    if guess == number:
        print("Bingo!")
        running = False # to end loop
    elif guess < number:
        print("Guess higher")
    else:
        print("Guess lower")
```

Exercise

Write a program to prompt the user for x positive integers, one at a time. Compute the sum of all x integers entered.

Use a list.

Exercise

Write a program to prompt the user for x positive integers, one at a time, until the user terminate with -1. Compute the sum of all x integers entered.

Extension: determine the largest and smallest numbers entered.

Practical 2

http://go.dhs.sg/y5cplab2

Due: Thu 7 Feb 2013