CS101: Intro to Computing
Fall 2015

Lecture 6

Administrivia

- Homework 4 is due ***tonight***
- Homework 5 assigned (due on Wed)

REVIEW

```
def fun(a):
  return a+2

x=fun(2)*fun(3)
```

What is the value of x?

a)9

b)4

c)16

d)None of the above.

```
def fun(m):
  return m.title().swapcase()

x=fun(“abb”)+fun(“acab”)
```

What is the value of x?

a)“AbbAcab”

b)“aBBaCAB”

c)“abbacab”

d)“ABBACAB”

```
def fun(a,b):
  c=((a+" ")*len(b)).title()

x=fun("ab","caa")
```

What is the value of x?

a) "ab ab ab"

b) "Ab Ab Ab"

c) "AB AB AB"

d) None of the above.

```
def fun(a,b):
  c=((a+" ")*len(b)).title()
  return c

x=fun("ab","caa")
```

What is the value of x?

a)"ab ab ab"

b)"Ab Ab Ab"

c)"AB AB AB"

WRITING FUNCTIONS

Defining functions

- We ***define*** a function by typing:

1. the keyword ***def***
2. the name of our function
3. a pair of parentheses
4. a ***block*** of code

```
def greetings():
  print “Hola!”
  print “Bonjour!”
  print “Ni hao!”
  print “Hello!”
  print “Shalom!”
  print “Guten tag!”
  print “Konnichiwa!”
  print “As-salamu alaykum!”
```

header

body

Block

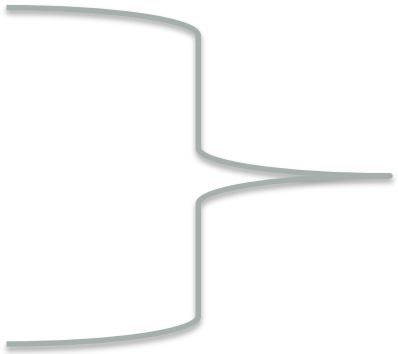
- A section of code grouped together
- Begins with a colon :
- Contents of the block are ***indented***
  - “Tabbed in”

```
def hello():
   print “hello”
```

Scope

- Variables declared ***inside*** a block are independent of variables ***outside*** the block.
- Variables inside a block ***do not exist*** outside the block.
- Blocks are their own little world!
- Blocks are ***isolated*** from the rest of our code.

```
a=5
def fun():
   a=3
   b=4
   a=a+b
fun()
print a
```



New scope

```
a=5
def fun():
    a=3
    b=4
    a=a+b
fun()
print a
```

Return

- Our function can ***return*** a value (output).
- We use the keyword ***return***.

```
def three():
   return 3
```

- Return ***immediately*** exits the function.

```
def hello():
   return 0
   print “hello”
```

Parameters

- Our function can take ***input*** (arguments) as well.
- Parameters are variables declared in function header.

```
def print_message(message):
    print message
```

- Multiple parameter are separated by commas.

```
def quadratic(a,b,c):
  s=(b**2-4*a*c)**.5
  d=2*a
  return (-b+s)/d
```

```
a=1
def fun(a,b):
  return a+b
a=fun(a,a)+a
```

What is the value of a?

a) 2

b) 3

c) 4

d) None of the above.

BOOLEAN TYPE

Booleans

- A type with only ***two values***:
  - **True** and **False**
- Used to represent ***logic***
- We'll use them to ***make decisions***.
- Based on ***Boolean algebra***
- Operators for Boolean type:
  - and, or, not

Logical operators

and	True	False
True	True	False
False	False	False

True when
BOTH inputs are True

or	True	False
True	True	True
False	True	False

True if
ONE input is True

Logical operators

not	result
True	False
False	True

Inverts the input

```
def fun():
  return True and False

x = fun() and not (True or False)
```

What is the value of x?

a) True

b) False

Comparison

- Operators that produce Boolean output

<	less than
<=	less than or equal <=
>	greater than
>=	greater than or equal
==	equal
!=	**not** equal

```
a=5
b=3

x=(a<5) and ((b<=5) or (a!=b))
```

What is the value of x?

a) True

b) False

```
a="HAWKEYE"
b="IRON MAN"

x=a<b and a[1]!=b[-2]
```

What is the value of x?

a) True

b) False

```
def fun(a,b):
  return a<b
a=3
b=4
x=fun(b,a)
```

What is the value of x?

a)True

b)False

CONDITIONAL EXECUTION

Conditional Execution

- Make decisions in our program
- Change program behavior
  - Based on a Boolean value
- Change the ***control flow***

If statement

- We create an ***if statement*** by typing:

1. the keyword ***if***
2. a Boolean expression
3. a ***block*** of code

```
print “Welcome to my program.”
input=raw_input(“Are you nice?”)
if input==“Yes”:
  print “Hello, friend!”
```

Alternative Execution

- Make decisions in our program
- Change program behavior
- Change the ***control flow***
- Execute one block OR another block

If… else statement

- We create an ***if… else statement*** with:

1. the keyword ***if***
2. a Boolean expression
3. a ***block*** of code
4. the keyword ***else***
5. another ***block*** of code

```
print “Welcome to my program.”
input=raw_input(“Are you nice?”)
if input==“Yes”:
  print “Hello, friend!”
else:
  print “HEY! BE NICE!”
```