

	Functions with Return Values CS303E: Elements of Computers and Programming July 13, 2012

	Announcements
	<ul style="list-style-type: none">■ Exam next Friday (7/20)<ul style="list-style-type: none">– Harder than the first exam!– Cumulative

	sys.stdout.write()
	<ul style="list-style-type: none">■ Recall that it allows you to print without a trailing space or newline■ Unlike print, it only accepts strings

	Functions: Review
	<ul style="list-style-type: none">■ Functions are useful:<ul style="list-style-type: none">– For carrying out actions that you will want to use multiple times– For grouping actions together for code readability■ Arguments allow us to pass different values to a function

Functions: Returning Values

- Functions may *return a value*
 - For example:
 - `seq=range(10)` #returns a list of values
 - `root=math.sqrt(9)` #returns the square root
 - `n=random.randint(1,100)` #returns a random int
 - In each case, the value returned is assigned to the left-hand-side variable (`seq`, `root`, `n`)
- Functions compute the value and then gives it back via a `return` statement

Syntax

To define a function and return a value:

```
def functionName(par1, par2):
```

`<code>`

```
    return someValue
```

← New!

To call this function:

```
result=functionName(arg1,arg2)
```

How It Works

- Given this call:


```
result=functionName(arg1,arg2)
```
- When the function is called:
 - `par1` set to `arg1`
 - `par2` set to `arg2`
 - The function is executed, and
 - The value that is returned by the function is assigned to `result`

Example

```
def times3(number):
    return 3*number
```

Function call:

```
result=times3(7) #result is 21
```

	How It Works
	You can also call a function that returns a value anywhere you would use a variable or value

	Example
	<pre>def times3(number): return 3*number</pre> <p>Function call:</p> <pre>if(x*3==times3(x)): print "They are equal!" #They are equal #The print statement executes</pre>

	iClicker Question
	<p>What is the expected output?</p> <pre>def addOne(x): return x+1</pre> <pre>result=addOne(3) print result</pre> <p>A. 2 C. 4 B. 3 D. Error</p>

	Multiple Return Statements
	<ul style="list-style-type: none"> ■ A function may have more than one return statement ■ Remember that execution returns to the calling function whenever a return statement is reached ■ Some consider this to be a bad programming practice, but it does have value

Multiple Returns: Example

```
#returns True if num is a power of 2
def powerOfTwo(num):
    while(num > 2):
        if(num%2!=0):
            return False
        num=num/2
    return True
```

Another Way: Flags

- A *flag* is a variable that is used to track some information
- Often used to know when to stop loops or exit a function

Flags: Example

```
def powerOfTwo_flags(num):
    poss_power=True #flag
    while(num>2 and poss_power==True):
        if(num%2!=0):
            poss_power=False
        num=num/2
    return poss_power
```

Good Programming Practice: Return Values

- In Python, a function may return more than one type of value
- Better to always return one type, so that the calling function can expect a particular type
 - For instance, `string.find()` returns a -1 instead of "Not Found"
 - Most languages require this

	Exercise
	<p>Write a function called <code>reverseIt()</code> that takes a string parameter and returns the reverse of the string. Write a main function that reads 5 strings from the user and prints the reverse of each string.</p>

	iClicker Question
<p>What is the output?</p> <pre>def addOne(x): return x+1 x=2 for i in range(3): for j in range(2): x=addOne(x) print x</pre>	<p>A. 6 B. 8 C. 9 D. 10</p>

	Another Note: Functions
	<ul style="list-style-type: none"> ■ Functions may be called from within functions ■ Realize: <ul style="list-style-type: none"> – <code>main()</code> is a function – We've called string functions from other functions – It works the same way!

	Exercise
	<ul style="list-style-type: none"> ■ Write a function that takes 3 numbers as arguments and returns the product of the three numbers. ■ Write a main function that calls this function for the following values and prints the result: <ul style="list-style-type: none"> 1, 2, 3 2, 3, 4 3, 4, 5, ... 20, 21, 22

	Gotchas!
	<ul style="list-style-type: none">■ ALL functions in Python return a value. If there is not a return statement, it returns the value <code>None</code>.■ When the execution reaches a return statement, execution immediately returns to the calling function