python™

Week 3

Parameters, return, math, graphics

Parameters

```
def name(parameter, parameter, ..., parameter):
    statements
```

– Parameters are declared by writing their names (no types)

```
>>> def print_many(word, n):
...     for i in range(n):
...         print word

>>> print_many("hello", 4)
hello
hello
hello
hello
```

Exercise

- Recreate the lines/boxes of stars example from lecture:

```
**********

*******

**********************************

**********
*        *
**********

*****
*   *
*   *
*****
```

Exercise Solution

stars.py

```
 1  # Draws a box of stars with the given width and height.
 2  def box(width, height):
 3      print width * "*"
 4      for i in range(height - 2):
 5          print "*" + (width - 2) * " " + "*"
 6      print width * "*"
 7
 8  # main
 9  print 13 * "*"
10  print  7 * "*"
11  print 35 * "*"
12  box(10, 3)
13  box(5, 4)
```

Default Parameter Values

`def` **name(parameter=value, ..., parameter=value):**
statements

- Can make parameter(s) optional by specifying a default value

```
>>> def print_many(word, n=1):
...     for i in range(n):
...         print word

>>> print_many("shrubbery")
shrubbery
>>> print_many("shrubbery", 4)
shrubbery
shrubbery
shrubbery
shrubbery
```

- **Exercise:** Modify `stars.py` to add an optional parameter for the character to use for the outline of the box (default `"*"`).

python™

Parameter Keywords

name(parameter=value, ..., parameter=value)

- Can specify the names of parameters as you call a function
- This allows you to pass the parameters in any order

```
>>> def print_many(word, n):
...     for i in range(n):
...         print word

>>> print_many(str="shrubbery", n=4)
shrubbery
shrubbery
shrubbery
shrubbery
>>> print_many(n=3, str="Ni!")
Ni!
Ni!
Ni!
```

Math commands

```
from math import *
```

Function name	Description
`ceil(`**value**`)`	rounds up
`cos(`**value**`)`	cosine, in radians
`degrees(`**value**`)`	convert radians to degrees
`floor(`**value**`)`	rounds down
`log(`**value**`, `**base**`)`	logarithm in any base
`log10(`**value**`)`	logarithm, base 10
`max(`**value1**`, `**value2**`, ...)`	largest of two (or more) values
`min(`**value1**`, `**value2**`, ...)`	smallest of two (or more) values
`radians(`**value**`)`	convert degrees to radians
`round(`**value**`)`	nearest whole number
`sin(`**value**`)`	sine, in radians
`sqrt(`**value**`)`	square root
`tan(`**value**`)`	tangent

Constant	Description
`e`	2.7182818...
`pi`	3.1415926...

Returning Values

```
def name(parameters):
    statements
    ...
    return value
```

```
>>> def ftoc(temp):
...     tempc = 5.0 / 9.0 * (temp - 32)
...     return tempc

>>> ftoc(98.6)
37.0
```

python™