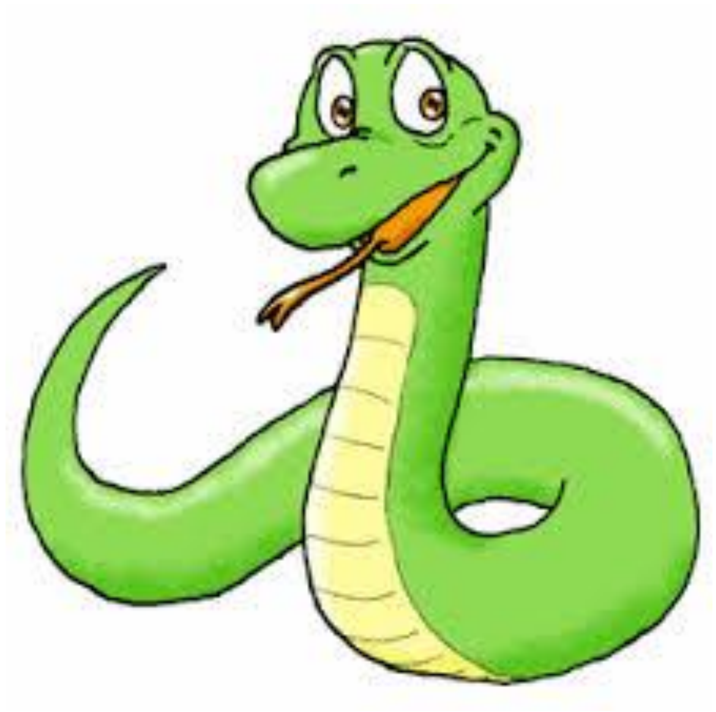
Application of GIS with Python

Chapter 2 :Introduction to python Function



https://www.python.org/

http://www.tutorialspoint.com/python/

Table of Contents

Python functions

- A function is a portion of code, which performs a specific task.
  1. Functions (may) have a name.
  2. Functions may need arguments.
  3. Functions may produce a result.
- A function may be called.

  Example:

```
>>> len('Python programming')
18
```

‘Python programming’	Argument
len	Function
18	Return value

Why functions?

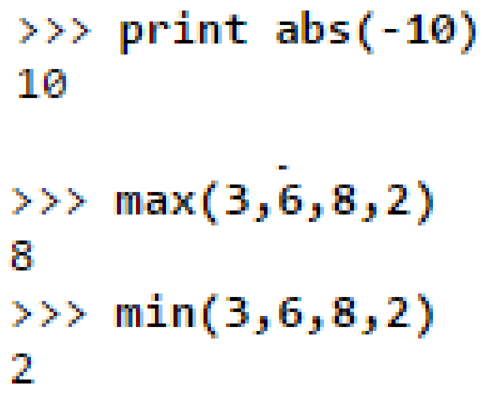
Functions are useful!

➢Functions:

1. Group statements
2. Eliminate repetitive code
3. Cut large programs in smaller bits
4. Allow re-use of fruitful functions

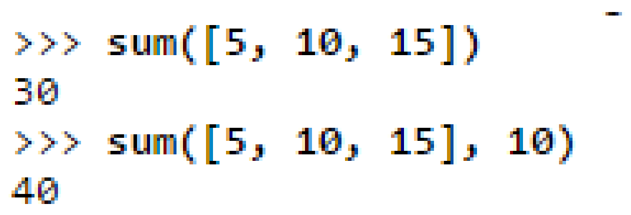
Built in functions

- ➢The Python interpreter has a number of functions built into it that are always available.



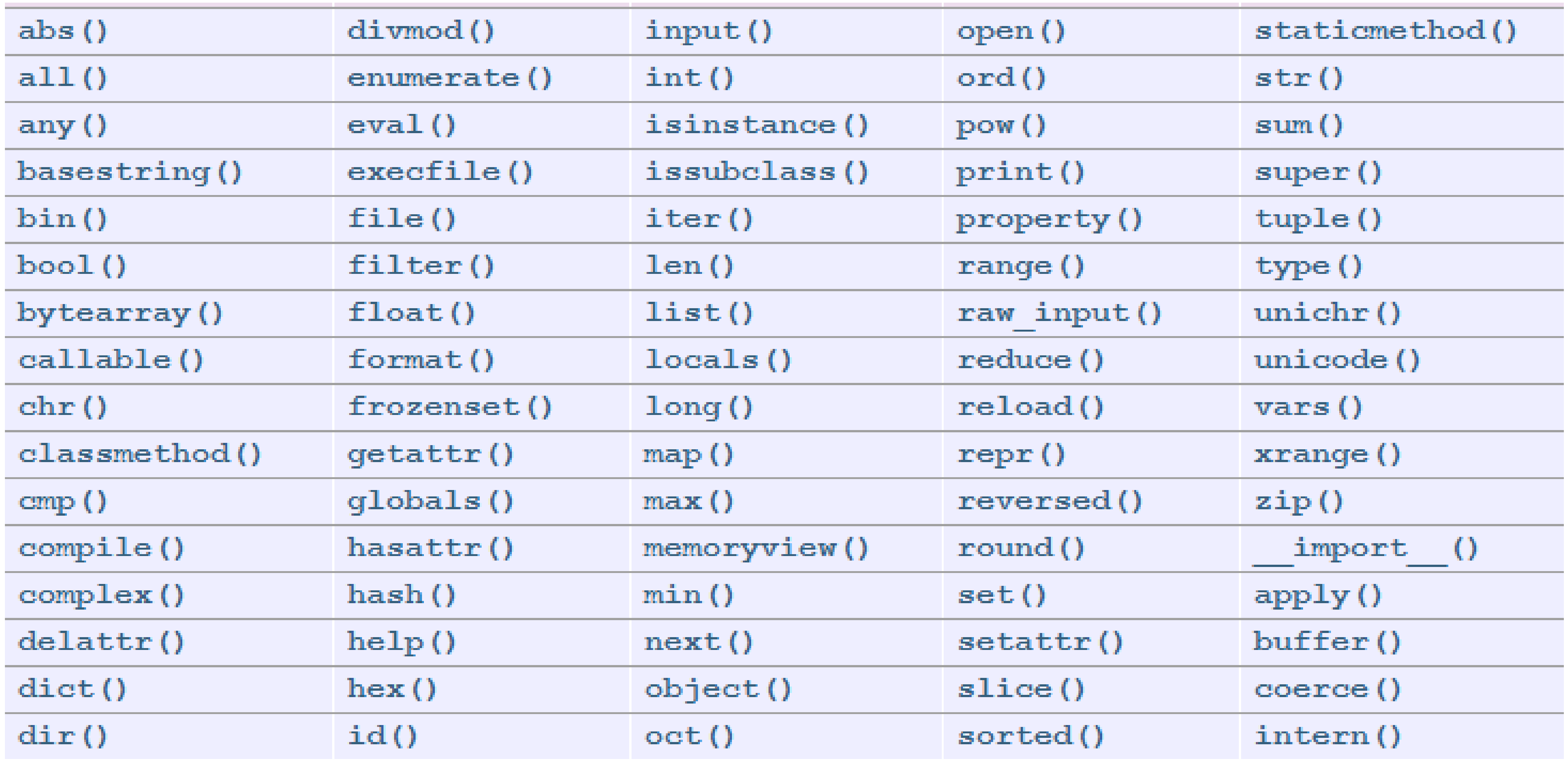
```
>>> print abs(-10)
10

>>> max(3,6,8,2)
8
>>> min(3,6,8,2)
2
```



```
>>> sum([5, 10, 15])
30
>>> sum([5, 10, 15], 10)
40
```

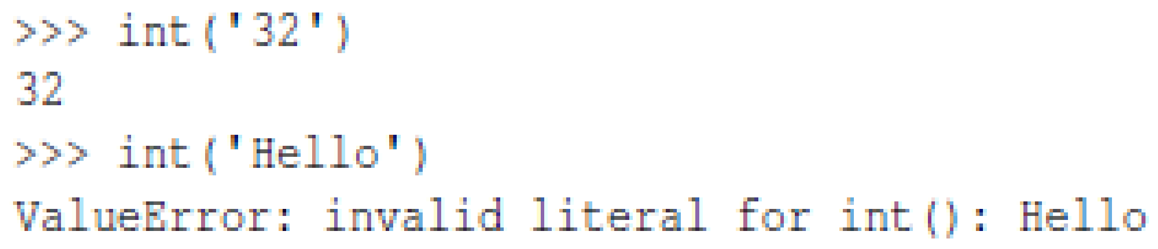
Built in functions



abs()	divmod()	input()	open()	staticmethod()
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	__import__()
complex()	hash()	min()	set()	apply()
delattr()	help()	next()	setattr()	buffer()
dict()	hex()	object()	slice()	coerce()
dir()	id()	oct()	sorted()	intern()

Conversion functions

- ➢Python provides built-in functions that convert values from one type to another.
- ➢For a case the int function takes any value and converts it to an integer, if it can, or complains otherwise



```
>>> int('32')
32
>>> int('Hello')
ValueError: invalid literal for int(): Hello
```

Conversion functions

- bool(x)
- chr(i)
- complex(real, imag)
- dict(sequence)
- float(x)
- int(x)
- hex(x)
- list(sequence)
- long(x)
- oct(x)
- ord(c)
- repr(object)
- round(x)
- set(iterable)
- str(object)
- tuple(sequence)

Conversion functions

```
>>> float(32)                          # int to float
32.0
>>> float(’3.14159’)                   # str to float
3.14159
>>> str(32)                            # int to str
’32’
>>> str(3.14149)                       # float to str
’3.14149’
```

Math functions

- Math functions from math module
  - Module is a file that contains a collection of related functions, statements, variables.
- Before use of the module, has to be imported
  - >>> import math **#creates a module object named math**.
- These functions must be accessed using the dot notation:
  - >>> math.sin(math.radians(45))
  - 0.7071067811865476

#Object.functionname(argument)

Composition

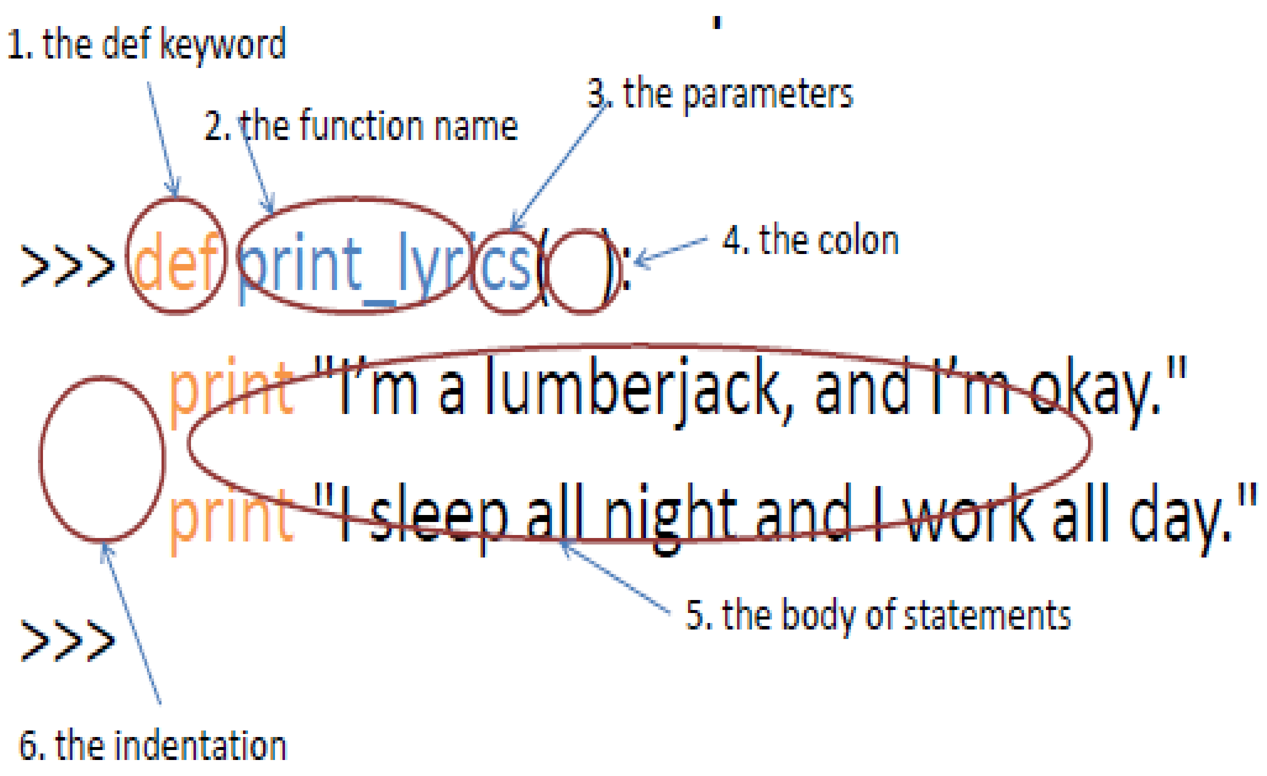
```
>>> import math
>>> d = 45
>>> r = math.radians(d)
>>> s = math.sin(r)
>>> print s
0.707106781187
>>> math.sin(math.radians(45))
0.7071067811865476
```

Function definition

1. The keyword def introduces a function definition.

   It must be followed by:
2. The function name
3. The parenthesized list of parameters
4. A colon ':'
5. The statements that form the body of

   the function.
6. These statements are indented!



```
>>> def print_lyrics( ):
        print "I'm a lumberjack, and I'm okay."
        print "I sleep all night and I work all day."
>>>
```

Readability

- Perhaps Python's most controversial feature is its use of indentation for statement grouping.
- Indentation makes Python code more readable in two ways:
  - It reduces visual clutter and makes programs shorter.
  - It allows less freedom in formatting, thereby enabling a more uniform style.

Argument and Parameter

- Arguments are passed to variables called parameters.

```
>>> def print_twice(x):      Parameter(Parameters are local!)
        print x
        print x
>>> print_twice('Hello World!')         Argument
Hello World!
Hello World!
```

Fruitful & Void functions

- A function may return a value by using the return keyword, such function is fruitful function

```
>>> def double_it(x):
        return x + x
>>> double_it(23)
46
>>> double_it('Spam')
'SpamSpam'
```

- A function may perform an action but doesn’t return value such function is void function or procedures

```
>>> def double_it(x):
        print x + x
>>> a=double_it(23)
>>> print a
None
```

Recursive Function

- Sometimes it's useful to define a function that calls itself.

```
>>> def count_down(n):
print n
count_down(n-1)
```

- Such a function is said to be recursive.

```
>>> count_down(4)
     4
     3
     2
     1
     0
     -1
     -2
     -3 ...
```

Whoops!

Our function doesn't stop at 0.

- An important part of a recursive function is a condition that stops the recursion.

```
>>> def
count_down(n):
        if n >= 0:
        print n
        count_down(n-1)
```

```
>>> count_down(4)
    4
    3
    2
    1
    0
    The recursion stops!
```

Some Python Modules

>>> import random

>>> help(random)

...

>>> help(random.random)

...

>>> random.random()

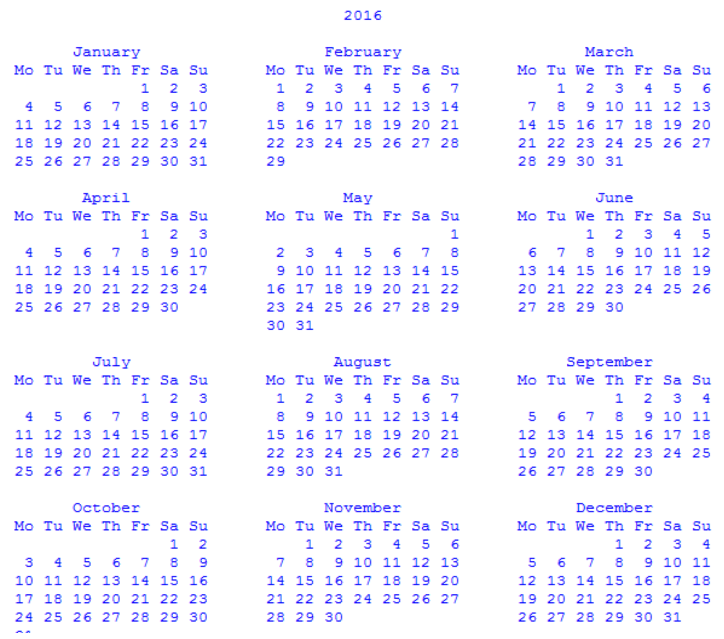
0.22719403737126942

The random() function returns random numbers

between 0 and 1.

>>> import calendar

>>> print calendar.calendar(2016)



```
                                  2016

      January                   February                   March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
             1  2  3       1  2  3  4  5  6  7          1  2  3  4  5  6
 4  5  6  7  8  9 10       8  9 10 11 12 13 14       7  8  9 10 11 12 13
11 12 13 14 15 16 17      15 16 17 18 19 20 21      14 15 16 17 18 19 20
18 19 20 21 22 23 24      22 23 24 25 26 27 28      21 22 23 24 25 26 27
25 26 27 28 29 30 31      29                        28 29 30 31

       April                      May                       June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
             1  2  3                         1             1  2  3  4  5
 4  5  6  7  8  9 10       2  3  4  5  6  7  8       6  7  8  9 10 11 12
11 12 13 14 15 16 17       9 10 11 12 13 14 15      13 14 15 16 17 18 19
18 19 20 21 22 23 24      16 17 18 19 20 21 22      20 21 22 23 24 25 26
25 26 27 28 29 30         23 24 25 26 27 28 29      27 28 29 30
                          30 31

        July                     August                  September
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
             1  2  3       1  2  3  4  5  6  7                1  2  3  4
 4  5  6  7  8  9 10       8  9 10 11 12 13 14       5  6  7  8  9 10 11
11 12 13 14 15 16 17      15 16 17 18 19 20 21      12 13 14 15 16 17 18
18 19 20 21 22 23 24      22 23 24 25 26 27 28      19 20 21 22 23 24 25
25 26 27 28 29 30 31      29 30 31                  26 27 28 29 30

      October                   November                  December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
                1  2          1  2  3  4  5  6                1  2  3  4
 3  4  5  6  7  8  9       7  8  9 10 11 12 13       5  6  7  8  9 10 11
10 11 12 13 14 15 16      14 15 16 17 18 19 20      12 13 14 15 16 17 18
17 18 19 20 21 22 23      21 22 23 24 25 26 27      19 20 21 22 23 24 25
24 25 26 27 28 29 30      28 29 30                  26 27 28 29 30 31
```

Day of the Week

- On what day was Albert Einstein born?

```
>>> calendar.weekday(1879, 3, 14)
4
```

14 March 1879

was a Friday!

Index	Day
0	Mon
1	Tue
2	Wed
3	Thu
4	Fri
5	Sat
6	Sun

To see what’s in the standard library of modules, check out the Python Library Reference: http://docs.python.org/lib/lib.html

Assignment 2 :

1. What do you understand by Standard library for python module?
2. What is function in python programming and what function can do in python programming?
3. What do you understand by recursive function in python programming? Highlight your answer with python programming using recursive function.
4. Write Short notes on following with suitable python programming example:
   - Fruitful and Void Function
   - Built in function
   - Math Function
   - Conversion function