University of
South Australia

Problem Solving and Programming

The Python Standard Library

University of South Australia

Copyright Notice

Do not remove this notice.

COMMONWEALTH OF AUSTRALIA
Copyright Regulations 1969

WARNING

This material has been produced and communicated to you by or on behalf of the University of South Australia pursuant to Part VB of the *Copyright Act 1968* (**the Act**).

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice.

Recap exercise…

- Assign the value 10 to a variable called width.

Recap exercise…

- Assign the value 10 to a variable called width.
- Assign the value 5 to a variable called height.

Recap exercise…

- Assign the value 10 to a variable called width.
- Assign the value 5 to a variable called height.
- Define a variable called area to store the result of width * height.

Recap exercise…

- Assign the value 10 to a variable called width.
- Assign the value 5 to a variable called height.
- Define a variable called area to store the result of width * height.
- Display the area to the screen.

- Solution…

Recap exercise…

- Assign the value 10 to a variable called width.
- Assign the value 5 to a variable called height.
- Define a variable called area to store the result of width * height.
- Display the area to the screen.

- Solution…

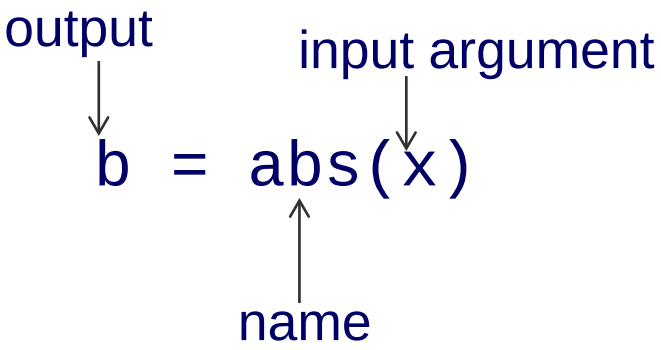
```
width = 10
height = 5
area = width * height
print(area)
```

UniSA

The Python Standard Library

- Built-in functions – extensive library
  - The Python interpreter has a number of functions that are always available.
  - They are available without having to import a library.
  - We will explore a few of the built-in functions.

- Functions consist of three components:
  - Name
  - Input argument(s)
  - Output



- The input (also called argument/parameter) goes inside the parentheses.
- The output is the returned value.

Built-in Python Functions

- Some functions require multiple inputs.

  For example:

  The pow function returns x to the power of y.

  `pow(x, y)`

  `pow(2,3)`

  `pow(2,3)` calculates 2 to the power of 3 = 8

- You can assign a name to the output (variable):

  `result = pow(2,3)`

  In this case, the name used to refer to the output is the variable name `result`.

Built-in Python Functions

- A few useful built-in functions:

Function	Description	Example
abs(x)	Returns the absolute value of x.	abs(-3) ans = 3
pow(x,y)	Returns x to the power y.	pow(5, 2) ans = 25
round(x[, *n*])	Returns the floating point value x rounded to *n* digits after the decimal point. If *n* is omitted, it defaults to zero.	round(1.6) ans = 2
int([number|string[, base]])	Converts a number or string to an integer. If no arguments are given, returns 0.	int('7') ans = 7
float([x])	Converts a string or number to floating point.	float('1.5') ans = 1.5

Built-in Python Functions

- A few useful built-in functions:

`range([start], stop[, step])`

Constructs progression of integer values, most often used in for loops. The arguments must be integers. The step value defaults to 1 if omitted. If the start argument is omitted, it defaults to 0.

`str(x)`

Converts object x to a string representation.

Built-in Python Functions

- A few useful built-in functions:

```
print([object, ...], sep=' ', end='\n', file=sys.stdout)
```

Print *object*(s) to the stream *file*, separated by *sep* and followed by *end*. *sep*, *end* and *file*, if present, must be given as keyword arguments.

Examples:

```
ans = 7
print("This is an example!")
print("This is another example:", 777)
print(ans)
print("The answer is:", ans)
print("i", "like", "donuts!")
print("i", "like", "donuts!", sep='-')
print("There are", ans, "donuts left...")
```

Built-in Python Functions

- A few useful built-in functions:

`input([prompt])`

If the *prompt* argument is present, it is written to standard output without a trailing newline. The function then reads a line from input, converts it to a string (stripping a trailing newline), and returns that.

Examples:

```
guess = input('Please enter your guess: ')
name  = input('Please enter your name: ')
number = int(input('Enter number: '))
```

Using the Help feature (Python Docs)

- Python has an extensive library of functions.
- We can not possibly teach you all of them (you would go crazy and so would we)!
- Use the help feature to help you use Python's functions (graduate quality - life long learning).
- Access the help feature from:
  - The Python Shell (IDLE).
  - Select help on the menu bar.
  - Then select Python Docs.
- Standard Library documentation also on the web
- http://docs.python.org/py3k/library/index.html

Numeric and Mathematical Modules

`math` – Mathematical functions

- Provide access to the mathematical functions.
- Need to place the following import statement at the top of your program.

```
import math
```

Numeric and Mathematical Modules

`math` – Mathematical functions

Function	Description
math.ceil(x)	Returns the ceiling of x, the smallest integer greater than or equal to x.
math.floor(x)	Returns the floor of x, the largest integer less than or equal to x.
math.sqrt(x)	Returns the square root of x.
math.cos(x)	Returns the cosine of x radians.
math.sin(x)	Return the sine of x radians.
math.tan(x)	Returns the tangent of x radians.
math.degrees(x)	Converts angle x from radians to degrees.
math.radians(x)	Converts angle x from degrees to radians.
math.pi	The mathematical constant pi.

Numeric and Mathematical Modules

`random` – generate pseudo-random numbers

- It provides access to random number generator.
- Need to place the following import statement at the top of your program.

```
import random
```

Function	Description
random.randint(a, b)	Returns a random integer N such that a <= N <= b.
random.choice(seq)	Returns a random element from the non-empty sequence seq.
random.random()	Returns the next random floating point number in the range 0.0, 1.0.
random.shuffle(x)	Shuffle the sequence x in place.

End

Python Standard Library