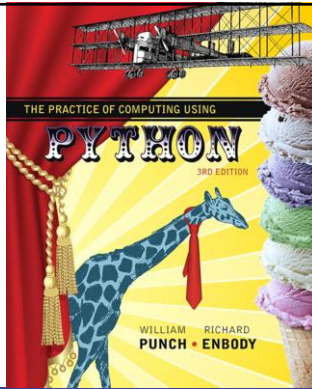


## chapter 5

## Functions -- QuickStart

PEARSON

ALWAYS LEARNING



## What is a function?

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Functions

- from Mathematics we know that functions perform some operation and return one value.
- they "encapsulate" the performance of some particular operation, so it can be used by others (for example, the `sqrt()` function)



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Why have them?

- support divide-and-conquer strategy
- abstraction of an operation
- reuse: once written, use again
- sharing: if tested, others can use
- security: well tested, then secure for reuse
- simplify code: more readable



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Mathematical Notation

- consider a function which converts temperatures in Celsius to temperatures in Fahrenheit
  - formula:  $F = C * 1.8 + 32.0$
  - functional notation:  
 $F = \text{celsius\_to\_Fahrenheit}(C)$  where  
 $\text{celsius\_to\_Fahrenheit}(C) = C * 1.8 + 32.0$



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Python Invocation

- math:  $F = \text{celsius\_to\_Fahrenheit}(C)$
  - in Python, the invocation is much the same:  
`F = celsius_to_Fahrenheit(cel_float)`
- terminology: `cel_float` is the **argument**



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Function Definition

- math:  $g(C) = C * 1.8 + 32.0$
- Python
 

```
def celsius_to_Fahrenheit(param_float):
    return param_float * 1.8 + 32.0
```
- terminology: `param_float` is the *parameter*



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

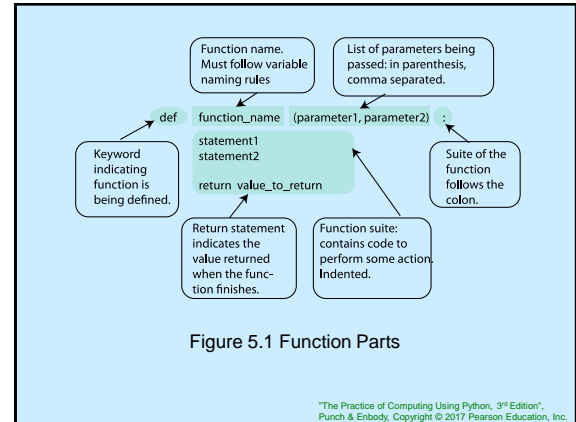


Figure 5.1 Function Parts

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## return Statement

- the **return** statement indicates the value that is returned by the function
- the statement is optional (the function can return nothing)
  - if no **return**, function is often called a procedure



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

```
1 # Temperature conversion
2
3 def celsius_to_fahrenheit(celsius_float):
4     """ Convert Celsius to Fahrenheit. """
5     return celsius_float * 1.8 + 32
```

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

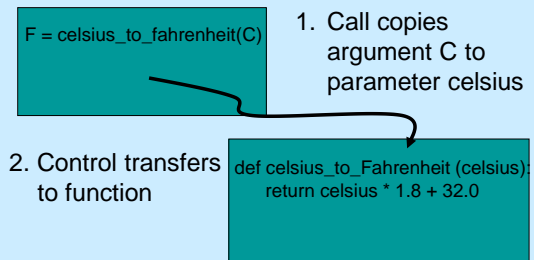
## Triple Quoted String in Function

- a triple quoted string just after the **def** is called a **docstring**
- docstring is documentation of the function's purpose, to be used by other tools to tell the user what the function is used for
  - more later



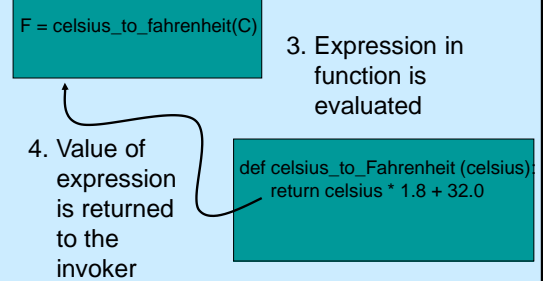
"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Operation



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Operation (con't)



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

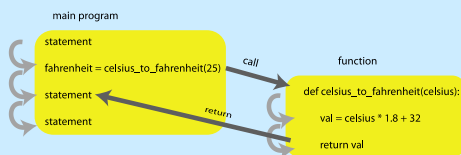


Figure 5.1 Function flow of control

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Code Listing 5.2 Full Temp Program

```
1 # Conversion program
2
3 def celsius_to_fahrenheit(celsius_float):
4     """ Convert Celsius to Fahrenheit. """
5     return celsius_float * 1.8 + 32
6
7 # main part of the program
8 print("Convert Celsius to Fahrenheit.")
9 celsius_float = float(input("Enter a Celsius temp: "))
10 # call the conversion function
11 fahrenheit_float = celsius_to_fahrenheit(celsius_float)
12 # print the returned value
13 print(celsius_float, " converts to ", fahrenheit_float, " Fahrenheit")
```

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Code Listing 5.3 digit extraction

```
def get_digit(number, position):
    '''return digit at position in number, counting from right'''
    return number//(10**position)%10
```

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Area of a Triangle

- the next few functions can be used together to find the area of a triangle
- note how we decompose the problem and then re-assemble the overall solution using the functions created



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

### Code Listing 5.4 Input

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

```
def get_vertex():
    x = float(input("    Please enter x: "))
    y = float(input("    Please enter y: "))
    return x,y
```

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

### Code Listing 5.5 get\_triangle

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

```
def get_triangle():
    print("First vertex")
    x1,y1 = get_vertex()
    print("Second vertex")
    x2,y2 = get_vertex()
    print("Third vertex")
    x3,y3 = get_vertex()
    return x1, y1, x2, y2, x3, y3
```

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

### Code Listing 5.6

#### side\_length

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

```
def side_length(x1,y1,x2,y2):
    ''' return length of a side (Euclidean distance) '''
    return math.sqrt((x1-x2)**2 + (y1-y2)**2)
```

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

### Code Listing 5.7

#### calculate\_area

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

```
def calculate_area(x1,y1,x2,y2,x3,y3):
    ''' return area using Heron's formula '''
    a = side_length(x1,y1,x2,y2)
    b = side_length(x2,y2,x3,y3)
    c = side_length(x3,y3,x1,y1)
    s = (1/2)*(a + b + c)
    return math.sqrt(s*(s-a)*(s-b)*(s-c))
```

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

### Code Listing 5.8

#### Full Triangle Program

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

```
import math

def get_vertex():
    x = float(input(" Please enter x: "))
    y = float(input(" Please enter y: "))
    return x,y

def get_triangle():
    print("First vertex")
    x1,y1 = get_vertex()
    print("Second vertex")
    x2,y2 = get_vertex()
    print("Third vertex")
    x3,y3 = get_vertex()
    return x1, y1, x2, y2, x3, y3

def side_length(x1,y1,x2,y2):
    '''return length of a side (Euclidean distance)'''
    return math.sqrt((x1-x2)**2 + (y1-y2)**2)

def calculate_area(x1,y1,x2,y2,x3,y3):
    '''return area using Heron's formula'''
    a = side_length(x1,y1,x2,y2)
    b = side_length(x2,y2,x3,y3)
    c = side_length(x3,y3,x1,y1)
    s = (1/2)*(a + b + c)
    return math.sqrt(s*(s-a)*(s-b)*(s-c))

x1, y1, x2, y2, x3, y3 = get_triangle()
area = calculate_area(x1,y1,x2,y2,x3,y3)
print("Area is",area)
```

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Did Functions Help?

- made our problem solving easier (solved smaller problems as functions)
- main program very readable (details hidden in the functions)



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## How to Write a Function

- **does one thing** if it does too many things, it should be broken down into multiple functions (refactored)
- **readable** how often should we say this? if you write it, it should be readable
- **reusable** if it does one thing well, then when a similar situation (in another program) occurs, use it there as well



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## More on Functions

- **complete** a function should check for all the cases where it might be invoked; check for potential errors
- **not too long** kind of synonymous with do one thing; use it as a measure of doing too much



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Rule 8

A function should do one thing



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Procedures

- functions that have no return statements are often called *procedures*
- procedures are used to perform some duty (print output, store a file, etc.)
- remember, return is not required



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Multiple Returns in a Function

- a function can have multiple **return** statements
- remember, the first **return** statement executed ends the function
- multiple returns can be confusing to the reader and should be used judiciously



"The Practice of Computing Using Python, 3rd Edition",  
Punch & Enbody, Copyright © 2017 Pearson Education, Inc.

## Rules

1. Think before you program!
2. A program is a human-readable essay on problem solving that executes on a computer.
3. The best way to improve your programming and problem solving skills is to practice!
4. A foolish consistency is the hobgoblin of little minds
5. Test your code, often and thoroughly
6. If it was hard to write, it is probably hard to read. Add a comment.
7. All input is evil, unless proven otherwise.
8. A function should do one thing.

"The Practice of Computing Using Python, 3rd Edition",  
Punch & Entbody, Copyright © 2017 Pearson Education, Inc.