Introduction to Computing

Functions

H. Turgut Uyar & Damien Jade Duff

2014-2018

Contents

- Writing Functions
- Keyword parameters
- More examples

Functions

functions: existing programs that take input (parameters) and produce output (return values)

```
from math import sqrt, radians

sqrt(25)
radians(90)
```

we can send output of one function to input of another:

```
from math import sin, radians
sin(1.57)
sin(radians(90))
```

Write your own functions

Define the function:

```
def my_radians(degrees):
    return degrees*pi/180
```

Use the function:

```
my_radians(90)
sin(my_radians(90))
```

General function definition template

```
def FUNCTION_NAME(PARAMETER1, PARAMETER2, ...):
    STATEMENT1
    STATEMENT2
    ...
    return EXPRESSION
```

Another example

```python
def c2f(c):
    f = 9 / 5 * c
    return f


raw_temp = input('Temperature (C): ')
celcius = float(raw_temp)
fahrenheit = c2f(celcius)
print(fahrenheit)
```

Variable Scope

- variables defined in the function
  are only accessible within the function
- multiple functions can define variables
  which have the same name
- input parameters are function-scoped

Celcius to Fahrenheit Conversion

```
def c2f(celcius):
    fahrenheit = 9 / 5 * celcius
    return fahrenheit


def main():
    raw_temp = input('Temperature (C): ')
    celcius = float(raw_temp)
    fahrenheit = c2f(celcius)
    print(fahrenheit)


main()
```

Keyword parameters

- most languages pass parameters by “place”:
- in Python you can pass name-value (“keyword”) parameters:

```
a = int("12")
b = int("12",base=16)
```

Keyword parameters in your own functions

```
def FUNCTION_NAME(PARAM1, KEYWORD_PARAM1 = DEFAULT_VALUE, ...):
    STATEMENT1
    STATEMENT2
    ...
    return EXPRESSION
```

Keyword parameters in your own functions

```
def to_celcius(farenheit=None,kelvin=None):
    if farenheight is not None:
        celcius = 5 / 9 * farenheit
    else:
        celcius = fahrenheit - 273.15
    return celcius
```

With error checking

```
def to_celcius(farenheit=None,kelvin=None):

    assert farenheit is not None or kelvin is not None

    if farenheight is not None:
        assert kelvin is None
        celcius = 5 / 9 * farenheit
    elif kelvin is not None:
        assert farenheit is None
        celcius = kelvin - 273.15

    return celcius
```

Example: String Filtering

```python
def filter(s,target):
    filtered = ''
    for c in s:
        if c != target:
            trimmed += c
    return trimmed
```

Example: String Splitting

```
def split(s, delimiter=' '):
    splitted = []
    start = 0
    end = 0
    while end < len(s):
        if s[end] == delimiter:
            word = s[start:end]
            splitted += [word]
            start = end + 1
        end += 1
    word = s[start:]
    splitted += [word]
    return splitted
```

Global Variables

a variable that can be read/write from multiple executions (calls) of functions is a global variable.

```
b = 5

def increment_b():
    global b # global variable
    print("b=" + str(b))
    b = b + 1

increment_b()
increment_b()
increment_b()
```

Global Variables

Fix this code:

```
CORRECT = "42"
all_guesses = []

def do_guess(n):
    if n == CORRECT:
          print('You win!')
          return True
    else:
          all_guesses = all_guesses + [n]
          print('Your guesses so far:',all_guesses)

while True:
    n = input('Your guess?')
    result = do_guess(n)
    if result:
        break
```