Functions

- function: unit of code that performs a task

- functions take parameters
- and return a result

Function Examples

- math library: trigonometry, radians

```
In [1]: import math

In [2]: math.sin(math.pi / 2)
Out[2]: 1.0

In [3]: math.radians(270)
Out[3]: 4.712388980384689

In [4]: math.sin(math.radians(270))
Out[4]: -1.0
```

Defining / Calling

- defining a function: describing how to perform the task
- using parameter names (formal parameters)

- calling a function: carrying out the computation
- with parameter values (actual parameters)

- libraries contains function definitions

Function Syntax

- defining:

```
def FUNCTION_NAME(FORMAL_PARAMETERS):
    STATEMENT1
    STATEMENT2
    ...
    return EXPRESSION
```

- calling:

```
FUNCTION_NAME(ACTUAL_PARAMETERS)
```

Celcius to Fahrenheit Conversion

```
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()
```

All Same

```
def all_same(nums):
    all_same_so_far = True
    value = nums[0]
    i = 0  # i = 1?
    while i < len(nums):
        if nums[i] != value:
            all_same_so_far = False
            break
        i += 1
    return all_same_so_far
```

Function Examples

- game of Yahtzee
- roll 5 dice
- place in one of the defined categories, get points
- all of a kind, n of a kind, full house, …

Five of a Kind

- count how many of each number
- is 5 one of those counts?

```
def five_of_a_kind(dice):
    counts = [0, 0, 0, 0, 0, 0]
    for num in dice:
        counts[num - 1] += 1
    return 5 in counts
```

Full House

- 3 of a kind and 2 of another kind

- count how many of each number
- are 3 and 2 in those counts?

```
def full_house(dice):
    counts = [0, 0, 0, 0, 0, 0]
    for num in dice:
        counts[num - 1] += 1
    return (3 in counts) and (2 in counts)
```

Count Dice

- instead of repeating code: function to count nums

```
def count_nums(nums):
    counts = [0, 0, 0, 0, 0, 0]
    for num in nums:
        counts[num - 1] += 1
    return counts
```

Rewrite Functions

```
def five_of_a_kind(dice):
    counts = count_nums(dice)
    return 5 in counts


def full_house(dice):
    counts = count_nums(dice)
    return (3 in counts) and (2 in counts)
```

Function Example

```
def greet(name):
    message = 'Hello, ' + name + '!'
    print(message)
```

- doing two things: prepare message, print
- interaction should be separated from computation

Function Example

```
def get_greeting(name):
    message = 'Hello, ' + name + '!'
    return message


name = input('What is your name? ')
message = get_greeting(name)
print(message)
```

Global Variables

- variables defined outside of all functions are global
- they can be accessed in all functions

- to change them, the function has to mark them as global