Functions

- You may be very familiar with the mathematical concept of a function

```
z = f(x,y)
```

f is a function that operates on the inputs x and y. The output of the function is z.

However, programming functions are much more generalized and versatile than this mathematical definition.

- How to define a function?

设置

Preview Exercise:

Which keyword is used to define function?

A Fun

B Define

C Def

D Function

提交

Functions

- Function is **a block of code** which only **runs** when it is **called**
- Reusable pieces/chunks of code
- Function characteristics:
  - has a **name**
  - has **a colon (:)** to mark the end of the function header
  - has **parameters(arguments)** (0 or more)
  - has a **docstring** (optional)：comment document
  - has a **body**
  - **Possibly returns** something

Functions

- A function is created with the **def** keyword. The statements in the block of the function must be **indented**.

```
def function_name(args):
    statements
    return values
```

- The **def** keyword is followed by the function name with round brackets enclosing the arguments and a **colon**.
- The **return** keyword is used to specify a list of values to be returned.

How to Write a Function

Name Arguments Colon

```
def is_even(i):
    """
    Input: i, a positive int
    Returns True if i is even, otherwise False"""
    print("inside is_even")
    return i%2 == 0

a=is_even(4)  # call the function

print(is_even.__doc__)  # print out comments document
```

Body

Comments document

If No Return Statement

```
def is_even( i ):
    """
    Input: i, a positive
    int   Does not return
    anything   """
    i%2 == 0
a=is_even(4)
```

Body

- Python returns the value **None, if no return given**

Functions: Name

- The rules for naming a function are a lot like rules for naming a variable:
  - They must start with **a letter or an underscore**:
  - They can have **numbers**.
  - They can be **any length** , but keep them short.
  - They should **uniquely** identify the function
  - They **can't be the same as a Python keyword**.
  - They can have the same name as an existing function (including a built-in), but avoid this for now.

```
def this_is_my_function():
```

```
def _f3(x):
```