

DS2000

2/14 - Tues.

Admin

- HW4 due Fri 9pm
- Problem 2 sticks only if useful
(quiz: clear ~~the~~ ticks on x axis)
- Quiz 5 today 4:30pm - Fri 9:50am

Agenda

1. Calling functions
2. Writing our own functions
3. Python

1. Calling Functions

- Functions organize our code, make it reusable
- Let us use other people's code
↳ start w/ calling functions

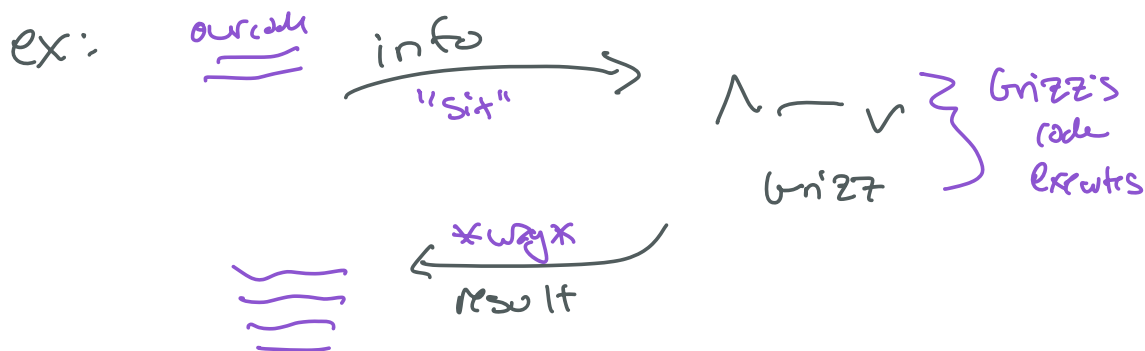
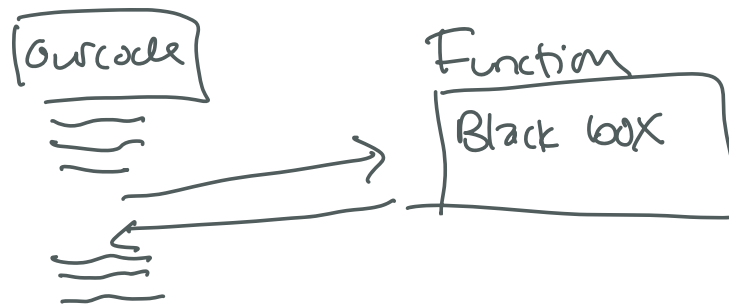
A function is a mini-program

- Contains code that gets executed more than once

- self-contained. own variables, can't see other code
- Does one job

call a function written into Python

↳ tell Python to execute that mini program



calling function examples

- | | | |
|-----------|-----------|-----------|
| • len() | • int() | • sum() |
| • print() | • input() | • float() |

To call a function, we need:

1. name - like a variable name (verb)

2. parameters → info. that it needs
function has zero or more
separated by commas
(data types)

3. return type → result that it gives
to its caller (data type)
returns at most one
thing

• parameters are how functions
get information

• return is how functions
give information } return is for computers ☒
print is for humans ☑

To call a function:

result = func(num)

↑ name

↓ param

↓
store what it returns

Ex

num = int("5")

name: int

param: string

return type: integer

xxx int("5") ∴ not saving xxx
result

Tex

```
name = input("Enter ")
```

```
name: input
param: string
return: string
```

2. Writing our own functions

- makes our code less messy, more reusable

Function Structure:

```
def func(m, n):  
    _____  
    _____  
    _____  
    _____  
    return result
```

ex:-

Ex: main - name

prams - none

returns - nothing

```
def main():
```

Code structure:

```
def func1():
```

```
def func2():
```

```
def main():
```

```
main()
```

} call functions
we've defined

Ex Func to add two numbers

```
def add(num1, num2):  
    ''' Block comment '''  
    result = num1 + num2  
    return result  
    print("hello")
```

define function

name - add

params - 2 numbers

return - one number

```
def main():
```

```
    added = add(3, 4)
```

```
    print(num1)
```

```
main()
```

call
function

add (3, 4)


num2 = 4
num1 = 3