

## *Last time:*

Function semantics

Function syntax

Function parameters and arguments

# *Today*

**Scope**

**Global variables**

**Keyword arguments**

**Default arguments**

## ***Recall: arguments and parameters***

**Arguments:** *values* passed into functions

```
f(1, 2+2, x*3)
```

**Parameters:** *variables* initialized by arguments

```
def f(x, y, z):  
    return x + y * z
```

# *Variable scope*

## Within a function:

```
def foo(x, y):  
    z = x % y  
  
    return z
```

These variables are *local* to the function  
(i.e., separate from other variables defined in the file)

## *More variable scope*

Example of separate variables:

```
name = "Jon"
def foo():
    name = "foo"
    print("name:", name)
foo()
```

Two `name` variables: **separate memory** and **separate values**

# *Global variables*

To access a global variable:

```
x = 12  
  
def foo():  
    global x  
    x = 13  
  
foo()  
print(x)
```

# *Keyword arguments*

Using the colour LCD screen:

```
rgb_lcd_colour(255, 0, 255)
```

(aside: what colour is this?)

**Easier to tell now:**

```
rgb_lcd_colour(red=255, green=0, blue=255)
```

No *positional* arguments after the first *keyword* argument

# *Default arguments*

Passed to the parameter if no argument in the call

```
def get_user_input(prompt='Input? '):  
    return input(prompt)
```

**One way to print:**

```
print('these', 'words', 'go', 'on', 'one', 'line')  
print('these', 'words', 'go', 'on', 'the', 'next', 'line')
```

**Another way:**

```
print(1, 2, 3, sep='*', end=' + ')  
print(4, 5, 6, sep='*')
```



# *Summary*

**Scope**

**Global variables**

**Keyword arguments**

**Default arguments**