## Functions

Lecture 5

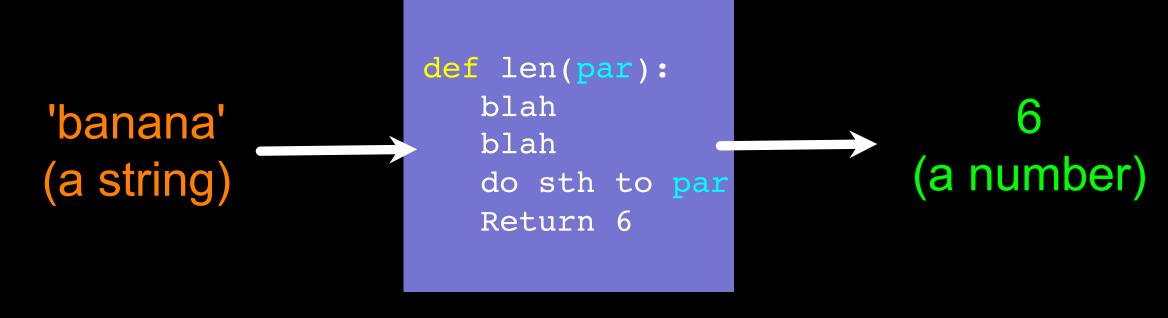
### len Function

```
>>> fruit = 'banana'
>>> x = len(fruit)
                                       Function:
>>> print(x)
                                       some stored code
6
                           len()
         'banana'
                        function
                                            (a number)
         (a string)
```

#### len Function

```
>>> fruit = 'banana'
>>> x = len(fruit)
>>> print(x)
6
```

A function is some stored code that we use. A function takes some input and produces an output.



#### Max Function

```
A function is some
>>> big = max('Hello world')
                                              stored code that we
>>> print(big)
                                              use. A function takes
W
                                                some input and
                                              produces an output.
                             max()
        'Hello world'
                                                   (a string)
                           function
         (a string)
```

```
x = len('python')

Assignment

6

Result
```

```
>>> x = len('python')
>>> print(x)
6
```

## Python Functions

- There are two kinds of functions in Python.
  - Built-in functions that are provided as part of Python print(), input(), type(), float(), int() ...
  - Functions that we define ourselves and then use

## Functions of Our Own...

## Building our Own Functions

```
Function Optional
keyword name parameters
def myfunction():
    print("This is my first function.")
    print('A test to see if it works.')
```

```
myfunction():
```

print("This is my first function.")
print('A test to see if it works.')

# Define the function but does not execute the body of the function

```
x = 5
print('Hello')

def myfunction():
    print("This is my first function.")
    print('A test to see if it works.')

print('Bye')
x = x + 2
print(x)
Output:
Hello
Bye
```

```
\mathbf{x} = \mathbf{5}
print('Hello')
def myfunction():
    print("This is my first function.")
    print('A test to see if it works.')
print('Bye')
myfunction()
                                    Hello
x = x + 2
                                    Bye
print(x)
                                  This is my first function.
                                    A test to see if it works.
```

#### Functions That Return Values

```
Function Optional
                                          returns values and exits
                   parameters
keyword name
   def test return():
        print("This is my first function.")
        print('A test to see if it works.')
        Return 6
                                    This is my first function.
   x = test return()
                                   A test to see if it works.
   print(x)
```

#### Functions That Return Values

```
def short word():
   wd = input('Type a word: ')
    if len(wd) > 4:
       return len(wd)
    else:
       print('The word is too short!')
res = short word()
                                         Type a word: python
print('The result: ',res)
```

## Arguments

- Where? in parentheses after the name
- What? a value we pass into the function as its input
- Why? do different work when we call it at different times

= len('Hello world')

Argument

#### Functions That Take Parameters

parameter: variable used in the function definition

```
def print_len(s):
    length = len(s)
    print('The word has', length , 'characters')
    return len(s)

res = print_len( "python" )
```

Argument

The word has 6 characters

#### Parameters

A parameter: a variable which we use in the function definition.

```
>>> def greet(lang):
        if lang == 'es':
           print('Hola')
        elif lang == 'fr':
           print('Bonjour')
    else:
           print('Hello')
>>> greet('en')
Hello
>>> greet('es')
Hola
>>> greet('fr')
Bonjour
>>>
```

# Arguments, Parameters, and Results

```
>>> length = len('python')
>>> print(length)

def len(par):
blah
blah
do sth to par
return 6

Result
```

## Multiple Parameters / Arguments

- We can define more than one parameter in the function definition
- We simply add more arguments when we call the function
- We match the number and order of arguments and parameters

```
def addtwo(a, b):
    added = a + b
    return added

x = addtwo(3, 5)
print(x)
```

#### To function or not to function...

- Organize your code into "paragraphs"
- Don't repeat yourself
- If something gets too long or complex, break it up into logical chunks and put those chunks in functions



#### Acknowledgements / Contributions



These slides are Copyright 2010- Charles R. Severance (www.dr-chuck.com) of the University of Michigan School of Information and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School of Information

... Insert new Contributors and Translators here

...