

## Lecture 02

Lect. PhD.  
Arthur Molnar

Procedural  
programming

What is a  
function

Variable scope

Passing  
parameters

# Procedural Programming

Lect. PhD. Arthur Molnar

Babes-Bolyai University

*arthur@cs.ubbcluj.ro*

# Overview

## Lecture 02

Lect. PhD.  
Arthur Molnar

### Procedural programming

What is a  
function  
Variable scope  
Passing  
parameters

## 1 Procedural programming

- What is a function
- Variable scope
- Passing parameters

# Procedural programming

## Lecture 02

Lect. PhD.  
Arthur Molnar

### Procedural programming

What is a  
function  
Variable scope  
Passing  
parameters

- A **programming paradigm** is a fundamental style of computer programming.
- **Imperative programming** is a programming paradigm that describes computation in terms of statements that change a program state.
- **Procedural programming** is imperative programming in which the program is built from one or more procedures (also known as subroutines or functions).

# What is a function

## Lecture 02

Lect. PhD.  
Arthur Molnar

Procedural  
programming

What is a  
function

Variable scope

Passing  
parameters

A **function** is a self contained block of statements that:

- Has a *name*,
- May have a list of (formal) *parameters*,
- May *return* a value
- Has a *documentation* (specification) which consists of:
  - A *short description*
  - *Type and description for the parameters*
  - conditions imposed over the input parameters (*precondition*)
  - Type and description for the return value
  - Conditions that must be true just after the execution (*post-condition*).
  - Exceptions

# What is a function

## Lecture 02

Lect. PhD.  
Arthur Molnar

Procedural  
programming

What is a  
function

Variable scope

Passing  
parameters

```
def max(a,b):  
    """  
    Return the maximum of two values  
    input: a,b - the parameters to compare  
    output: The largest of the parameters  
    Error: TypeError - if the parameters cannot be compared  
    """  
    if a > b:  
        return a  
    return b
```

# What is a function

## Lecture 02

Lect. PhD.  
Arthur Molnar

Procedural  
programming

What is a  
function

Variable scope

Passing  
parameters

- Does the following function work?
- Can you tell what it does?
- What if it were more complex?

```
def f(a):  
    sum = 0  
    for i in range(0, a):  
        sum += i  
    return sum
```

# What is a function

## Lecture 02

Lect. PhD.  
Arthur Molnar

Procedural  
programming

What is a  
function

Variable scope

Passing  
parameters

- We do not consider a function without specification to be correctly implemented!

```
def f(a):  
    sum = 0  
    for i in range(0, a):  
        sum += i  
    return sum
```

# What is a function

## Lecture 02

Lect. PhD.  
Arthur Molnar

Procedural  
programming

What is a  
function  
Variable scope  
Passing  
parameters

Every function written by you should:

- Use meaningful names (function name, variable names)
- Provide specification
- Include comments
- Have a test function (see later)

```
def isPrime(nr):  
    """  
    Verify if a number is prime  
    nr - integer number, nr > 1  
    return True if nr is prime, False otherwise  
    """  
    div = 2  
    while div < nr and nr % div > 0:  
        div += 1  
    # For non-primes, the number itself is the  
    # first divider  
    return div >= nr
```



# What is a function

## Lecture 02

Lect. PhD.  
Arthur Molnar

Procedural  
programming

What is a  
function

Variable scope

Passing  
parameters

- A **function definition** is an executable statement introduced using the keyword **def**.
- The function definition does not execute the function body; this gets executed only when the function is called. A function definition defines a user-defined function object.

```
def max(a,b):  
    """  
    Return the maximum of two values  
    input: a,b - the parameters to compare  
    output: The Largest of the parameters  
    Error: TypeError - if the parameters cannot be compared  
    """  
    if a > b:  
        return a  
    return b
```

# Variable scope

## Lecture 02

Lect. PhD.  
Arthur Molnar

### Procedural programming

What is a  
function

Variable scope

Passing  
parameters

A *scope* defines the visibility of a name within a block. If a local variable is defined in a block, its scope includes that block. All variables defined at a particular indentation level or scope are considered local to that indentation level or scope

- Local variable
- Global variable

# Variable scope

## Lecture 02

Lect. PhD.  
Arthur Molnar

Procedural  
programming

What is a  
function  
**Variable scope**  
Passing  
parameters

Rules to determine the scope of a particular name (variable, function name):

- A name defined inside a block is visible only inside that block
- Formal parameters belong to the scope of the function body (visible only inside the function)
- A name defined outside a function (at the module level) belongs to the module scope
- When a name is used in a code block, it is resolved using the nearest enclosing scope.

# Variable scope

## Lecture 02

Lect. PhD.  
Arthur Molnar

### Procedural programming

What is a  
function  
Variable scope

Passing  
parameters

At any time during execution, names are resolved using :

- The innermost scope, which is searched first, contains the local names (inside the block)
- The scopes of any enclosing functions, which are searched starting with the nearest enclosing scope
- The next-to-last scope contains the current module's global names
- The outermost scope (searched last) is the namespace containing built-in names

# Variable scope

## Lecture 02

Lect. PhD.  
Arthur Molnar

### Procedural programming

What is a  
function

Variable scope

Passing  
parameters

- Use the **globals()** and **locals()** functions to figure out the scope of each variable
- What other python built-in functions do you know?

# Demo

## Lecture 02

Lect. PhD.  
Arthur Molnar

Procedural  
programming

What is a  
function

**Variable scope**

Passing  
parameters

## Variable scope

04-VariableScope.py

# Calls

## Lecture 02

Lect. PhD.  
Arthur Molnar

### Procedural programming

What is a  
function  
Variable scope

Passing  
parameters

A **block** is a piece of Python program text that is executed as a unit. Blocks of code are denoted by line indentation. A **function body** is a block. A block is executed in an *execution frame*. When a function is invoked a new execution frame is created.

# Calls

## Lecture 02

Lect. PhD.  
Arthur Molnar

### Procedural programming

What is a  
function  
Variable scope

Passing  
parameters

An execution frame contains:

- Some administrative information (used for debugging)
- Determines where and how execution continues after the code block's execution has completed
- Defines two namespaces, the local and the global namespace, that affect execution of the code block.
- A *namespace* is a mapping from names (identifiers) to objects. A particular namespace may be referenced by more than one execution frame, and from other places as well.



# Calls

## Lecture 02

Lect. PhD.  
Arthur Molnar

Procedural  
programming

What is a  
function  
Variable scope

Passing  
parameters

- Adding a name to a namespace is called binding a name (to an object); changing the mapping of a name is called rebinding.
- Removing a name is unbinding.
- Namespaces are functionally equivalent to dictionaries (and often implemented as dictionaries).

## Discussion

What did the output of `locals()`, `globals()` look like?

# Parameter passing

## Lecture 02

Lect. PhD.  
Arthur Molnar

Procedural  
programming

What is a  
function  
Variable scope

Passing  
parameters

- **Formal parameter** - an identifier for an input parameter of a function. Each call to the function must supply a corresponding value (argument) for each mandatory parameter
- **Actual parameter** - a value provided by the caller of the function for a formal parameter.
- The actual parameters (arguments) to a function call are introduced in the local symbol table of the called function when it is called (arguments are passed *by object reference*)

# Parameter passing

## Lecture 02

Lect. PhD.  
Arthur Molnar

### Procedural programming

What is a  
function  
Variable scope

Passing  
parameters

- **Pass by value** - the argument is evaluated, and the result of the evaluation is bound to the formal parameter of the function
- **Pass by reference** - function receives a reference to the actual argument, rather than a copy to its value
- **Side effect** - a function that modifies the caller's environment (beside producing a value) is said to have side effects

# Parameter passing

## Lecture 02

Lect. PhD.  
Arthur Molnar

### Procedural programming

What is a  
function

Variable scope

Passing  
parameters

## Discussion

What are the advantages and disadvantages of pass by value and pass by reference?

# Parameter passing

## Lecture 02

Lect. PhD.  
Arthur Molnar

### Procedural programming

What is a  
function

Variable scope

Passing  
parameters

How about in Python?

- **Object references are passed by value**

# Demo

## Lecture 02

Lect. PhD.  
Arthur Molnar

### Procedural programming

What is a  
function

Variable scope

Passing  
parameters

## Parameter passing

05-ParameterPassing.py

# Passing parameters

## Lecture 02

Lect. PhD.  
Arthur Molnar

### Procedural programming

What is a  
function  
Variable scope

Passing  
parameters

What happened in the studied example?

- At first, Python behaves like call-by-reference
- When you change a variable's value, it "switches" to call-by-value

# Demo

## Lecture 02

Lect. PhD.  
Arthur Molnar

Procedural  
programming

What is a  
function

Variable scope

Passing  
parameters

## Side Effects

06-SideEffects.py