

Bahan Ajar

Chapter 9



Materi Pembelajaran

Matakuliah :

INTERFACING PERIPHERAL

Kode Matakuliah : SKO 20416

Prodi : SISTEM KOMPUTER

Dosen Pengampu Matakuliah:

Bayu Nugroho, S.Kom., M.Eng

Tables of Content

Python Functions

- Creating a Functions
- Calling a Function
- Arguments
- Number of Arguments
- Arbitrary Arguments, *args
- Keyword Arguments
- Default Parameter Value
- The pass Statement

Tugas Mandiri

Python Functions

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result

Creating a Function

In Python a function is defined using the def keyword:

Example

```
def my_function():  
    print("Hello from a function")
```

Calling a Function

To call a function, use the function name followed by parenthesis:

Example

```
def my_function():  
    print("Hello from a function")
```



Arguments

Information can be passed into functions as arguments. Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

The following example has a function with one argument (fname). When the function is called, we pass along a first name, which is used inside the function to print the full name:

Example

```
def my_function(fname):  
    print(fname + " Refsnes")  
  
my_function("Emil")  
my_function("Tobias")  
my_function("Linus")
```

Number of Arguments

By default, a function must be called with the correct number of arguments. Meaning that if your function expects 2 arguments, you have to call the function with 2 arguments, not more, and not less

Example

This function expects 2 arguments, and gets 2 arguments:

```
def my_function(fname, lname):  
    print(fname + " " + lname)  
  
my_function("Emil", "Refsnes")
```

Arbitrary Arguments, *args

If you do not know how many arguments that will be passed into your function, add a * before the parameter name in the function definition.

This way the function will receive a tuple of arguments, and can access the items accordingly:

Example

If the number of arguments is unknown, add a * before the parameter name:

```
def my_function(*kids):  
    print("The youngest child is " + kids[2])  
  
my_function("Emil", "Tobias", "Linus")
```

Keyword Arguments

You can also send arguments with the key = value syntax.

This way the order of the arguments does not matter.

Example

```
def my_function(child3, child2, child1):  
    print("The youngest child is " + child3)  
  
my_function(child1 = "Emil", child2 = "Tobias", child3 = "Linus")
```

Default Parameter Value

The following example shows how to use a default parameter value.

If we call the function without argument, it uses the default value:

Example

```
def my_function(country = "Norway"):  
    print("I am from " + country)  
  
my_function("Sweden")  
my_function("India")  
my_function()  
my_function("Brazil")
```

The pass Statement

function definitions cannot be empty, but if you for some reason have a function definition with no content, put in the pass statement to avoid getting an error.

Example

```
def myfunction():  
    pass
```



Tugas Mandiri (teori):

1. If both the function and the main program can read and process global variables, why do I need to pass parameters to a function? Can't I just use global variables? Explained.
2. You must define a function before you can use it in your Python script. True or false? Explained

Tugas Mandiri (prakt):

Tuliskan script program python **Functions** untuk menghidupkan 6 buah LED secara di Raspberry Pi.

end

