Difference Between Function Prototype and Function Definition in C

www.differencebetween.com

Key Difference - Function Prototype vs Function Definition in C

A function is a group of statements used to perform a specific task. In C programming, the execution starts from main (). It is a function. Rather than writing all statements in the same program, it can be divided into multiple functions. Each function will perform different functionalities. The function prototype tells the compiler about the function name, return types and parameters. It is also known as a function declaration. Each function has a particular name to identify it. The function statements are written inside a pair of curly braces. The functions can return a value. There are some functions that do not return a value. The data is passed to the function using the parameter list. The function definition has the actual functionality performed by the function. In C programming, there is function prototyping and function definition. The **key difference** between the function prototype and function definition is that the **function prototype only contains the declaration of the function while the function definition contains the actual implementation of the function.** The function definition has the local variables and the statements that determine what the function does.

What is Function Prototype in C?

Function Prototype provides the function declaration. It specifies the name of the function, the return types, the parameters. The return types are the data type that returns from the function. When a function is returning an integer, then the return type is int. When a function is returning a float value, then the return type is a float. If the function is not returning any value, it is a void function. The function name is used to identify it. C keywords cannot be used as function names. The data is passed to the function using parameters. The function prototype does not contain the real implementation of the function. The function prototype has the following syntax.

<return type> <function name> (parameter list);

If there is a function to calculate the maximum of two numbers the declaration can be written as int max (int num1, int num2); The maximum value should be found in num1 and num2. Those are integers, and they are passed to the function. The return type, in the beginning, is also int. So, the function returns an integer value. It t is not necessary to write the parameter names in the function prototype. But it is necessary to write the data types. Therefore, int max (int, int); is also a valid function prototype. If there are two integers as num1, num2, num3 and the prototype is written as int max(int num1, int num2, num3); it is invalid. The num1, num2 have the data types, but num3 does not have a data type. Therefore, it is invalid.

Figure 01: C Programming

Refer the below program.

```
#include <stdio.h>

int CarMax(int x, int y);

int main(){

int p =10;
int q= 20;
int answer;
answer = calMax(p,q);
printf(“The maximum value is %d\n”, answer);
return 0;
}
int calMax(int p, int q){
int value;
if(p>q) {
value = p;
}
else {
value = q;
}
return value;
}
```

According to the above, the second statement shows the function prototype. It does not have the implementation. The actual implementation is after the main program. The function prototypes are more useful when defining a function in one source file and call that function in another in another file.

What is Function Definition in C?

The function definition has the actual implementation of the function. It contains what the function should do. When the program calls the function, the control is transferred to the called function. After the execution of the function, the control returns back to the main function. The required data is passed to the function as a parameter list. If there is a value returning, then the return type is mentioned. If there are no returning values, the return type is void. Refer the below function to calculate the area of a triangle.

```
#include <stdio.h>
float calArea(int x, int y);
int main () {
int p =10;
int q= 20;
flaot area;
area = calArea(p,q);
printf (“The maximum value is %f\n”, area);
return 0;
}
float calArea (int x, int y) {
float value;
value = 0.5 * x * y;
return value;
}
```

According to the above program, the second statement indicates the function prototype. The actual implementation of what the function performs is written after the main program. It is the function definition. The p and q values are passed to the calArea function. The variable value is a local variable to the calArea function. The area is calculated and assigned to the variable value. Then it is returned back to the main program.

What are the Similarities Between Function Prototype and Function Definition in C?

- Both are related to functions.
- Both contain function name.
- Both contain return types.
- Both contain parameters.

What is the Difference Between Function Prototype and Function Definition in C?

Function Prototype vs Function Definition in C	
The function prototype specifies the function name, return type, parameters but omits the function body.	The function definition specifies the function name, return type; parameters include a function body.
Implementation	
The function prototype does not have the function implementation.	The function definition has the function implementation.

Summary - Function Prototype vs Function Definition in C

Using functions in programs has advantages. Functions increase code reusability. It is not necessary to write the same code again and again. Instead, the programmer can divide the program and call the necessary function. In C there are library functions. These functions are declared in the C header files. Some of them are printf (), scanf () etc. The programmer can also write their own functions. There are two terms which are associated with functions in C. They function prototype and function definition. The difference between the function prototype and function definition in C is that the function prototype only contains the declaration of the function while the function definition contains the actual implementation of the function.

Reference:

1.C Function Definitions. Available here
2.tutorialspoint.com. “C Functions.” The Point. Available here

Image Courtesy:

1.'The C Programming Language logo'By Rezonansowy (Public Domain) via Commons Wikimedia

How to Cite this Article?

APA: Difference Between Function Prototype and Function Definition in C.(2018 February 08). Retrieved (date), from http://differencebetween.com/difference-between-function-prototype-and-vs-function-definition-in-c/

MLA: "Difference Between Function Prototype and Function Definition in C" Difference Between.Com. 08 February 2018. Web.

Chicago: “Difference Between Function Prototype and Function Definition in C.” Difference Between.Com. http://differencebetween.com/difference-between-function-prototype-and-vs-function-definition-in-c/ accessed (accessed [date]).