

Difference

Return

(i) Used to terminate the execution of a function and return a value to the calling code.

(ii) Typically found within functions to output a value back to the caller.

Example :-

```
function add($a,$b)
{
    return $a+$b;
}
```

```
$result = add(3,4);
echo $result;
```

Output → 7.

(iv) When return is encountered, the function stops executing & returns the

Print

Used to print output data to the screen.

Can be used outside of functions to directly output content to the browser or command line.

Example :-

```
$name = "manta";
print "Hello, $name;"
```

Output →

Hello, manta.

Prints directly outputs its argument (which

specified value (if any) to the caller. It can only be used inside function.

can be a string or a variable). to the screen. It always returns, so it can be used in expressions, although this is less common.

v) return is used to pass a value back to the caller and exit the current function, while point is used purely

while point is used purely for displaying output.

vi) return is used inside functions.

point is used outside function to directly output content.

vii) return can return any data type (int, float, strings, arrays, objects).

while point is primarily used for strings and variables.

viii) return is used to return values from function.

Point is used to output content directly to the screen in PHP.

①. first class function :-

- functions can be assigned to variables.
- functions can be passed as arguments to other functions.
- functions can be returned as values from other functions.
- functions can be stored in data structure (like lists or dictionaries).

e.g :-

```
def greet(name):
```

```
    return f"Hello, {name}!"
```

assigning function to a variable

```
greet_func = greet
```

passing function as an argument

```
def call_func(func, arg):
```

```
    return func(arg).
```

```
result = call_func(greet, "Alice") # passing  
# greet function as an argument
```

```
print(result)
```

→ Hello, Alice!

(ii) Higher-order functions ->

- A higher order function is a function that takes one or more functions as argument or return a function as its result.
- It essentially operates with functions at a higher level, either by taking as arguments or by returning them.

e.g:- `def multiply(x):
 return x * 2`

`def add(x):
 return x + 2.`

`def apply_operation(func, n):
 return func(n)`

`result1 = apply_operation(multiply, 3)`
passing 'multiply' fun as argmt

`result2 = apply_operation(add, 3)` # Passing
add function as an argument

`print(result1) # 6.
print(result2) # 5`

→ 6.
5.

- first class functions :- functions that can be treated as values, they can be assigned to variables passed as arguments returned from other functions and stored in data structures.
- High order functions :- functions that either take other functions as arguments or return functions as their result.

e.g.: def square(x):
return ($x * * 2$)

def addvalues(x, y):
print(x, y (30))

addvalues(10, square.)

$\Rightarrow 10 \ 900$.