Function in Hindi - Python in Hindi

Python - Function

Python Function ये statements का एक समूह होता है | हर एक Program में एक function तो होता ही है | जहाँ पर Function की statements की जरुरत होती है वहाँ पर Function को call किया जाता है |

Python Functions Advantages

- Function में लिखा हुआ code बार-बार लिखना नहीं पड़ता |
- बड़े Program को छोटे-छोटे function में विभाजित किया जा सकता है |
- Function Programmer का समय, Program की space और memory बचाता है |
- अगर Program में कहा पर error आ जाए तो उसे आसानी से निकाला जा सकता है |
- जहाँ पर जरुरत हो वहाँ पर function को बार-बार call किया जा सकता है |

Types of Function in Python

1. In-Built/Predefined Function
2. User-Defined Function

1. In-Built/Predefined Function

Python में अभीतक print() नामक function बहुत ही बार इस्तेमाल किया है | print() function ये Python में in-built function है |

2. User-Defined Function

Python में user-defined function में दो प्रकार पड़ते है |

- Function Definition
- Function Call

2.1 Function Definition

Syntax for Function Definition

```
def function_name(parameter(s)):
 "function_docstring"
 function_body
 return statement
```

- **def :** Python में function को create करना हो तो शुरुआत में 'def' keyword का इस्तेमाल किया जाता है |
- **function_name :** def keyword के बाद function का नाम दिया जाता है | हर function का नाम उसके statement से related हो तो अच्छा रहता है |
- **(parameter(s)) :** Optional. parameters optional होते है parenthesis(()) नहीं होते है | function के name के बाद parenthesis(()) दिया जाता है | उन parenthesis में एक या एक से ज्यादा parameters दिए जाते है | parameters; optional होते है |
- **:** : parenthesis के बाद colon(:) देना अनिवार्य होता है | colon देने के बाद Python interpreter द्वारा automatic code indent होता है |
- **"function_docstring" :** Optional. यहाँ पर documentation string दिया जाता है |
- **function_body :** Optional. ये function की body होती है | जिसमे कुछ local variables और statements हो सकते है | body में दिए हुए variables को global भी बनाया जा सकता है |
- **return statement :** Optional. ये return statement होता है | return statement के लिए 'return' keyword का इस्तेमाल किया जाता है | ये statement से function end होता है | अगर return statement दिया नहीं जाता है तो default 'None' return होता है |

Example for Function Definition

Source Code :

```
#Function definition
def func(param):         #function name, parameter and colon
   "This is a docstring" #docstring
   print(func.__doc__)   #function body
   return param          #return statement
```

Example पर 'func' ये function का नाम है |

उसके बाद parenthesis(()) में 'param' नाम का एक parameter और parenthesis के बाहर colon(:) दिया गया है |

function के indent code में पहले docstring दी गयी है |

उसके बाद function की body में class attribute की मदद से docstring को print किया गया है

और आखिर में return statement में 'param' इस parameter को print किया गया है |

2.1 Function Call

Function के call में सिर्फ function का नाम और अगर function को parameter हो तो parameter की value दी जाती है | जबतक function को call नहीं किया जाता तबतक function का code execute नहीं होता है |

Example for Function Call

Source Code :

```
#Function definition
def func(param):          #function name, parameter and colon
    "This is a docstring" #docstring
    print(func.__doc__)   #function body
    return param          #return statement

print("Call 1")
print(func(5))           #Function call

print("Call 2")
print(func(10))          #Function call
```

Output :

```
Call 1
This is a docstring
5
Call 2
This is a docstring
10
```

Python में जब function को call करके जिस data type की value as a argument दी जाती है तब उसका data type decide हो जाता है |

Source Code : Output :

```
Function Call1 :
Hello
2
Function Call2 :
2
Hello
```

Function Argument in Python

Python में चार प्रकार के function parameter होते है |

1. Default Argument
2. Required Argument

3. Keyword Argument
4. Variable Number of Argument

1. Default Argument

Default argument में definition पर argument के लिए default value '='(assignment operator) से set की जाती है |

Function Call पर जब definition पर assign किया हुआ argument नहीं दिया जाता तो default value pass की जाती है |

Source Code : Output :

```
Call1
5
Hello
Call2
8
Hello
Call3
8
Hii
```

---

2. Required Argument

Required Argument में function call पर argument देना अनिवार्य होता है |

जब function definition पर argument दिया जाता है तब उसे function call पर उसकी value देना required होता है |

Source Code : Output :

```
Call1
5
Hello World
Call2
Hello World
5
```

अगर function call पर argument दिया नहीं जाता है तो , TypeError का exception आ जाता है |

Source Code : Output :

```
    ReqArg()
TypeError: ReqArg() missing 2 required positional arguments: 'num' and 'str'
```

अगर function पर दो arguments है | अगर call करते वक्त एक ही argument value दी जाती है तो तब भी 'TypeError' exception आ जाता है |

Source Code : Output :

```
    ReqArg(5)
TypeError: ReqArg() missing 1 required positional argument: 'str'
```

---

3. Keyword Argument

अबतक function पर positional argument का इस्तेमाल किया गया है | Positional argument ये normal parameters होते है |

positional argument में जब values दी जाती है तब उसकी position की हिसाब से call पर value assign की जाती है |

Keyword Argument में function call पर दिया जाता है | Keyword Argument की मदद से arguments की positions को बदला जाता है |

Source Code : Output :

```
Call1
int : 10
float : 1.5
String : H
Call2
int : 20
float : 2.8
String : G
Call3
int : 5
float : 8.6
String : R
```

---

4. Variable Number of Arguments

किसी वक्त पर पता नहीं होता कि function पर कितने arguments pass करने है | ऐसे वक्त पर 'Number of Arguments' का महत्व काफी बढ़ जाता है |

जब number of argument का इस्तेमाल करना हो तो Function के definition पर argument से पहले '*'(asterisk) का इस्तेमाल किया जाता है |

'Function definition पर दिया हुआ argument 'tuple' जैसा होता है |

Source Code : Output :

```
(5, 10, 'Hello')
5
10
Hello
5
10
Hello
```

Anonymous or Lambda Function

जब Normal Function create किया जाता है तब function को विशिष्ट नाम दिया जाता है | लेकिन Anonymous या lambda Function का कोई नाम नहीं होता है |

Normal Function के लिए 'def' keyword का इस्तेमाल किया जाता है लेकिन Anonymous/Lambda Function के लिए 'lambda' keyword का इस्तेमाल किया जाता है | ये function काफी छोटा होता है |

Syntax for Anonymous/Lambda Function

```
lambda arg1,arg2,..,argN : expression
```

arg1,arg2,..,argN : lambda function में एक या एक से ज्यादा arguments हो सकते है |

expression : lambda function में एक ही expression होता है | expression; return भी होता है |

Example पर 'anony' नाम के function object पर anonymous function को assign किया गया है | उसके बाद function object का इस्तेमाल function के रूप में call करके और argument पर values pass की गयी है |

Source Code : Output :

```
False
True
```

Another Example for Anonymous/Lambda Function

Source Code : Output :

```
False
True
```

Another Example for Anonymous/Lambda Function

Source Code :

```
anony = lambda a, b : print(a < b) Aanonymous Function

#same as

def anony(a, b):     #Named/Normal Function
    return a < b
```