Python in Hindi – Introduction

March 29, 2019August 6, 2018 by Javahindi.com

- Introduction to **python in Hindi**
- Features of *python in Hindi*
- Usage of python in Hindi

Introduction to Python

Python एक powerful dynamic programming language है। यह Guido Van Rossum द्वारा 1991 में Netherlands के National Research Institute for Mathematics and Computer Science में create की गयी थी।

It is Beginner Friendly

Python एक beginner friendly language है। इसे सबसे अधिक इसके simple syntax और readable code के लिए जाना जाता है। Python को अत्यधिक readable design किया गया है।

Python का syntax बहुत ही आसान है और English language की तरह है। जँहा दूसरी programming languages विराम चिन्ह जैसे की semicolon (;) आदि का प्रयोग करती है वँहा python simple English keywords का प्रयोग करती है। English के अलावा python mathematics से भी प्रभावित देखी जाती है।

Both Functional & Object Oriented

Python functional भी है और object oriented भी है। इसे आप scripting language की तरह use कर सकते है या फिर programming language की तरह use कर सकते है।

Python में आप चाहे तो Java और C# की तरह classes आदि object oriented features का प्रयोग करते हुए program लिख सकते है या फिर simply C और C++ languages की तरह functions द्वारा भी program create कर सकते है।

Python को आसानी से C, C++ और Java आदि के साथ integrate किया जा सकता है। इसके अलावा python interpreter को C और C++ के functions और data types से extend भी किया जा सकता है।

Open Source

Python एक open source programming language है। इसका code GNU General Public License के under available है।

Platform Independent

Python एक platform independent language है। यह Windows, Mac, Linux आदि अलग अलग platform पर कार्य करती है।

Features of Python

Python एक unique language है। इसके features इसे दूसरी languages से अलग बनाते है। Python के कुछ popular features के बारे में निचे बताया जा रहा है।

Interpreted

Python एक interpreted language है। Program को run करने से पूर्व आपको अपने code को compile करने की आवश्यकता नहीं होती है। Python code run time पर ही interpreter द्वारा process किया जाता है।

Python के इस feature की वजह से python rapid prototyping के लिए एक बेहतरीन language साबित हुई है। क्योंकि आपको code को बार बार compile करने की आवश्यकता नहीं होती है।

इससे बड़ी बड़ी applications के rapidly prototype generate करके उन्हें test किया जा सकता है।

हालाँकि intepreted होने की वजह से python C और C++ जैसी compiled languages से थोड़ी slow है। क्योंकि उन languages में एक बार code को compile करने के बाद directly run कर दिया जाता है और कोई processing नहीं होती है और python code पूरी तरह run होने से पूर्व interpreter द्वारा process किया जाता है।

लेकिन फिर भी python का यह feature अत्यधिक महत्वपूर्ण है क्योंकि compiled languages द्वारा आप rapid prototyping जैसा important कार्य नहीं किया जा सकता है।

Less Code Writing

Python में कोई भी task perform करने के लिए किसी भी दूसरी programming language से कम statements लिखने की आवश्यकता होती है।

Python का syntax कुछ इस प्रकार बनाया गया है ताकि कम से कम code में task perform किया जा सके। Java में कोई task जितने code में perform किया जाता है वही task python के आधे से भी कम code में perform किया जा सकता है।

Dynamically Typed

Python में आपको variables के data type पहले से define करने की आवश्यकता नहीं होती है। Python स्वयं ही variables में store की गयी values के आधार पर उनके data types का अनुमान लगा लेती है।

इससे developers का workload कम हो जाता है। इसके अलावा developer data types से related संभावित गलतियों से भी बच जाता है।

Strongly Typed

Python एक strongly typed language है। इसमें दूसरी programming languages की तरह आप एक data type की value को दूसरे data type की value में convert नहीं कर सकते है।

यदि आपने C या C++ languages का प्रयोग किया होगा तो पाया होगी के वे languages एक data type के data को दूसरे data type की value में automatically convert करती है। लेकिन Python में ऐसा संभव नहीं है।

Usage of Python

जैसा की मैने पहले बताया python को एक scripting language की तरह भी use किया जा सकता है और एक programming language की तरह भी use किया जा सकता है। इससे इस language का scope दूसरी languages से अधिक है।

आगे python के कुछ popular और important usage बताये जा रहे है।

- Python को server side web development के लिए use किया जाता है।
- Python को software development (Gaming, GUI etc.) के लिए use किया जाता है।
- Python को Big data को handle (analyse) करने के लिए use किया जाता है।
- Python को rapid prototyping के लिए use किया जाता है।
- Python को complex mathematical problems को solve करने के लिए use किया जाता है।

कई बड़ी organizations जैसे की Google, Facebook और Quora आदि python को use करती है।

Python in Hindi – Installation

March 29, 2019August 10, 2018 by Javahindi.com

Python एक interpreted language है। Python में programming करने के लिए आपको python interpreter की आवश्यकता होती है। इसलिए python के बारे में और अधिक आगे बताने से पूर्व इस tutorial में आपको python interpreter को install करना बताया जा रहा है।

Python interpreter को install करना बहुत ही आसान है। क्योंकि India में ज्यादातर सभी Windows operating system को use करते है, इसलिए इस tutorial में Windows में ही python को install करना बताया जा रहा है।

Python को install करने के लिए आपको सबसे पूर्व इसे download करना होता है। Python को आप इसकी website https://www.python.org के download page https://www.python.org/downloads/ से download कर सकते है।

Python एक cross platform language है। इसलिए अलग अलग operating systems जैसे की (Windows, Mac OS और linux आदि) के लिए अलग अलग interpreters provide किये गए है।

Python को install करने के लिए आपको अपने OS के लिए python interpreter को ढूँढने की आवश्यकता नहीं है। Python website के download page को आप जिस operating system से visit करते है वह उसी operating system के लिए interpreter download का option automatically show करता है।

जैसा की निचे दी गयी image में show हो रहा है।

ऊपर दी गयी image में python website के download page पर Windows के लिए python interpreter download का option show हो रहा है। ऐसा इसलिए हुआ क्योंकि python website को Windows OS से visit किया गया था।

Python interpreter को download करने के लिए आप इस page पर available python के latest version को download करने के option पर click करते है। ऐसा करने पर python installer की download शुरू हो जाती है।

Python installer की size सभी operating systems के लिए लगभग 25 MB है। Installer के download होने के बाद आप उसे open करते है। जैसे ही आप उसे open करेंगे तो आपको administration permission की window show होती है।

इस window पर आप run button पर click करते है। इसके बाद आपको python installation wizard की window show होती है। जैसा की निचे दी गयी image में show हो रहा है।

इस window में आप python install करने के option select कर सकते है। आप चाहे तो customize installation button पर click करके installation को customize (change installation directory etc.) कर सकते है।

इसके बाद आपको Add python 3.7 to PATH option को अवश्य check करना चाहिए। इससे python आपके system में एक service के रूप में install हो जाती है और आपको programming करने में सहूलियत रहती है।

इसके बाद आप Install Now button पर click करते है। इसके बाद आपके system में python installation की process शरू हो जाती है।

Installation को पूर्ण होने में 2 से 3 minute तक का समय लग सकता है। Installation पूर्ण होने पर आपको successful installation की window show होती है।

Python आपके system में install हुई या नहीं यह आप Star menu में जा कर confirm कर सकते है।

अब आप python में programming करना शुरू कर सकते है।

Python in Hindi – Interpreter

March 29, 2019August 13, 2018 by Javahindi.com

- Introduction to **python interpreter in Hindi**
- Python interpreter modes in Hindi

Introduction to Python Interpreter

Python को अपने system में install करने के बाद अब आप python में programming करना शुरू कर सकते है। इसके लिए आप python interpreter का प्रयोग करते है। Python interpreter को use करने के आपके pass दो options available है।

- Python Command Line Interpreter
- Python IDLE (Integrated Development and Learning Environment)

इन दोनों में same ही python interpreter काम करता है और दोनों same ही libraries का प्रयोग करते है। ये दोनों ही options python द्वारा provide किये जाते है। जब आप python को install करते है तो ये दोनों आपके system में install होते है।

जैसा की निचे दी गयी image में आप देख सकते है।

आइये अब इन दोनों options के बारे में detail से जानने का प्रयास करते है।

Python Command Line Interpreter

Python command line interpreter एक simple command line window होती है। Command line interpreter में कोई भी कार्य commands द्वारा किया जाता है। Python command line interpreter थोड़ा कम user friendly है।

Python command line interpreter की image निचे show की गयी है।

Python IDLE (Integrated Development and Learning Environment)

Python IDLE एक GUI interface है। इसकी image निचे दी जा रही है।

विभिन्न tasks perform करने के लिए IDLE में menus और options provide किये गए है।

यह किसी सामान्य editor की तरह होता है और इसके साथ कार्य करना बहुत आसान होता है। यह python command line interpreter से अधिक user friendly होता है।

हालाँकि python code को आप किसी दूसरे editor में लिखकर और उसे .py extension के साथ save करके भी run करवा सकते है। लेकिन IDLE में आपको लिखने से लेकर program को run करने तक के सभी options provide किये गए है।

Python Interpreter Modes

Python interpreter दो modes में कार्य करता है। ये दोनों ही mode अलग अलग purposes के लिए प्रयोग किये जाते है। Python interpreter के सही utilization के लिए आपको इनके बारे में जानकारी होना आवश्यक है।

Python interpreter के दोनों modes के बारे में आगे detail से बताया जा रहा है।

Interactive Mode

Python interpreter का interactive mode command prompt की तरह कार्य करता है। यह आपके command (statement) enter करने का wait करता है। जब आप command type करते है तो python interpreter उस command को execute करके output show कर देता है। इसके बाद वह फिर से दूसरी command के लिए wait करता है।

Interactive mode में python interpreter को python shell कहा जाता है। Python shell को start करने के लिए आप command prompt window में निचे दी गयी command execute करते है।

```
python
```

यदि आपके system में python installed होगी और आपने PATH set किया होगा तो आपको python shell की string show होगी। जैसे की निचे दी गयी है।

```
>>>
```

इस string का मतलब होता है की आप commands execute कर सकते है। Python shell से बाहर निकलने के लिए आप Ctrl + z keys को press करके enter key press करते है।

Script Mode

Interactive mode छोटे codes को execute करने के लिए उपयोगी है लेकिन इसके साथ एक problem है। Interactive mode में जो commands आप execute करते है वो कँही भी save नहीं होती है।

मान लीजिये आपको same code को बार बार execute करने की आवश्यकता पड़ सकती है। ऐसे में यदि आप interactive mode को use करते है तो आपको उस code को बार लिखना होगा और execute करना होगा।

इस समस्या से बचने के लिए आप python interpreter का script mode use कर सकते है। Script mode में आप एक file में code को लिख लेते है और इसके बाद उसे .py extension से save कर लेते है।

इसके बाद आप अपना command prompt open करके उस directory को access करते है जिसमें आपने python code को save किया था और निचे दिए गए तरीके से command लिखकर उसे execute करते है।

```
python file-name.py
```

इस प्रकार script mode को बहुत अधिक code को execute करते समय use किया जा सकता है।

Python in Hindi – Syntax

March 29, 2019August 15, 2018 by Javahindi.com

- Introduction to **python syntax in Hindi**
- *Python indentations in Hindi*
- Python comments in Hindi

Introduction to Python Syntax

C, Perl, Java आदि programming languages और python के बीच कई समानताएँ पायी जाती है। यदि आपने इन languages में पहले programming की है तो python को आप और भी आसानी से सिख पायेंगे।

लेकिन जैसा की आप पहले पढ़ चुके है python का syntax दूसरी programming languages से अलग है। उदाहरण के लिए जँहा दूसरी programming languages में space का अधिक महत्व नहीं होता है वँही python में spaces बहुत महत्वपूर्ण होते है।

Python में programming शुरू करने से पूर्व आपको इसके syntax से related कुछ जरुरी बातें जानना आवश्यक है। इन्हें आप python syntax rules भी कह सकते है। इनसे आपको python और दूसरी programming languages के बीच syntax का difference पता चलता है।

Python Indentation

Python में किसी class या function के code block को represent करने के लिए braces नहीं use किये जाते है। किसी भी प्रकार के code block को represent करने के लिए line indentation use किया जाता है।

```
if 10 < 5:
 print("10 is less than 5")
```

```
else:
 print(“10 is greater than 5”)
```

Line indentation किसी भी line में text से पूर्व दिया गया space होता है। हालाँकि आप कितने भी space indentation के रूप में दे सकते है लेकिन उस block के सभी statements के लिए indentation समान होना चाहिए।

ऐसा नहीं होने पर python interpreter द्वारा error generate की जाती है।

Python Multiline Statements

Python में एक line में एक ही statement लिखा जाता है। यदि आप कोई ऐसा statement लिख रहे है जो multiple lines में आता है तो दूसरी line में लिखे गए code को अलग statement माना जायेगा। हो सकता है की इससे error generate हो जाए।

किसी statement को multiple lines में लिखने के लिए python में line continuation character () का प्रयोग किया जाता है।

```
if num == 0
    and sum == 100:
     print(“Something is wrong”)
```

यदि statement brackets जैसे की [], {}, () आदि में लिखा गया है तो उसे multiple lines में लिखने के लिए आपको line continuation character की आवश्यकता नहीं होती है।

Python Quotes for Strings

Strings को define करने के लिए python में single (‘), double (“) और triple (“””) quotes use किये जाते है। Single और double normal single line strings के लिए प्रयोग किये जाते है।

```
msg = 'Python is simple.'
msg2 = "Python is different."
```

Python में triple quotes multiline strings को define करने के लिए use किये जाते है। यदि आप किसी document या article को string के रूप में python program में add करना चाहते है तो triple quotes के बीच उसे define कर सकते है। ऐसी strings को python में docstrings कहा जाता है।

```
myMsg = """this is a multiline docstring
in python. This string can be written in many
lines and it must be closed with triple quotes."""
```

आपको एक बात ध्यान रखनी चाहिए की starting और ending quotes same होने चाहिए।

Single Line Multiple Statements

वैसे तो python में एक line में एक ही statement लिखा जाता है। लेकिन यदि आप एक line में multiple statements define करना चाहते है तो इसके लिए आप semicolon का प्रयोग करते है।

Semicolon को आप एक statement के बाद define करते है और उसके बाद दूसरा statement लिखते है। इससे python उन statements को अलग अलग interprete करता है।

```
print("first statement");print("second statement")
```

आखिर statement के बाद semicolon define करने की आवश्यकता नहीं होती है।

Blank Lines

कोई भी blank line जिसमे सिर्फ whitespace होता है वह python interpreter द्वारा ignore कर दी जाती है। हालाँकि यदि आप interactive mode में programming कर रहे है तो multiline statement के बाद आप एक blank line produce करते है।

Suites

Python में multiple statements के group को suites कहा जाता है।

Compound Statements Header Line

Python में compound statements जैसे की if, else, while और class आदि के लिए header line define की जाती है।

Header में उस statement का नाम और उसके बाद colon (:) को define किया जाता है।

```
if (condition) :
```

Header line के बाद suite (group of statements) define किया जाता है जो की multiple statements का group होता है।

Comments in Python

Python में comments define करने के लिए hash symbol (#) का प्रयोग किया जाता है। लेकिन यह symbol quotes में नहीं define किया जाना चाहिए।

Hash symbol के बाद उस line में लिखा गया सारा text comment माना जाता है और python interpreter उसे ignore करता है।

```
# this is a comment.
```

```
print("comments are good") #this is another comment
```

आप किसी statement के बाद भी hash symbol define करके comment लिख सकते है।

Python in Hindi – First Program

March 29, 2019August 17, 2018 by Javahindi.com

Python में आप कोई भी code दो प्रकार से लिख और execute करवा सकते है। पहला तो interactive mode और दूसरा script mode होता है। Python interpreter इन दोनों ही modes में कार्य करता है।

Python interpreter और modes के बारे में briefly एक separate tutorial में बताया जा चूका है। यँहा पर यह assume किया जा रहा है आप यह tutorial पढ़ चुके है।

इस tutorial में आपको इन दोनों modes में अपना पहला python program run करना बताया जा रहा है।

Interactive Mode (Shell)

Interactive mode में अपना पहला program run करने के लिए सबसे पहले आप python command line interpreter या python IDLE को start कीजिये और उसमे निचे दिया गया code type कीजिये।

```
print("Hello Python")
```

ऊपर दिए गए code को लिखने के बाद enter press कीजिये। आपके enter press करते है ऊपर दिया गया code execute हो जायेगा और आपको निचे दी गयी images की तरह output show होगा।

Interactive mode में code save नहीं होता है उसे साथ ही execute कर दिया जाता है। बड़े code को save करने और उसे दुबारा use करने के लिए script mode use किया जाता है।

Script Mode

Script mode में python program run करने के लिए सबसे पहले आप किसी text editor में अपना python program (code) लिखते है। आप चाहे तो इसके लिए python IDLE के editor को भी use कर सकते है।

Python IDLE को use करने के लिए File menu पर click करके New option को choose कीजिये।

अपने code को लिखने के बाद आप इसे एक नाम देकर .py extension से save करते है। आपके ऐसा करने पर वह file एक python code file का रूप ले लेती है।

जैसे की निचे दी गयी image में HelloWorld एक python code file आपको show हो रही है।

अपना python code लिखने और उसे .py extension के साथ save करने के बाद आप उसे दो प्रकार से execute कर सकते है। पहला तो आप Python IDLE को

use कर सकते है और दूसरा आप python command line interpreter use कर सकते है।

आइये इन दोनों ही तरीकों से आपका पहला python program run करने का प्रयास करते है।

Python IDLE Script Mode

IDLE में script mode में program को execute करने के लिए सबसे पहले आप IDLE को open करते है। इसके बाद आप File menu पर click करते है और open option को choose करते है।

इसके बाद आप .py extension से save की गयी file को ढूँढ़ते है और open करते है।

इसके बाद उस file का code IDLE में खुल जाता है।

इसके बाद आप menu bar में दिए गए Run menu option पर click करते है और फिर उसमें Run Module option पर click करते है। ऐसा करने पर आपका code execute हो जाता है और आपको output show हो जाता है।

आप चाहे तो directly F5 press करके भी program को run करवा सकते है।

Python Command Line Interpreter Script Mode

Python script mode file को command line interpreter से execute करने के लिए सबसे पहले आप Windows Command Prompt को open करते है।

इसके बाद आप command prompt से उस directory को access करते है जिसमे .py extension से save की गयी आपकी python code की file है।

इसके बाद आप उस python file का नाम extension के साथ लिखकर enter press करते है। ऐसा करते ही वह file execute हो जाती है और आपको output show हो जाता है।

आप program के नाम के पूर्व python लिखकर भी उस program को execute कर सकते है।

Python in Hindi – Variables

March 29, 2019August 20, 2018 by Javahindi.com

- Introduction to **python variables in Hindi**
- Declaring *python variables in Hindi*
- Scope of python variables in Hindi

Introduction to Python Variables

एक variable computer की memory में किसी location का नाम होता है। यह नाम उस location पर value store करने और वँहा से value प्राप्त करने के लिए use किया जाता है।

किसी भी variable में value मुख्यतः इसलिए store की जाती है ताकि उसके साथ कोई operations perform किया जा सके या उसे process किया जा सके।

Variables में store की जाने वाली values अलग अलग types की हो सकती है। जैसे की numbers, strings आदि।

Python में variables की values को print करने के लिए print() function use किया जाता है।

किसी text को variable के साथ combine करके print करने के लिए python में + character को use किया जाता है। इसके अलावा आप इस character से एक variable को दूसरे variable से भी जोड़ सकते है।

Python में हर variable एक object है। असल में python में आप किसी variable को एक value नहीं assign करते है बल्कि आप एक object का reference assign करते है जिसमे वह value store की गयी है।

इसका मतलब यह हुआ की python में जब आप कोई variable की value access करने का प्रयास करते है तो असल में आप उस object को point करते जिसमे वह value store की गयी है।

उदाहरण के लिए निचे दिए गए code को देखिये।

```
num = 50
```

ऊपर दिए गए code में 50 के integer object है और num एक नाम है जो उस object को point करता है।

Python में आप एक string और number को directly concatenate नहीं कर सकते है। पहले आप दोनों को separate variables के रूप में declare करते है इसके बाद उन variables को आपस में concatenate किया जा सकता है।

Variable Rules in Python

Python में कुछ variable सम्बधित rules होते है जिन्हें follow करना आवश्यक होता है।

- एक variable का नाम या तो किसी letter या underscore (_) character से शुरू होना चाहिए।
- एक variable के नाम में सिर्फ alphanumeric characters (a-z, 0-9 और _) ही हो सकते है।
- Python के reserved words को आप variable names के रूप में नहीं use कर सकते है।

एक बात आपको हमेशा ध्यान रखनी चाहिए की python एक case sensitive language है। यानी की python में sum, Sum और SUM तीन अलग अलग variables माने जायेंगे।

Declaring Variables in Python

Variables को एक single alphabet से लेकर किसी भी नाम से declare किया जा सकता है।

Python में variables को use करने के लिए पहले से define करने की आवश्यकता नहीं होती है जैसा की कई programming languages में होता है। Python में जैसे ही आप variable को कोई value assign करते है वह declare हो जाता है। यही कारण है की python को dynamically typed language कहा जाता है।

Python में variables को declare करने के लिए data type define करने की भी आवश्यकता नहीं होती है। Python किसी भी value का data type automatically judge कर लेती है। किसी भी variable को value assign करने के लिए assignment operator (=) use किया जाता है।

Python में variable create करने का general syntax निचे दिया जा रहा है।

```
<variable-name> = <value>
```

Python interpreter variable की value के आधार पर ही उसे memory assign करता है। Python में variable create करना निचे उदाहरण द्वारा समझाया जा रहा है।

```
num = 100
```

Python में variables को एक बार declare करने के बाद दुबारा किसी दूसरे type की value से redeclare किया जा सकता है। उदाहरण के लिए ऊपर declare किये गए num variable को दुबारा एक string value से इस प्रकार redeclare किया जा सकता है।

```
num = “Hundred”
```

एक single variable declare करने के अलावा आप एक साथ कई variables भी declare कर सकते है और उन्हें एक ही value assign कर सकते है।

उदाहरण के लिए निचे दिए गए code को देखिये।

```
num1=num2=num3=10
```

ऊपर दिए गए code में 3 variables create किये गए है और उन सभी को 10 value assign की गयी है।

Deleting Variables in Python

Python आपको किसी variable को delete करने की facility भी provide करती है। इसके लिए del command का प्रयोग किया जाता है। Python में variable delete करने का general syntax निचे दिया जा रहा है।

```
<del> <variable-name>
```

उदाहरण के लिए मान लीजिये आप num variable को delete करना चाहते है तो इसके लिए आप इस प्रकार statement लिखते है।

```
del num
```

Swapping Variable Values in Python

Python आपको single statement द्वारा variables की values आपस में swap करने की ability provide करती है। इसके लिए आप निचे दिया गया syntax follow करते है।

```
var1, var2 = var2, var1
```

Python में variables की values swap करना निचे उदाहरण द्वारा समझाया जा रहा है।

```
num1 = 5
num2 = 7
```

```
num1, num2 = num2, num1

print(num1)
print(num2)
```

Local & Global Variables in Python

जब किसी variable को किसी function या code block में define किया जाता है तो वह local variable कहलाता है। जब कोई variable function के बाहर global scope में declare किया जाता है तो उसे global variable कहते है।

Python में जब आप किसी variable को पुरे program में use करना चाहते है तो उसे global variable declare करते है। जबकि यदि आप किसी variable को एक function या code block में use करना चाहते है तो use local variable declare करते है।

यदि किसी function में आप किसी global variable को refer करना चाहते है तो इसके लिए global keyword use करते है।

Python in Hindi – Numbers

March 29, 2019August 22, 2018 by Javahindi.com

- Introduction to **python numbers in Hindi**
- Different python numbers in Hindi

Introduction to Python Numbers

Mathematical operations perform करने के लिए यह आवश्यक है की एक programming language में अलग अलग तरह के numbers के लिए support available हो।

Python में सभी प्रकार के numbers का एक type (class) होता है जो numbers का structure और functionality define करता है। इस type से यह पता चलता है की number किस प्रकार से store किया जायेगा और उसके साथ कौनसे operations perform किये जा सकते है।

Python में सभी number types को classes के रूप में implement करने से आप यह check कर सकते है की कोई number किसी specific type का है या नहीं। इसके लिये python type() और isinstance() जैसे functions provide करती है।

Python 3 numerical types को support करती है।

- Integer
- Floating Point
- Complex

Integer और floating point numbers में operation का order लगभग समान रहता है। सबसे पहले multiplication और division perform किया जाता है, इसके बाद addition और subtraction perform किया जाता है।

Integer

Python में integer numbers int class से सम्बंधित होते है। Integer numbers पूर्ण numbers होते है। इनमें दशमलव नहीं होता है। एक integer negative, positive या zero हो सकता है।

उदाहरण के लिए निचे दिए गए numbers को देखिये ये सभी integers है।

```
7, 0, -2, 9, -55
```

जब एक integer negative होता है तो वह signed integer कहलाता है।

Floating Numbers

Python में floating point numbers float class से सम्बंधित होते है। Floating point numbers ऐसे numbers होते है जिनमें दशमलव और उसके बाद कुछ सँख्या होती है।

Floating point numbers negative या positive हो सकते है।

उदाहरण के लिए निचे दिए गए numbers को देखिये।

```
4, 5.9, -2.2
```

ऊपर दिए गए numbers में 4 एक integer है और 5.9 एक floating point number है।

Floating point numbers को 10 की power दर्शाने के लिए e के साथ एक scientific number के रूप में भी use किया जाता है।

Floating point numbers के सम्बन्ध में एक बात आपको हमेशा यह ध्यान रखनी चाहिए की floating point numbers पर किये गए operations का result भी एक floating point number ही होता है।

उदाहरण के लिए यदि आप 2 और 2.0 को add कर रहे है तो इनका result 4.0 होगा।

Complex Numbers

Python में complex numbers complex class से सम्बंधित होते है। Complex numbers के real और imaginary दो parts होते है।

एक complex number a+bi की form में represent किया जाता है। इसमें a और b real numbers (floats) होते है और i एक imaginary number होता है।

Python में कुछ तकनीकी कारणों की वजह से imaginary number को j के द्वारा represent किया जाता है।

Python in Hindi – Strings

March 29, 2019August 24, 2018 by Javahindi.com

- Introduction to **python strings in Hindi**
- *Python string operators in Hindi*
- Python string methods in Hindi

Introduction to Python Strings

एक string characters की sequence होती है। यानी की जब characters को एक क्रम में organize किया जाता है तो वे string बन जाते है।

Python में सब कुछ object है और एक string भी object ही होती है। Python में strings immutable objects होती है। यानि एक बार string create करने के बाद आप उसे change नहीं कर सकते है। किसी भी दूसरे variable की तरह del command से आप string को भी delete कर सकते है।

Python में string create करने के लिए आप एक variable create करते है और single या double quotes में string को लिखकर उसे assign करवाते है।

उदाहरण के लिए निचे दिए गए code में एक string message create किया गया है।

```
message = "Hello Reader"
```

Python में strings को create करते समय आप single या double कोई से भी quotes use कर सकते है। Python में ये दोनों same ही माने जाते है। यानि की ऊपर दिए गए code को इस प्रकार भी लिखा जा सकता है।

```
message = 'Hello Reader'
```

Python में character data type available नहीं है। Python में character create करने के लिए आप single character से string create करते है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
language = “C”
```

Python में strings को print() function के द्वारा screen पर output किया जा सकता है। उदाहरण के लिए निचे दिए गए code में एक string को print function द्वारा screen पर show किया जा रहा है।

```
print(message)
```

दूसरी modern programming languages की तरह python में भी strings array के रूप में represent की जाती है। String के elements को index numbers assigned होते है जिनके द्वारा उन्हें access किया जा सकता है। इसके लिए विभिन्न operators का प्रयोग किया जाता है जिनके बारे में आप आगे जानेंगे।

Python String Operators

Python में strings के साथ कार्य करने के लिए कुछ operators provide किये गए है। इनके बारे में आगे बताया जा रहा है।

+

यह operators दो strings को जोड़ने के लिए use किया जाता है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
msg = “Hello”
name = “Reader”
```

```
print msg+name
```

ऊपर दिए गए उदाहरण में + operator द्वारा msg और name strings को concatenate करके print करवाया गया है।

*****

यह किसी भी string को number of times print करने के लिए use किया जाता है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
msg = “Hello”
print msg*2
```

ऊपर दिए गए उदाहरण में * operator के द्वारा msg string को 2 बार print करवाया गया है।

[]

यह string के elements को access करने के लिए प्रयोग किया जाता है। उदाहरण के लिए string के first element को access करने के लिए आप इसे इस प्रकार use कर सकते है।

```
msg = “Hello Python”
print(msg[0])
```

ऊपर दिए गए उदाहरण में Hello Python string के first element H को print करवाया गया है। एक बात आपको हमेशा ध्यान रखनी चाहिए की python में किसी भी array की तरह string elements की index भी zero से शुरू होती है। यानि की string का first element [0] position पर होगा।

[:]

यह किसी string से substring extract करने के लिए प्रयोग किया जाता है। इसमें colon के एक तरफ आप substring की starting position लिखते है और दूसरी और ending position लिखते है। इसके बाद execute होने पर उन position के बीच की substring return की जाती है।

उदाहरण के लिए निचे दिए गए code को देखिये।

```
msg = “Hello Python”


print(msg[0:4])
```

ऊपर दिए गए उदाहरण में Hello Python string से Hello substring को print करवाया गया है।

in

यह operator किसी string में कोई character है या नहीं यह पता लगाने के लिए प्रयोग किया जाता है। यदि वह character available है तो true (1) return किया जाता है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
msg = “Hello Python”
print “o” in msg
```

ऊपर दिए गए उदाहरण में क्योंकि Hello Python string में o available है इसलिए true return किया जायेगा।

not in

यह operator in operator के विपरीत कार्य करता है। यानि की in operator जँहा string में कोई character available होने पर true return करता है वँही

दूसरी तरफ not in operator string में कोई character नहीं होने पर true return करता है।

उदाहरण के लिए निचे दिए गए code को देखिये।

```
msg = “Hello Python”
print “a” not in msg
```

ऊपर दिए गए code को execute किये जाने पर true return होगा क्योंकि a Hello Python string में available नहीं है।

%

यह operator string formatting के लिए प्रयोग किया जाता है। इस operator के साथ मुख्यतः d और s specifiers को use किया जाता है। इसमें %d तो numbers को format करके display करवाने के लिए और %s को string को display करवाने के लिए use किया जाता है। इसका उदाहरण निचे दिया जा रहा है।

```
name = “Javahindi.com”
Age = 100


print(“%s, %d”,name,Age)
```

ऊपर दिए गए उदाहरण में % operator के द्वारा name और Age print करवाई गयी है।

Python String Methods

Python में str एक built in class है। यह class strings के साथ work करने के लिए important methods provide करती है। क्योंकि python में सभी strings

objects होती है। इसलिए आप str class के methods को strings से आसानी से call कर सकते है।

strip()

यह method string के beginning या end से whitespace को remove करता है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
msg = ” Good Morning “
print(msg.strip())
```

ऊपर दिए गए उदाहरण में string को create करते समय पहले और बाद में space दिया गया है। Strip method के execute किये जाने पर यह space remove हो जाता है।

len()

यह method किसी भी string की length return करता है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
msg = “hi there…”
print(msg.len())
```

ऊपर दिए गए उदाहरण में msg string की length को len() method द्वारा print करवाया गया है।

lower()

यह method किसी भी string को lowercase में return करता है। इसका उदाहरण निचे दिया जा रहा है।

```
msg = “HELLO”
```

```
print(msg.lower())
```

ऊपर दिए गए उदाहरण में HELLO string को lower() method द्वारा lowercase में convert किया गया है।

upper()

यह method किसी string को uppercase में return करता है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
msg = “hello”
print(msg.upper())
```

ऊपर दिए गए उदाहरण में hello string को upper() method द्वारा HELLO में convert करके print करवाया गया है।

replace()

यह method एक string को दूसरी string से replace करता है। इसका उदाहरण निचे दिया गया है।

```
msg = “Good Morning”
print(msg.replace(G, F))
```

ऊपर दिए गए उदाहरण में string Good Morning से G को F द्वारा replace किया गया है।

split()

यह method string को split करता है। इस method में आप argument के रूप में एक separator pass करते है। इसी separator के आधार पर string को split किया जाता है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
msg = “Good, Morning”
print(msg.split(“,”))
```

ऊपर दिए गए उदाहरण में string Good, Morning को comma के आधार पर split() method द्वारा separate किया गया है।

isalpha()

यह method test करता है की क्या string alphanumeric है। यदि string alphanumeric है तो true return किया जायेगा। इसका उदाहरण निचे दिया जा रहा है।

```
msg = “70 Independence Day”
print(msg.isalpha())
```

ऊपर दिए गए उदाहरण में क्योंकि msg एक alphanumeric string है इसलिये true return होगा।

isdigit()

यह method test करता है की क्या एक string digit है। Digit होने पर true return किया जाता है। इसका उदाहरण निचे दिया जा रहा है।

```
num = “99”
print(msg.isdigit())
```

ऊपर दिए गए उदाहरण में num एक digit string है इसलिए true (1) print किया जायेगा।

isspace()

यह method test करता है की क्या कोई specific string space character है। यदि string एक space character है तो true return किया जायेगा। इसका उदाहरण निचे दिया जा रहा है।

```
msg = ” “
print(msg.isspace())
```

ऊपर दिए गए उदाहरण में msg एक space string है इसलिए true return किया जायेगा।

startswith()

यह method check करता है की क्या string किसी pass की गयी string से start होती है। यदि start होती है तो true return किया जाता है।

```
msg = “Hello Python”
print(msg.startswith(‘Hello’))
```

ऊपर दिए गए उदाहरण में क्योंकि msg string Hello से शुरू होती है इसलिए true return किया जायेगा।

find()

यह method इसमें pass की गयी string को call की गयी string में ढूंढता है। यदि वह string मिलती है तो उसकी starting index return की जाती है नहीं तो -1 return किया जाता है। इसका उदाहरण निचे दिया जा रहा है।

```
msg = “Hello Python”
print(msg.find(‘Python’))
```

ऊपर दिए गए उदाहरण में क्योंकि Python string exits करती है इसलिए उसकी starting index return की जाएगी।

join()

यह method split() method का उल्टा होता है। इस method द्वारा किसी specific string द्वारा दो strings को जोड़ा जाता है। इसका उदाहरण निचे दिया जा रहा है।

```
numList = ['1','2','3','4']
x = "-"
print(x.join(numList))
```

ऊपर दिए गए उदाहरण में – string द्वारा 1, 2, 3, 4 numbers को जोड़ा गया है।

Python in Hindi – Lists

March 29, 2019August 27, 2018 by Javahindi.com

- Introduction to **python lists in Hindi**
- *Creating python lists in Hindi*
- Python list operations in Hindi

Introduction to Python Lists

Python में list एक ordered collection होता है। Lists में duplicate members add किये जा सकते है और यह changeable होता है।

Python lists के बारे में एक अद्भुत बात यह है की इसे अलग अलग types के items के द्वारा create किया जा सकता है।

एक list collection के अंदर आप run time के दौरान भी नए members add कर सकते है।

Lists में add किये गए सभी elements objects ही होते है। इसलिए python में list को आप objects का ordered collection भी कह सकते है।

एक list के अंदर member के रूप में दूसरी list को भी add किया जा सकता है। ऐसी lists को sublists या nested lists कहा जाता है।

Python में lists List class के objects होती है।

Creating Python Lists

Python में lists को square brackets द्वारा define किया जाता है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
myList = [“Cow”,”Camel”,”Cat”]
print (myList)
```

ऊपर दिए गए उदाहरण में myList के नाम से एक list collection create किया गया है जिसमें क्रमशः Cow, Camel और Cat strings को add किया गया है। बाद में print statement द्वारा इस list को print किया गया है।

Python में आप एक और तरीके से list collection define कर सकते है जिसमे list() constructor का प्रयोग किया जाता है। List constructor द्वारा list define करने के लिए list() constructor के अंदर members को pass किया जाता है।

Members को define करने के लिए square brackets की बजाय round brackets का प्रयोग किया जाता है और उन्हे list() constructor के अंदर comma separated list के रूप में define किया जाता है।

उदाहरण के लिए निचे दिए गए code को देखिये।

```
myList = list((“Cow”,”Camel”,”Cat”))
print(myList)
```

ऊपर दिए गए उदाहरण में पहले create की गयी myList को दुबारा list() constructor के प्रयोग से create करना बताया गया है। जैसा की आप साफ़ तौर पर देख सकते है इसमें list() constructor के अंदर members को round brackets के अंदर comma से spearate करके pass किया गया है।

यह python की list class का constructor होता है और आपको list object provide करता है। List constructor के प्रयोग को double round brackets से भी identify किया जा सकता है।

जैसा की पहले बताया गया यह आवश्यक नहीं है की list के सभी items same type के हो। आप अलग अलग types जैसे की numbers, strings आदि की

values को एक ही list में add कर सकते है। उदाहरण के लिए निचे दिया गया code देखिये।

```
myList = list((“Cow”,50,Camel”20",”Cat”,10))
print(myList)
```

ऊपर दिए गए उदाहरण में numbers और strings के प्रयोग से myList list को create किया गया है।

Python Lists Operations

Strings की ही तरह list items भी indexed होते है। इनकी index भी zero से ही शुरू होती है और strings की ही तरह इनको भी slice, concatenate किया जा सकता है। इसके लिए आप strings के लिए प्रयोग किये जाने वाले operators का प्रयोग करते है। इनके बारे में आगे बताया जा रहा है।

List में किसी भी item को access करने के लिए आप index operator [] के साथ उसका index number use करते है।

```
myList = list((“Cow”,”Camel”,”Cat”))
print(myList[1])
```

List में से कुछ members को slice (एक निश्चित range की बीच members को access) करने के लिए strings की ही तरह square brackets में colon का प्रयोग किया जाता है।

```
myList = list((“Cow”,”Camel”,”Cat”,”Dog”,”Elephant”))
print(myList[1:3])
```

Remove method के अलावा list के किसी भी member को del keyword द्वारा भी delete किया जा सकता है। इसके लिए आप उस member के index number का प्रयोग करते है।

```
myList = list((“Cow”,”Camel”,”Cat”))
print(myList)
del myList[1]
print(myList)
```

आप दो list को + operator द्वारा strings की तरह concatenate कर सकते है।

```
myList1 = list((“Cow”,”Camel”,”Cat”))
myList2 = list((“Dog”,”Elephant”))
print(myList1+myList2)
```

इसके साथ ही आप * operator द्वारा एक list को multiple times print भी करवा सकते है।

```
myList = list((“Cow”,”Camel”,”Cat”))
print myList*2
```

आप in operator द्वारा यह भी check कर सकते है की list में कोई member है या नहीं, जैसा की आप strings के साथ करते है। यदि कोई element list में available होता है तो true return किया जाता है।

```
myList = list((“Cow”,”Camel”,”Cat”))
print “Cow” in myList
```

Python List Functions and Methods

सभी methods को list object द्वारा call किया जाता है और functions को independently call किया जाता है।

append()

List में कोई भी नया item जोड़ने के लिए append() method का प्रयोग किया जाता है। इस method के अंदर argument के रूप में आप वह item pass करते है जिसे आप list में जोड़ना चाहते है। इसका उदाहरण निचे दिया जा रहा है।

```
myList = list((“Cow”,”Camel”,”Cat”))
print(myList)
myList.append(“Cobra”)
print(myList)
```

remove()

List में से किसी भी existing item को remove करने के लिए remove() method का प्रयोग किया जाता है। इस method में argument के रूप में आप वह item pass करते है जिसे आप list से remove करना चाहते है। इसका उदाहरण निचे दिया जा रहा है।

```
myList = list((“Cow, “Camel”,”Cat””))
print(myList)
myList.remove(“Camel”)
print(myList)
```

len()

किसी भी list की length यानि की उसमें कितने items है प्राप्त करने के लिए len() function का प्रयोग किया जाता है। इस function में argument के रूप में list को pass किया जाता है। इसका उदाहरण निचे दिया जा रहा है।

```
myList = list((“Cow”,”Camel”,”Cat”))
print(len(myList))
```

clear()

यह function किसी list से सभी elements को remove करने के लिए use किया जाता है। इससे list खाली हो जाती है।

```
myList = list((“Cow”,”Camel”,”Cat”))
print(myList)
myList.clear()
print(myList)
```

count()

len() function की ही तरह list में members की सँख्या बताने के लिए count() method भी available है। इसे आप string object के साथ call करते है।

```
myList = list((“Cow”,”Camel”,”Cat”))
print (myList.count())
```

sort()

इस method द्वारा किसी list को sort किया जा सकता है।

```
myList = list((1,2,4,3))
print (myList.sort())
```

Python in Hindi – Tuples

March 29, 2019August 29, 2018 by Javahindi.com

- Introduction to **python tuples in Hindi**
- Creating *python tuples in Hindi*
- Methods of python tuples in Hindi

Introduction to Python Tuples

Python में list की ही तरह tuple भी एक ordered collection है। इन दोनों के साथ perform किये जाने वाले operations भी लगभग समान ही है।

इन दोनों में फर्क सिर्फ इतना है tuples unchangeable या immutable होते है। एक बार tuple create करने के बाद आप उसके items को change नहीं कर सकते है।

दूसरा फर्क इनमें यह होता है की tuple को round brackets के साथ create किया जाता है। हालाँकि आप tuples को बिना round brackets के भी create कर सकते है लेकिन round brackets use करना एक अच्छी practice मानी जाती है।

इनके अलावा एक और जो मुख्य अंतर इन दोनों में होता है वह यह है की आप tuples के items को remove नहीं कर सकते है।

Creating Python Tuples

Python में tuples को list की तरह ही create किया जाता है। लेकिन tuples को create करते समय round brackets use किये जाते है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
myTuple = (“Red”,”Green”,”Blue”)
print(myTuple)
```

ऊपर दिए गए उदाहरण में myTuple के नाम से एक tuple create किया गया है। इस tuple में Red, Green और Blue items को add किया गया है।

यदि आप एक item का tuple create करना चाहते है तो इसके लिए आप round brackets में सिर्फ एक item define करते है लेकिन उस item के बाद comma लगाना अनिवार्य होता है। यदि आप comma नहीं लगाते है तो error generate होती है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
myTuple = (“Red”,)
print(myTuple)
```

ऊपर दिए गए उदाहरण में एक single item वाला tuple create किया गया है लेकिन जैसा की आप देख सकते है item को double quotes में define करने के बाद comma लगाया गया है। यदि आप एक ही item से tuple create कर रहे तो ऐसा करना आपके लिए अनिवार्य है।

यदि आप empty tuple create करना चाहते है तो इसके लिए आप tuple के नाम के बाद only round brackets define करते है। उन brackets को empty रखा जाता है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
myTuple = ()
```

ऊपर दिए गए उदाहरण में myTuple के नाम से एक empty tuple create किया गया है। कोई दूसरा tuple इस tuple को assign किया जा सकता है।

Python में tuples create करने के लिए आप tuple() constructor भी use कर सकते है। इसका उदाहरण निचे दिया जा रहा है।

```
myTuple = tuple((“Red”,”Green”,”Blue”))
print(myTuple)
```

ऊपर दिए गए उदाहरण में myTuple tuple को tuple() constructor द्वारा create किया गया है। जब आप tuple() constructor द्वारा tuple create करते है तो members को double round brackets में add किया जाता है।

Accessing Python Tuples

Lists की ही तरह tuple members की भी indexing होती है। आप किसी भी specific member को उसके index number से extract कर सकते है। Index numbers को index operator [] के साथ use किया जाता है। Tuples की index array और list की तरह 0 से शुरू होती है।

इसका उदाहरण निचे दिया जा रहा है।

```
myTuple = tuple((“Red”,”Green”,”Blue”))
print(myTuple[1])
```

ऊपर दिए गए उदाहरण में myTuple tuple के second element Green को print करवाया गया है।

इसके अलावा आप tuple members को slicing के द्वारा भी access कर सकते है। इस operation में आप जिस range के members को access करना चाहते है उनकी starting और ending index numbers define करते है।

Starting index को square brackets में colon से पहले और ending index number को colon के बाद define किया जाता है। इसका उदाहरण निचे दिया जा रहा है।

```
myTuple = tuple((“Red”,”Green”,”Blue”,”White”,”Black”))
print(myTuple[1:3])
```

ऊपर दिए गए उदाहरण में 2 से 4 range के बीच के tuple elements को print करवाया गया है।

Deleting Python Tuples

जैसा की मैने पहले बताया tuples में individual elements को delete करना possible नहीं है। लेकिन आप existing tuple को दूसरे tuple के साथ combine करके एक नया tuple बना सकते है और उसमे से वो items remove कर सकते है जो आप नहीं चाहते है।

इसके अलावा यदि आप सम्पूर्ण tuple को delete करना चाहते है तो del command को use कर सकते है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
myTuple = tuple((“Red”,”Green”,”Blue”))
print(myTuple)
del myTuple
```

ऊपर दिए गए उदाहरण में del command द्वारा myTuple को delete किया गया है।

Advantages of Python Tuples

निचे lists के comparison में tuples की कुछ advantages बताई जा रही है।

- Tuple elements को list elements के comparison में तेजी से iterate किया जा सकता है।
- Tuple elements को dictionary collection में एक key के रूप में use किया जा सकता है। ऐसा lists के साथ करना संभव नहीं है।
- यदि आप चाहते की आपका data किसी प्रकार change नहीं किया जा सके तो उसे tuple में store करना सबसे बेहतर होता है।

Tuples और lists दोनों similar tasks के लिए भी use किये जाते है लेकिन कुछ ऐसे operations होते है जिनमें सिर्फ tuples ही appropriate होते है।

Functions of Python Tuples

Python में tuples के साथ use करने के लिए निचे दिए जा रहे functions available है।

- all() – यह function true return करता है यदि tuple के सभी elements true होते है। इसके अलावा यदि tuple empty है तो भी यह function true return करता है।
- any() – यह function true return करता है यदि tuple का कोई भी element true होता है। यदि tuple empty होता है तो यह function false return करता है।
- enumerate() – यह function enumerate object return करता है। Enumerate object में सभी tuple members उनके index number pairs के रूप में stored रहते है।
- len() – यह function tuple में items की सँख्या return करता है।
- max() – यह function tuple में largest item को return करता है।
- min() – यह function tuple से smallest item return करता है।
- sorted() – यह function एक tuple को input के रूप में लेता है और एक sorted list return करता है।
- sum() – यह function tuple के सभी elements का sum return करता है।
- tuple() – इस function द्वारा tuple का iterator create किया जाता है।

इन methods के अलावा आप + operator द्वारा दो tuples को concatenate भी कर सकते है और * द्वारा multiple times print भी करवा सकते है।

Python in Hindi – Dictionaries

March 29, 2019August 31, 2018 by Javahindi.com

- Introduction to **python dictionaries in Hindi**
- Creating *python dictionaries in Hindi*
- Accessing python dictionaries in Hindi

Introduction to Python Dictionaries

Python आपको एक और Lists जैसा collection provide करती है जिसे dictionary कहा जाता है। Lists की ही तरह एक dictionary को आसानी से change किया जा सकता है और आवश्यकतानुसार घटाया और बढ़ाया जा सकता है।

List और dictionary में फर्क यह होता है की list में elements ordered होते है, लेकिन एक dictionary में elements ordered नहीं होते है।

इसके अलावा dictionary collection indexing को support नहीं करता है। इसका मतलब यह हुआ की dictionary में किसी element को आप उसकी index या position के द्वारा access नहीं कर सकते है। Dictionary में elements को keys के द्वारा access किया जाता है।

Dictionary में कोई भी element key और value के pair में store किया जाता है। हर key एक value को point करती है या फिर उसे store और access करने के लिए use की जाती है।

एक dictionary के अंतर्गत किसी key की value python के किसी भी type की हो सकती है। यानी की strings, lists, tuples और dictionary को आप key की value के रूप में use कर सकते है।

लेकिन dictionary में आप हर type की key नहीं define कर सकते है। Dictionary में key define करने के लिए आप सिर्फ immutable (ऐसे types जो

unchangeable है या जिनकी values को change नहीं किया जा सकता है) types को ही use कर सकते है। यानी tuples, strings, integers, floats आदि को key के रूप में use किया जा सकता है।

आप dictionary में value के रूप में किसी list को store कर सकते है और एक list में भी value के रूप में dictionary को store किया जा सकता है।

Dictionary collection list, tuples आदि के साथ perform होने वाले operations को support नहीं करता है। आप dictionary के साथ concatenation आदि operations perform नहीं कर सकते है। Dictionary का मुख्य उद्देश्य mapping होता है। यानि की एक key को एक value से map करना।

Creating Python Dictionaries

एक Dictionary के items को curly brackets में define किया जाता है। Dictionary में items को key:value के pair में लिखा जाता है। Key और value को colon से separate किया जाता है। Colon के left side में key होती है और right side में value होती है। Key:value के हर pair को comma से separate किया जाता है।

उदाहरण के लिए निचे दिए गए code को देखिये।

```
myDict = {“c1":”Red”,”c2":”Green”,”c3":”Blue”}
print(myDictionary)
```

ऊपर दिए गए उदाहरण में myDict के नाम से एक dictionary collection create किया गया है। इस dictionary में c1, c2 और c3 keys है और Red, Green और Blue values है। एक बात आपको ध्यान रखनी चाहिए की string keys और string values को हमेशा double quotes में ही लिखा जाना चाहिए।

Tuples को आप keys के रुप में use कर सकते है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
myDict = {(Apple, Chilli, Tomato):”Red”,(Calabash,
Cabbage,Beans):”Green”}
```

ऊपर दिए गए उदाहरण में tuples को keys के रूप में use किया गया है।

यदि आप एक existing dictionary में कोई नया item add करना चाहते है तो ऐसा आप एक नया नाम और नयी value define करके करते है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
myDict = {“c1":”Red”,”c2":”Green”,”c3":”Blue”}
print(myDict)



myDict[“c4”] = “Lime”
print(myDict)
```

ऊपर दिए गए उदाहरण में myDict dictionary में c4 key और Lime value बाद में separately add की गयी है।

आप dict() constructor द्वारा भी dictionary create कर सकते है। जब आप dict() constructor के प्रयोग से dictionary create करते है तो keys को double quotes में नहीं लिखते है और colon (:) की जगह assignment operator (=) का प्रयोग करते है। इसका उदाहरण निचे दिया जा रहा है।

```
myDict = dict(c1=”Red”,c2=”Green”,c3=”Blue”)
print(myDict)
```

ऊपर दिए गए उदाहरण में dict() constructor द्वारा dictionary collection create किया गया है।

Accessing Python Dictionaries

किसी dictionary से items access करने के लिए उन्हें keys के द्वारा refer किया जाता है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
myDict = {“c1":”Red”,”c2":”Green”,”c3":”Blue”}
print(myDict[“c1”])
```

ऊपर दिए गए उदाहरण में c1 key द्वारा Red value को print किया गया है।

किसी dictionary में से value access करने के लिए आप get() method का भी प्रयोग कर सकते है। यह method argument के रूप में key को लेता है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
myDict = {“c1":”Red”,”c2":”Green”,”c3":”Blue”}
print(myDict.get(“c2”))
```

ऊपर दिए गए उदाहरण में get() method द्वारा c2 key की value Green को print किया गया है।

आप किसी dictionary के सभी elements को print करने के लिए उसे iterate कर सकते है। इसके लिए for loop और in operator का प्रयोग किया जाता है। उदाहरण के लिए निचे दिए गए code को देखिये।

```
myDict = {“c1":”Red”,”c2":”Green”,”c3":”Blue”}


for key in myDict
print(myDict[key])
```

ऊपर दिए गए उदाहरण में for loop द्वारा myDict dictionary के सभी elements को print किया गया है। जब आप इस प्रकार किसी dictionary को iterate करते है तो dictionary की keys return की जाती है।

इसके अलावा आप किसी dictionary की keys को iterate करने के लिए iterkeys() method और values को iterate करने के लिए itervalues() method use कर सकते है। उदाहरण के लिये निचे दिए गए code को देखिये।

```
myDict = {“c1":”Red”,”c2":”Green”,”c3":”Blue”}



for key in MyDict.iterkeys()
print(myDict[key])

for value in myDict.itervalues()
print(myDict[value])
```

ऊपर दिए गए उदाहरण में myDict dictionary की keys और values को for loop के साथ iterkeys() और itervalues() methods को प्रयोग करके print किया गया है।

इन दोनों functions के अलावा आप items() function को भी use कर सकते है। यह function keys और values को एक साथ return करता है।

Updating Python Dictionaries

Dictionaries में यदि आप किसी member की value को change करना चाहते है तो ऐसा आप उसकी key के द्वारा कर सकते है। आप key के द्वारा उस member को नयी value assign करवा सकते है।

```
myDict = {“c1":”Red”,”c2":”Green”,”c3":”Blue”}
```

```
print(myDict)

myDict[“c1”] = “Lime”

print(myDict)
```

ऊपर दिए गए उदाहरण में c1 key की value को Red से Lime में change किया गया है।

Deleting Python Dictionaries

Dictionaries में से items को delete करने के लिए आपको tuples की तरह उन्हें combine करने की आवश्यकता नहीं होती है। Del keyword द्वारा आप किसी specific key वाले item को remove कर सकते है। उदाहरण के लिए निचे दिया गया code देखिये।

```
myDict = {“c1":”Red”,”c2":”Green”,”c3":”Blue”}
print(myDict)



del myDict[“c2”]
print(myDict)
```

ऊपर दिए गए उदाहरण में del command द्वारा second item को remove किया गया है।

इसके अलावा आप pop() method द्वारा भी किसी item को remove कर सकते है। इस method में key को argument के रूप में pass किया जाता है। इसका उदाहरण निचे दिया जा रहा है।

```
myDict = {“c1":”Red”,”c2":”Green”,”c3":”Blue”}
print(myDict)
```

```
myDict.pop("c3")
print(myDict)
```

ऊपर दिए गए उदाहरण में pop() method द्वारा myDict dictionary के तीसरे item को remove किया गया है।

इसके अलावा dictionary के last item को remove करने के लिए आप popitem() method को use कर सकते है।

Del keyword द्वारा आप एक सम्पूर्ण dictionary को भी delete कर सकते है। इसके अलावा यदि आप एक dictionary को खाली करना चाहते है तो इसके लिए clear() method use कर सकते है। इस method से dictionary के सभी elements delete हो जाते है।

Methods for Python Dictionaries

Python में dictionaries के लिए निचे दिए गए methods available है।

- copy()
- sort()
- len()
- cmp()
- update()
- str()
- items()

इन methods के अलावा आप dictionaries के साथ in और not in जैसे operators भी use कर सकते है।

Python in Hindi – Modules

March 30, 2019October 31, 2018 by Javahindi.com

- Introduction to **python modules in Hindi**
- Python standard modules in Hindi

Introduction to Python Modules

एक बार कोई program लिखने के बाद यदि आप python interpreter को close करके दोबारा start करते है तो आपके द्वारा लिखा गया वह program ग़ायब हो जाता है। ऐसा इसलिए होता है क्योंकि python interpreter के interactive mode में लिखे गए code को store नहीं किया जाता है।

यही कारण है की जिन programs को आप बाद में भी use करना चाहते है उन्हें आप scripting mode में write करते है। Scripting mode में सबसे पहले आप program को लिखते है और .py extension से save कर लेते है। इसके बाद interpreter के माध्यम से आप उस file को execute करते है।

When You Need Same Functionality Across Many Scripts

इस प्रकार scripting mode में program में execute किये गए programs हमेशा available होते है। जब आप scripting mode में कोई बड़ा project या program create करते है तो उसे manage करने के लिए आपको उसे कई अलग अलग files में store करने की आवश्यकता होती है।

जब आप ऐसा करते है तो इन files में कुछ definitions (variables और functions) ऐसे भी हो सकते है जिन्हे project की सभी files में include करने की जरुरत होती है। ऐसे में हर file में अलग से उन definition को define करना बहुत ही time consuming होगा और इससे program की readability भी कम हो जाएगी।

File Containing Python Definitions and Statements

ऐसी situations के लिए python आपको modules का concept provide करती है। एक module एक file होती है जिसमे python definitions जैसे की variables, functions और दूसरे statements को store किया जाता है।

ऐसी definition जिन्हें आप कई multiple scripts में use करना चाहते है उन्हें एक module में define कर सकते है। इसके बाद आप उस module को अलग अलग script files में import कर सकते है और उन definitions को use कर सकते है।

इसके अलावा एक module की definitions को python interpreter के interactive mode में भी use किया जा सकता है।

Modules को use करने से आपको same definitions को अलग अलग scripting files में बार बार लिखने की आवश्यकता नहीं होती है।

Can Import Other Modules

एक module को कई दूसरे modules में भी import किया जा सकता है। इस प्रकार जिस module ने दूसरे module को import किया है उसे अपनी script file में import करके आप दोनों modules की definitions को एक साथ use कर सकते है।

एक module file को आप कुछ भी नाम दे सकते है। लेकिन Module file का extension .py ही होना चाहिए।

Global Variable _name_

सभी module files के साथ python द्वारा आपको by default _name_ global variable available होता है। यह variable module का नाम store करता है। इस variable के माध्यम से किसी module का नाम access किया जा सकता है।

Can Define Executable Statements Also

एक module के अंदर definitions के अलावा executable statements जैसे की function call आदि भी define किये जा सकते है। ये statements module को import किये जाने पर automatically execute हो जाते है।

Definition Names Never Conflict

अलग अलग modules में define की गयी same नाम की definitions एक दूसरे से conflict नहीं करती है। यह modules को use करने का बहुत ही महत्वपूर्ण फायदा है।

यदि आप एक से अधिक modules को import करते है तो उनकी definitions के बीच conflict के बारे में आपको चिंता करने की आवश्यकता नहीं है।

Variants of import Statement

एक import statement के माध्यम से ही आप अपनी script के अंदर modules को import करते है। लेकिन यह आवश्यक नहीं की आप हमेशा संपूर्ण module को ही import करना चाहते है ऐसा भी हो सकता है की आप सिर्फ module के किसी particular part को ही import करना चाहते है।

ऐसी ही कुछ situations को ध्यान में रखते हुए python में import statement के कई variants provide किये गए है।

एक module को directly import statement से import किया जा सकता है।

```
import <module-name>
```

लेकिन ऐसा करने से उसकी defintions को access करने के लिए आपको उस module के नाम की आवश्यकता होती है।

आप चाहे तो module की definitions को directly import कर सकते है ऐसा करने से उन defintions को use करने के लिए आपको module के नाम की आवश्यकता नहीं होगी।

उदाहरण के लिए आप किसी एक ही function को module से import करना चाहते है तो ऐसा आप इस प्रकार कर सकते है।

```
from <module-name> import <function-name>
```

यदि आप एक से अधिक functions या variables को access करना चाहते है तो इसके लिए आप उन्हें comma से separate करके लिखते है।

यदि आप एक module की सभी definitions को python script में import करना चाहते है तो ऐसा आप इस प्रकार कर सकते है।

```
from <module-name> import *
```

Import statement के बाद define किया गया * operator यह बताता है की आप module की सभी definitions को import करना चाहते है।

जब आप from modifier का उपयोग करते हुए module की definitions को access करते है तो उन्हें आप बिना module के नाम के use कर सकते है।

Example of Python Modules

myModule.py

```
def printMessage(name)
{
print('Hello',name)
}
```

Interpreter Interactive Mode

```
>>>import myModule
>>>myModule.printMessage(Javahindi.com)
Hello Javahindi.com
```

Python in Hindi – Packages

March 30, 2019November 2, 2018 by Javahindi.com

- Introduction to **python packages in Hindi**
- Creating *python packages in Hindi*
- Importing python packages in Hindi

Introduction to Python Packages

जब आप किसी organization में कार्य करते है तो अलग अलग programmers द्वारा create किये गए modules को use करते है। ऐसे में यह संभव है की कुछ programmers के module names आपस में conflict हो जाये।

Same Module Name Conflicts With Each Other

Name conflict तब होता है जब एक ही नाम के दो modules exist करते है। यह situation critical होती है क्योंकि एक ही नाम के दो modules को आप program में use नहीं कर सकते है। क्योंकि इससे interpreter confuse हो जाता है की आप किस module को import करना चाहते है।

इस situation को handle करने के लिए python में packages को introduce किया गया है।

Way of Structuring Python's Module Namespace

Python में packages modules को structure करने का एक तरीका है। एक package एक directory (folder) होती है जिसमे आप modules को और दूसरे sub packages को store करते है।

इस प्रकार यदि आप same नाम के दो modules दो अलग अलग package directories में store करते है तो same नाम के होने पर भी वे एक दूसरे से conflict नहीं करते है।

आप अलग अलग विषयों से सम्बंधित modules को अलग अलग packages (directories) में store करके अपने सभी modules को एक निश्चित structure में organize कर सकते है।

__init__.py File

हर python package में एक __init__.py file होती है। Python interpreter द्वारा किसी directory को package के रूप में identify करवाने के लिए उस directory में इस file का होना अनिवार्य होता है।

आप इस file को empty भी create कर सकते है। इसके अलावा इस file में package का initialization code भी हो सकता है।

Importing Python Packages

Modules की ही तरह आप python packages को भी import करते है हालांकि packages में available modules को import करने का तरीका थोड़ा अलग होता है।

Uses Dotted Module Names

Packages में available modules को access करने के लिए dotted module names का प्रयोग किया जाता है। यानी की package और उसके अंदर define किये गए modules को बताने के लिए पहले package का नाम लिखा जाता है इसके बाद उस module का नाम लिखा जाता है।

Dotted module names से interpreter को पता चलता है की आप किस package से कौनसा module access करना चाहते है।

Import Single Module

```
<import> <packageName> <.> <moduleName>
```

जैसे की यदि आप myPackage में stored myModule को access करना चाहते है तो वह इस प्रकार होगा।

```
import packageName.moduleName
```

Import Subpackage Module

```
<import> <packageName> <.> <subPacakgeName> <.>
<moduleName>
```

```
import myPackage.mySubPacage.myModule
```

Import Sub-Module Using from Keyword (Alternative Way)

```
<from> <packageName> <.> <moduleName> <import> <subModule>
```

```
from myPaccakge.myModule import mySubModule
```

Import Desired Function or Variable From Module

```
from myPackage.myModule import myFunction
```

Importing Complete Package (All Sub Modules)

एक पुरे package को import करने के लिए निचे दिया गया तरीका use किया जा सकता है।

```
from myPackage.myModule import *
```

यह statement myModule के सभी sub modules और definitions को import करता है। लेकिन यह एक सही approach नहीं है। इस प्रकार import करने पर आप अनचाहे submodules और definitions को import कर सकते है।

इस situation को handle करने के लिए python में __all__ variable को introduce किया गया है। यह variable __init__.py file में set किया जाता है।

इस variable के द्वारा आप उन submodules की index provide करते है, जिन्हें आप सम्पूर्ण module के import किये जाने पर import करना चाहते है।

```
__all__ = [“module-1”,”module-2”,”module-3”]
```

इस variable को set करने के लिए angular brackets के बीच comma से separate करके आप modules को define करते है।

Python in Hindi – Functions

March 30, 2019November 5, 2018 by Javahindi.com

- Introduction to **python functions in Hindi**
- Local Symbol table of *python functions in Hindi*
- Arguments to python functions in Hindi

Introduction to Python Functions

हालाँकि python में functions create करना दूसरी ज्यादातर programming languages से अलग है, लेकिन यदि python आपकी first programming language नहीं है और आप पहले किसी दूसरी programming language में programming कर चुके है तो आप python functions को आसानी से define और use कर पाएंगे।

Block of Statements

यदि python आपकी first programming language है और आप functions के बारे में पहली बार पढ़ रहे है तो में आपको बताना चाहता है हूँ की एक function किसी program के अंदर कई statements का एक block होता है।

ये statements मिलकर किसी specific task को perform करते है। Function block का एक नाम होता है जो की पुरे program में unique होता है। इसी नाम के द्वारा इस function में लिखे गए सभी statements को एक साथ execute किया जाता है।

No Need to Redefine a Code

यदि आपके program में कोई ऐसा code है जिसे आप बार बार program में कई जगह पर execute करना चाहते है तो इसके लिए आप उस code को function में define कर सकते है। इसके बाद आप उस function को उसके नाम के द्वारा कई जगह पर call (execute) करवा सकते है।

Functions को use करने से आपको उस code को बार बार लिखने की जरुरत नहीं होती है। इससे program की redability बढ़ती है, आपका time बचता है और memory space भी बचता है।

Defined Using def Keyword

Python में function def keyword द्वारा define किया जाता है।

```
<def> <function-name>(parameters-list) :
   <function body>
```

Python में def keyword के बाद function का नाम define किया जाता है। Function के नाम के बाद parameters की list define की जाती है।

Function Body with Indentation

Next line से function की body शुरू होती है। जैसा की आपको पता है की python में spaces का अलग महत्व है। Function की body को शुरू करने से पूर्व tab-space (indentation) आता है। यह space बताता है की वह statement function की body में आता है।

Function को define करने के बाद आप जितने भी statements function body में define करना चाहते है उनको indentation के साथ लिखते है। जब आप कोई statement बिना indentation के लिखते है तो वँही पर function का end मान लिया जाता है।

Function Documentation String or Docstring

Python functions में आप documentation string define कर सकते है। यह documentation string function के बारे में जानकारी देती है। इसे docstring भी कहा जाता है।

यह function की पहली line में define की जाती है। Docstring को define करना optional होता है।

```
def myFunction():
     a normal function as usual.       //Docstring
```

यदि आप multiline docstring define करना चाहते है तो उसे आप triple quotes में define करते है।

```
“”” This is
a multiline docstring”””
```

Returning Value

यदि आप python function में से कोई value return नहीं करते है तो ऐसे में python द्वारा automatically None string return की जाती है। इसे आप print() में ऐसे function को execute करके देख सकते है जो कुछ भी value return नहीं करता है।

Local Symbol Table of Python Functions

जब एक function को execute होता है तो python द्वारा automatically एक symbol table create की जाती है। इस symbol table को local symbol table कहा जाता है।

Stores Local Variables of a Function

Local symbol table function में define किये गए local variables को store करती है। Variables names और उनको assign की गयी values इस table में store की जाती है।

Function को pass किये जाने वाले arguments local symbol table में सिर्फ तब ही store किये जाते है जब function call होता है। Python functions में arguments call by value method से pass किये जाते है और वे हमेशा एक object reference होते है।

यदि कोई function किसी दूसरे function को call करता है तो उस situation में एक नयी symbol table create की जाती है।

Function Body Represented Using Function Name

Local symbol table में function की body function के नाम के द्वारा represent की जाती है। यानी की किसी function की body को उसके नाम की value के रूप में store किया जाता है।

Python में function body का भी एक certain type होता है जो सिर्फ interpreter द्वारा identify किया जाता है। जब local symbol में function की body function के नाम की value के रूप में store की जाती है तो python interpreter उसे identify कर लेता है।

इसके बाद उस value को (function-body) को किसी दूसरे नाम (function name) को भी assign किया जा सकता है। ऐसा होने पर वह function भी original function की तरह ही कार्य करेगा।

आसान शब्दों में कहा जाये तो python में function body एक value के रूप में treat की जाती है और उसे किसी भी दूसरे function name को assign किया जा सकता है।

Python Function Arguments

Arguments वे values होती है जो function को call करते समय pass की जाती है। Python में आप मुख्यतः दो प्रकार के arguments use कर सकते है।

Default Arguments

Python आपको parameters की default value set करने की ability provide करती है। आप किसी parameter की value को एक default value provide करते है और वह value तब use की जाती है जब function को call करते समय उस parameter के लिए कोई value नहीं pass की जाती है।

Default arguments parameters को assignment operator द्वारा set किये जाते है।

```
def hello_Func(name='User'):
     print('hello'+name)
```

ऊपर दिए गए function को call करते समय यदि कोई argument नहीं pass किया जाता है तो default argument use किया जाता है।

Keyword Arguments

Python में एक special type के arguments और introduce किये गए है जिन्हें keyword arguments कहा जाता है। Python आपको key/value pair के रूप में arguments pass करने की ability provide करती है।

इसमें parameter को key के रूप में और argument को उसकी value के रूप में pass किया जाता है।

```
def myFunction(name):
   print('hello'+name)

myFunction(name='Javahindi.com')
```

Python in Hindi – Loops

March 22, 2019November 7, 2018 by Javahindi.com

- Introduction to **python loops in Hindi**
- Different *python loops in Hindi*
- Python loop control statements in Hindi

Introduction to Python Loops

कई बार कोई ऐसा code होता है जिसे आपको बार बार execute करने की आवश्यकता होती है। यदि यह कार्य manually किया जाये तो इसे करने के लिए आपको उस code को बार बार लिखना होगा। इसमें programmer का बहुत अधिक समय और effort लगता है।

मान लीजिये आप एक function को 100 बार execute करवाना चाहते है। इसके लिए आपको उस function को 100 लिखना (call करना) होगा। आप परेशान हो जायेंगे और आपका बहुत अधिक समय ख़राब हो जायेगा।

इस समस्या के समाधान के लिए loops का प्रयोग किया जाता है।

Execute Code Multiple Times

सभी programming languages की तरह python में भी आपको loops की facility दी गयी है। Loops के माध्यम से आप किसी code को multiple times execute करवा सकते है।

जिन statements को आप loop के द्वारा execute करवाना चाहते है उन्हें loop block में लिखा जाता है। यह एक function block की तरह ही होता है।

Condition to Start And End A Loop

किसी loop को start और end करने के लिए condition का प्रयोग किया जाता है। Normally जब तक कोई condition true होती है तब तक वह loop execute होता रहता है और जब वह condition false हो जाती है तो loop terminate हो जाता है।

यदि loop को terminate करने के लिए condition का प्रयोग नहीं किया जाये तो loop infinite time तक चल सकता है। ऐसे में आपकी application computer की संपूर्ण memory को खपा सकती है। ऐसी situations से बचने के लिए ही condition का प्रयोग किया जाता है।

Loop Control Variable

Loop control variable एक ऐसा variable होता है जो loop की condition को control करता है। इसे loop से पहले ही define किया जाता है। इस variable को condition में use किया जाता है।

इस variable को हर iteration में increase या decrease किया जाता है ताकि loop terminate होने की condition को match किया जा सके।

Python Loops

Python आपको 3 प्रकार के loops provide करती है।

1. while
2. for
3. nested

Python While Loop

While सबसे simple loop होता है। इसे बहुत ही आसानी से define किया जाता है।

```
while condition
   statements to be executed here…
```

While loop को while keyword द्वारा define किया जाता है। इसके बाद वह condition होती है जिसके आधार पर loop execute होता है। While loop तब तक चलता है जब तक define की गयी condition false नहीं हो जाती है।

```
num=0



while num<5
    print(“Hello”)
    num=num+1
```

ऊपर दिए गए उदाहरण में num एक loop control variable है। जब तक num 5 से कम होगा तब तक loop execute होता रहेगा और Hello print होता रहेगा। जैसे ही num की value 5 या 5 से अधिक होती है तो loop terminate हो जाता है।

Python For Loop

Python में for loop का प्रयोग सिर्फ किसी sequence को iterate करने के लिए किया जाता है। जैसे की list, tuples, dictionaries और set आदि sequences को आप इस loop द्वारा iterate कर सकते है।

Python में for loop दूसरी programming languages से अलग है। Python में for loop एक iterator की तरह कार्य करता है।

```
for <variableName> in <sequenceName>
```

```
    statements here…
```

For loop को for keyword द्वारा define किया जाता है। इसके बाद वह variable define किया जाता है जिसके द्वारा sequence के members को iterate किया जायेगा। इसके बाद in operator define किया जाता है और उसके बाद उस sequence का नाम आता है जिसे आप iterate करना चाहते है।

```
myList = [“Red”,”Green”,”Blue”]



for x in myList:
{
    print(x)
}
```

ऊपर दिए गए उदाहरण में for loop द्वारा myList को iterate किया गया है।

Python Nested Loop

ज्यादातर programming languages की तरह python में भी आप एक loop के अंदर दूसरा loop define कर सकते है। उदाहरण के लिए एक while loop के अंदर दूसरा while loop define किया जा सकता है।

Python in Hindi – Decision Making

March 30, 2019November 9, 2018 by Javahindi.com

- Introduction to **python decision making in Hindi**
- Different python decision making statements in Hindi

Introduction to Python Decision Making

जब आप एक certain code को किसी condition के आधार पर execute करना चाहते है तो इसके लिए आप decision making apply करते है।

Python में आपको decision making के लिए कुछ statements provide किये गए है जिनके आधार पर आप define कर सकते है की किसी code को कब execute करना है और कब execute नहीं करना है। इन statements को decision making statements कहा जाता है।

1. if Statement
2. if-else Statement
3. nested if Statement

Python If Statement

If में define किये गए statements केवल तब ही execute होते है जब condition true होती है। If statement define करने के लिए if keyword का प्रयोग किया जाता है।

```
if condition:
    #statements to be executed when above condition is
true…
```

इसके बाद condition define की जाती है। यह condition basically एक expression होता है और evaluate किये जाने पर इसका result या तो true या false होता है।

यदि expression का result true होता है केवल तब ही if block के statements execute होते है। Condition के false होने पर if block को skip कर दिया जाता है और उसका कोई भी statement execute नहीं होता है।

```
if 5>3:
    print(“5 is greater than 3”)
```

Python If-Else Statement

If-else statement को आप if statement का advanced version समझ सकते है। If statement के द्वारा सिर्फ condition true होने की situation को ही handle किया जाता है।

लेकिन if-else statement द्वारा आप condition true or false दोनों प्रकार की situation को control कर सकते है।

If else statement को if और else keywords के माध्यम से define किया जाता है।

```
if condition:
    #statements to be executed when condition is true
else:
    #statements to be executed when condition is false
```

If block के statement तब execute किये जाते है जब condition true होती है और else block के statement तब execute किये जाते है जब condition false होती है।

```
if 10 < 5:
    print(“10 is less than 5”)
else:
    print(“10 is greater than 5”)
```

Python elif Statement

If-else statement सिर्फ condition true और false होने की situation में ही कार्य करता है। यदि आप multiple conditions को check करना चाहते है तो इसके लिए आपको elif statement use करना चाहिए।

यह statement if else की तरह ही define किया जाता है। बस और conditions define करने के लिए if और else के बीच elif block भी add कर दिया जाता है।

यह statement if, elif और else keywords के द्वारा कार्य करता है।

```
if condition:
    #statements to be executed when if condition is true
elif condition:
    #statements to be executed when elif condition is true
else:
    #statement to be executed when all the above conditions
are false
```

If और else के बीच आप कितने भी elif blocks add कर सकते है। यदि if condition true होती है तो if block के statements execute हो जाते है।

If condition false होने पर elif condition evaluate होती है। यदि यह condition true होती है तो elif block के statements execute हो जाते है नहीं तो else block के statements execute हो जाते है।

```
if 10 < 9:
    print(“9 is greater than 10”)
elif 9 < 10:
    print(“9 is smaller than 10”)
else:
    print(“In appropriate condition.”)
```

Python Nested If Statement

जब एक if block के अंदर दूसरा if block define किया जाता है तो वह nested if statement कहा जाता है।if 10 > 3: if 10 > 6: print(“10 is greater than 3 and 6”)

Python in Hindi – Classes & Objects

March 30, 2019November 12, 2018 by Javahindi.com

- Introduction to **python classes in Hindi**
- Creating python class objects in Hindi

Introduction to Python Classes

Python एक object oriented programming है। Object oriented programming का सबसे important feature class होती है।

Bounds Data and Its Functionality Together

Classes data और उनकी functionality को bound करने का एक माध्यम होता है। यदि python की बात की जाए तो data को class variables द्वारा represent किया जाता है जिन्हें attributes भी कहते है।

Functionality को functions द्वारा represent किया जाता है। जँहा attributes एक class instance की state को maintain करते है वँही methods का प्रयोग instances की states को modify करने के लिए किया जाता है।

Class Creates a New Type

एक class create करके आप एक नया type create करते है जिसके instances create किये जाते है। Instances के द्वारा data और उसकी functionality को handle किया जाता है।

एक class एक blueprint होती है और इसके आधार पर कितने भी instances create किये जा सकते है। सभी instances के attributes एक समान होते है लेकिन उनमें store की जाने वाली information अलग अलग होती है।

Python Adds Classes Using Minimum New Syntax

Python में classes add करना दूसरी programming languages से ज्यादा अलग नहीं है। इसके लिए आपको कुछ नया syntax use करने की आवश्यकता नहीं है। Python में classes add करने के लिए आपको बहुत कम syntax use करना पड़ता है।

Combination of C++ and Modula 3 Class Mechanisms

Python में classes C++ और Modula 3 के class mechanism का combination है। C++ की ही तरह python में भी सभी class members by default public होते है और सभी member functions virtual होते है।

Objects Can Have As Much Data of As Many Types

Python में एक object कितना भी amount का data store कर सकता है। Data की size और उसके type के लिए python में कोई limit नहीं है।

Creating Python Classes

जैसा की पहले बताया गया है python में class create करने के लिए कुछ नया syntax नहीं use किया जाता है। यह कार्य python कम से कम syntax के द्वारा perform करती है।

class Keyword

ज्यादातर languages की ही तरह python में भी एक class define करने के लिए class keyword का प्रयोग किया जाता है। मुख्य अंतर यह है की python में class header declaration के बाद colon (:) लगाना अनिवार्य होता है।

```
class <class-name>:
    #class attributes (variable) & class members
```

Python में classes को आप if block या function block के अंदर भी define कर सकते है।

Python Empty Class

Python में आप एक empty class भी create कर सकते है। इसके लिए class define करके उसके अंदर pass statement declare किया जाता है जो यह बताता है की वह एक empty class है।

```
class myClass:
    pass
```

Python Class Attributes

एक class में आप attributes (variables) define कर सकते है।

```
class Apple:
    color = “red”
```

जब आप class का object create कर लेते है तो उसके बाद उस object के द्वारा attributes को access किया जा सकता है। इसके लिए dot (.) operator का प्रयोग किया जाता है।

```
obj = Apple()                #creating object of class
Apple()
print(Apple.color)       #prints red
```

Python Class Methods

Attributes के अलावा python classes में methods भी define किये जा सकते है। Methods usually attributes पर operations perform करने के लिए define किये जाते है।

Normal functions की ही तरह class में methods create करने के लिए def keyword का प्रयोग किया जाता है।

```
class Apple:
    color = “Red”



    def printColor(self)
        print(self.name)
```

Methods को भी attributes की तरह ही class के object के साथ dot operator use करके call किया जा सकता है।

```
Apple.printColor()      #prints Red
```

self Keyword

Python में self keyword का प्रयोग class methods के अंदर से attributes को refer करने के लिए किया जाता है। जैसा की ऊपर दिए गए उदाहरण में दिया गया है।

Self keyword को use करने के लिए यह आवश्यक है की method को define करते समय self keyword को parameter के रूप में define किया जाये।

यदि आप self keyword के अलावा दूसरे parameters define करते है तो method को call करते समय उनके लिए arguments pass करते है।

__init__ Method

Class attributes के लिए values run time पर भी pass की जा सकती है। ऐसी values को apply करने के लिए __init__ method का प्रयोग किया जाता है।

```
class myClass:
    def __init__(self,name):
        self.myName = name
```

```
    def printName(self):
        print(self.myName)

obj = myClass("Javahindi.com")
obj.printName()      #prints Javahindi.com
```

Python in Hindi – Date & Time

March 30, 2019November 14, 2018 by Javahindi.com

- Introduction to **python date & time in Hindi**
- python date & time function in Hindi

Introduction to Python Date & Time

कई popular programming language जैसे की java आदि primitive types (int, float, char आदि) की तरह date type भी provide करती है। इससे dates और time को handle करने में आसानी होती है।

Date is Not a Data Type

हालाँकि python भी date और time को represent करने के लिए अच्छा mechanism provide करती है लेकिन python में date एक data type नहीं है। Python में आप date objects create कर सकते है।

Provides datetime Module

Python आपको datetime module provide करती है। इस module को import करके आप date और time के साथ कार्य कर सकते है।

इस module में datetime class define की गयी है। इस class के constructor को call करके आप date objects create करते है।

Datetime class ऐसे बहुत से methods provide करती है जिनके द्वारा आप किसी date object से date और time सम्बंधित जानकारी प्राप्त कर सकते है।

Datetime class के सभी methods मुख्यतः datetime class के साथ dot operator के प्रयोग से access किये जाते है। हालाँकि आप datetime class का object create करके उसके द्वारा भी इन methods को access कर सकते है।

उदाहरण के लिए यदि आप current date प्राप्त करना चाहते है तो इसके लिए now() method इस प्रकार प्रयोग कर सकते है।

```
import datetime    #Importing datetime module


Date = datetime.datetime.now()     #Storing current date in
Date variable

print(Date)   #printing current date
```

ऊपर दिए गए उदाहरण में now() method को call करने के लिए पहले datetime module उसके बाद datetime class को access किया गया है।

datetime Class

जैसा की मैने पहले बताया date object create करने के लिए आप datetime class का constructor call करते है। इस constructor में initially year, month और day pass किया जाना अनिवार्य होता है।

```
import datetime


obj = datetime.datetime(2018, 11, 13)  #calling datetime
class constructor

print(obj)
```

यह constructor time के लिए भी arguments लेता है। लेकिन ऐसा करना अनिवार्य नहीं है। यह optional है।

```
datetime(year, month, day, hour, minute, second, microsecond)
```

Time arguments के लिए यदि कोई value pass नहीं की जाती है तो default zero value use की जाती है।

जब आप date ओर time दोनो को एक ही object द्वारा प्रदर्शित करना चाहते है तो ऐसे में आपको datetime class के द्वारा ही object create करना चाहिए। यदि आप इन date और time की information को अलग अलग प्रदर्शित करना चाहते है तो उसके लिए date और time classes द्वारा objects create करने चाहिए।

Python में datetime class date और time class के सभी attributes और methods को provide करती है।

date Class

Datetime class के अलावा datetime module में और भी classes available है जैसे की time, date, timedelta आदि। यदि आप सिर्फ date object create करना चाहते है या फिर date information को ही represent करना चाहते है तो इसके लिए आप datetime class के बजाय date class के constructor को भी call कर सकते है।

```
datetime.date(year, month, day)
```

Current date प्राप्त करने के लिए आप today() method को call कर सकते है।

- date.today()

- date.fromtimestamp(timestamp)
- date.fromordinal(ordinal)

Month प्राप्त करने के लिए month attribute को use किया जा सकता सकता है।

- date.year
- date.month
- date.day
- date.min
- date.max
- date.resolution

time Class

यदि आप object द्वारा सिर्फ time को प्रदर्शित करना चाहते है तो इसके लिए time class के constructor को call कर सकते है।

```
datetime.time(hour, minute, second, microsecond, timezone-info)
```

यह class date class की ही तरह min, max और resolution attributes provide करती है।

- time.hour
- time.minute
- time.second
- time.microsecond
- time.tzinfo

किसी object के current time को दूसरे time से replace करने के लिए आप time class का replace() method use कर सकते है।

- time.replace(hour, minute, second, microsecond, tzinfo)
- time.isoformat()
- time.__str__()

- time.strftime(format)
- time.__format__()
- time.utcoffset()
- time.dst()
- time.tzname()

timedelta Class

जब आप time duration को represent करना चाहते है तो इसके लिए timedelta class का object create कर सकते है। Time duration एक date से लेकर दूसरी date तक के बीच का difference होती है।

```
datetime.timedelta(days, seconds, microseconds,
milliseconds, minutes, hours, weeks)
```

strftime() Method

Python आपको date objects को readable string format में convert करने के लिए strftime() method provide करती है। इस method को string format time method कहा जाता है।

यह method argument के रूप में format लेता है।

```
datetime.strftime(format)
```

इस format द्वारा ही determine किया जाता है की किस string format में date को show किया जायेगा।

- %A – Weekday का नाम।
- %w – Weekday का Number
- %B – Month का नाम।
- %Y – year
- %P – AM या PM

- %H – Hour
- %M – Minute
- %S – Second
- %Z – Time zone name

Python in Hindi – Errors & Exceptions

March 30, 2019November 16, 2018 by Javahindi.com

- Introduction to **python errors & exceptions in Hindi**
- *Handling python exceptions in Hindi*
- Python user defined exceptions in Hindi

Introduction to Python Errors & Exceptions

जब आप कोई application create करते है और उसके code को execute करने का प्रयास करते है तो आपको कई बार errors का सामना करना पड़ता है। Mostly errors या तो program को execute होने नहीं देती (compile-time) या फिर execute होते हुए program को रोक (run-time) देती है।

Python code को execute करते समय दो प्रकार की errors से आपका सामना हो सकता है।

- Syntax Errors / Parsing Errors
- Exceptions/ Run Time Errors

Syntax Errors or Parsing Errors

Syntax errors basically गलत syntax लिखने की वजह से generate होती है। इन्हे आप grammatical errors कह सकते है। उदाहरण के लिए यदि function

को define करते समय आप colon (:) ना लगाए और code को execute करने का प्रयास करे तो ऐसे में parser syntax-error generate करता है।

Syntax errors program को parse या compile करते समय ही parser द्वारा identify कर ली जाती है। इन errors को गलत syntax हटाकर और सही syntax apply करके remove किया जाता है।

जब कोई syntax error आती है तो parser message show करता है। उस message में parser उस complete line को display करता है जिसमें error आयी है। इसके अलावा parser उस line में up arrow (^) के माध्यम से वह जगह भी show करता है जँहा पर error detect हुई है।

ज्यादातर error parser द्वारा दिखाई गयी up arrow के बाद आती है। इसके अलावा parser द्वारा file name और line number भी show किये जाते है ताकि आप उस code को आसानी से ढूँढ सके जँहा पर error आयी है।

Exceptions or Run Time Errors

Exceptions वे errors होती है जो code execution के दौरान आती है। इन्हें run time errors भी कहा जाता है। किसी भी application के लिए exceptions बहुत घातक होती है। क्योंकि ये चलते हुए program को रोक देती है। इससे user experience ख़राब होता है और महत्वपूर्ण काम बीच में अटक सकते है।

जब एक exception generate होती है तो parser एक message display करता है। इस message में parser exception का type और वह location display करता है जँहा से संभवतः exception generate हुई है।

Exception type असल में exception का नाम होता है जो briefly यह बताता है की कौनसी exception generate हुई है। Exception की location stack traceback के रूप में दिखाई जाती है ताकि आप वह exact location track कर सके जँहा पर exception generate हुई है।

Handling Exceptions in Python

Syntax error की तरह आप exception को ठीक नहीं कर सकते है और generate होने से भी नहीं रोक सकते है। हाँ लेकिन आप generate होने वाली exceptions को handle अवश्य कर सकते है।

Python आपको exception को handle करने का mechanism provide करती है। यह mechanism आपको exception generate होने पर application के बुरी तरह fail होने की बजाय एक organized way में work करते हुए situation को handle करने की ability provide करता है।

Python में exception handling mechanism कुछ keywords द्वारा कार्य करता है।

- try
- except
- else
- raise
- finally

इन सभी keywords के द्वारा अलग अलग blocks define किये जाते है जो अलग अलग situations में execute होते है।

try Block

Try keyword के द्वारा try block define किया जाता है। Try block में उस code को रखा जाता है जो संभवतः exception generate कर सकता है।

```
try:
        #statements that may generate exception
```

यदि try block में कोई भी exception generate नहीं होती है तो except blocks का कोई भी statement execute नहीं होता है और सभी except blocks को skip कर दिया जाता है।

यदि try block में exception genrate होती है और उस exception से सम्बंधित except block define किया होता है तो उस except block के statements execute कर दिए जाते है।

यदि exception के लिए कोई भी except block नहीं मिलता है तो application रुक जाती है और parser द्वारा exception message display कर दिया जाता है।

Except block द्वारा exception को handle किये जाने के बाद execution आगे बढ़ता है।

except Block

Try block के तुरंत बाद except keyword द्वारा except block define किया जाता है। Except block को define करते समय except keyword के बाद exception type define किया जाता है।

यदि except block से पूर्व define किये गए try block में except block में define की गयी type की exception आती है तो except block में लिखे गए सभी statements execute हो जाते है।

एक try block के बाद multiple except blocks define किये जा सकते है। आप जिस भी type की exception को handle करना चाहते है उसके लिए अलग से except block define करते है। Try block में जिस type की exception generate होती है उसी से सम्बंधित except block execute किया जाता है और बाकि skip हो जाते है।

```
try:
```

```
    #statements that may generate exception
except <exception-Type>:
    #statements to be executed when <exception-Type> occurs
in above try block
except <exception-Type>:
    #statements to be executed when <exception-Type> occurs
in above try block
```

एक except block द्वारा multiple exceptions का नाम भी define किया जा सकता है। इसके लिए आप उन exceptions के नामों को comma से separate करके brackets में लिखते है।

```
except(ValueError, NameError, TypeError)
```

Exception Argument

हर exception के साथ एक value associated रहती है जिसे exception argument कहा जाता है। इसे except block में exception के नाम के बाद define किया जाता है।

```
except valueError as exObj:
    print(exObj)
```

यह argument exception के बारे में detail provide करता है। इसे print करके भी exception की details देखी जा सकती है।

else Block

Else block को except blocks के बाद define किया जाता है। इस block में वे statements लिखते जाते जिन्हें आप try block द्वारा कोई भी exception raise नहीं किये जाने की situation में execute करना चाहते है।

```
else:
```

```
    #statements to be executed if try block does not
generate an exception
```

Else block ऐसी situations को avoid करने में मदद करता है जब except blocks उन exceptions को handle करने का प्रयास करते है जो try block में generate ही नहीं हुई है।

raise Statement

यदि आप समझते है की कोई statement या condition exception generate कर सकती है तो उस condition के लिए आप manually भी exception generate कर सकते है। इसके लिए raise keyword का प्रयोग किया जाता है।

```
raise TypeError('Type Exception Occurred!')
```

कोई भी exception raise करने के लिए आप raise keyword के बाद exception का नाम लिखते है और उसके बाद brackets में जो message आप display करना चाहते है उसे argument के रूप में pass करते है।

Python Built-in Exceptions

Built-in exceptions ऐसी exceptions होती है जो python में पहले से ही define की गयी है। जैसे की TypeError, ValueError और NameError आदि कुछ built-in exception के उदाहरण है।

Python में available सभी built-in applications की जानकारी के लिए Python Built-in Exceptions page को visit करें।

User Defined Exceptions

Python आपको स्वयं की custom exceptions define करने की ability भी provide करती है। इन्हे user defined exceptions कहा जाता है।

एक custom exception define करने के लिए आप class define करते है। इस class को custom application के रूप में treat किये जाने के लिए यह आवश्यक होता है की उसे python की Exception class से derive किया जाये।

```
class myCustomException:Exception
    //define exception here…
```

Python में Exception class सभी built in और user defined exceptions की base class होती है।

Python in Hindi – Lambda Functions

March 30, 2019November 19, 2018 by Javahindi.com

- Introduction to **python lambda functions in Hindi**
- *Syntax of python lambda functions in Hindi*
- Example of python lambda functions in Hindi

Introduction to Python Lambda Functions

Mostly सभी popular programming languages में anonymous functions का concept available है। कई programming languages में इन्हे lambda expressions भी कहा जाता है।

किसी programming languages में यह आवश्यक नहीं है की lambda function और anonymous function एक ही चीज़ हो। एक anonymous function lambda category में तब ही आता है जब वह function object return करता है और उसे argument के रूप में pass किया जा सकता है।

Python में lambda function और anonymous function एक ही चीज़ है। इसलिए आपको इनके बीच confuse नहीं होना चाहिए।

Defined Anonymously

Lambda functions को बिना किसी नाम के anonymously define किया जाता है। इनको आप एक object को assign करते है और उसी object नाम के द्वारा इन्हे use किया जाता है। यह कई keywords का use करते हुए एक पूर्ण function लिखने की आवश्यकता को खत्म कर देता है।

Lambda Keyword

Python में normal functions को def keyword द्वारा और lambda functions को lambda keyword द्वारा define किया जाता है।

```
lambda <arguments> : expression
```

Can Have Any Number of Arguments

Lambda keyword के बाद arguments को define किया जाता है। Normal functions की ही तरह python में lambda functions को भी आप अपनी इच्छा अनुसार कितने भी arguments pass कर सकते है।

```
lambda <x1, x2, x3, x4……xN> : <expression>
```

Can Only Have a Single Expression

Arguments के बाद colon (:) लगाकर expression define किया जाता है। इस expression को आप function की body की तरह समझ सकते है।

```
lambda <x1,x2,x3,….xN> : <single – Expression>
```

Lambda functions में केवल एक ही expression define किया जा सकता है। Lambda functions का syntax multiple expressions use किया जाना

allow नहीं करता है। यही कारण है की lambda functions को one line functions भी कहा जाता है।

No Return Statement

Normal functions की तरह lambda functions में कोई return statement नहीं होता है। Define किये गए single expression को evaluate किया जाता है और उसके result को automatically return कर दिया जाता है।

When Function Object is Required

जैसा की मैने आपको पहले ऊपर बताया lambda functions को define करके एक object को assign किया जाता है। इसके बाद उस object के द्वारा ही आप lambda function को use करते है। यह एक function object होता है।

```
obj = lambda <x1, x2, x3,…..xN> : <expression>
```

इस प्रकार आप lambda functions को जँहा भी किसी function object की आवश्यकता हो वँही पर use कर सकते है।

Single Use Function

Lambda functions single use functions होते है। इसलिए आप इन्हें वँहा पर ही create करते है जँहा पर आपको इनकी आवश्यकता हो। इन्हे throwaway functions भी कहा जाता है।

जब भी आपको एक single expression logic की आवश्यकता हो तो आप एक सम्पूर्ण function create करने की बजाय lambda function create कर सकते है।

उदाहरण के लिए आप एक object को value पहले calculate करके assign करना चाहते है। ऐसे में lambda function द्वारा आप वह value evaluate कर सकते है और इसके बाद उसे object को assign किया जा सकता है।

Can Be Passed as an Argument to Another Function

क्योंकि एक lambda function object होता है। इसलिए आप एक lambda function को दूसरे function में argument के रूप में भी pass कर सकते है। ऐसी situation में expression द्वारा return की गयी और object द्वारा hold की गयी value ही argument के रूप में काम करेगी।

```
LamObj = lambda x1, x2 : x1 + x2


def myFunction(LamObj):
    print("Sum is"+LamObj)
```

Python में lambda functions का प्रयोग अधिकतर filter(), map() और reduce() functions में argument के रूप में किया जाता है।

filter() Function

Python में filter एक ऐसा function है जो logic के आधार पर किसी list में से data को filter करके extract करने के लिए प्रयोग किया जाता है। इस function में आप एक function और एक list को argument के रूप में pass करते है।

List के सभी items के साथ logic perform किया जाता है जो item उस logic के अनुरूप होता है उसे store कर लिया जाता है। इस प्रकार वे सभी items जो उस logic को fulfil करते है उन्हें return कर दिया जाता है।

उदाहरण के लिए आप उन सब list items को list करना चाहते है जो 3 के multiple है तो ऐसे में lambda functions द्वारा ऐसा किया जा सकता है।

```
myList = (2, 3, 4, 6, 8, 9, 10 ,12)


result_List = list(filter(lambda x : x%3==0, myList))

//return 3 6 9 12
```

यँहा पर जैसा की आप महसूस कर सकते है normal function की बजाय lambda function एक शानदार fit होगा। क्योंकि आपको अलग से एक function create करने की आवश्यकता नहीं होगी। जँहा पर आपको आवश्यकता है आप वँही पर एक single line में उसे define कर लेते है।

इसी प्रकार आप map() function में argument के रूप में lambda expression को use कर सकते है और logic के आधार पर नयी list create कर सकते है।

Python in Hindi – File Handling

March 30, 2019November 21, 2018 by Javahindi.com

- Introduction to **python file handling in Hindi**
- *Python file operations in Hindi*
- Python file handling methods in Hindi

Introduction to Python File Handling

सभी popular programming languages की तरह python भी file handling के लिए support provide करती है। इसके लिए python में कई methods available है।

Python द्वारा file handling ability provide किये जाने की कई advantages है।

- आप disk पर available files को access और modify कर सकते है।
- बहुत सारा data जिसे manually insert किया जाना संभव नहीं है directly file से input के रूप में read किया जा सकता है।
- बहुत सारा data जिसे की screen पर देखा जाना संभव नहीं है आसानी से file में store किया जा सकता है और उसे बाद में analyse किया जा सकता है।

Python file handling mechanism द्वारा आप कई operations perform कर सकते है।

- Read a file
- Write to a file
- Append to a file
- Close a file

Python द्वारा disk files के साथ किसी भी प्रकार का operation (read, write आदि) perform करने से पूर्व file को memory में load किया जाना आवश्यक होता है। इसे file को open करना भी कहते है। इसके लिए python में open() function available है।

Python open() Function

Python file handling mechanism में open() ही एक function है। बाकी आप जो operation perform करेंगे वे object methods द्वारा perform करेंगे।

जब आप open() function द्वारा किसी file को open करते है तो ऐसे में यह function उस file को object के रूप में memory में load कर देता है। जिसे file object भी कहा जाता है।

एक बार जब file object के रूप में available हो जाती है तो उसके साथ automatically कई methods available हो जाते है जिन्हे आप उस object द्वारा call कर सकते है और अलग अलग operations perform कर सकते है।

Python open() एक बहुत ही simple function है। यह दो arguments लेता है।

```
obj = open('file','mode')      #Opens a file in given mode
```

पहला argument एक string होता है जो की basically उस file का नाम होता है जिसे आप open करना चाहते है। यदि file same directory में है तो आप सिर्फ उसका नाम दे सकते है। यदि file अलग directory में है तो आपको उसका पूरा path देना होगा।

दूसरा argument भी एक string ही है लेकिन यह उस mode को दर्शाता है जिस mode में आप file को open करना चाहते है। उदाहरण के लिए यदि आप एक file को सिर्फ read करना चाहते है तो आप उसे read mode में open करेंगे। यदि आप file से data read करने के साथ write भी करना चाहते है तो इसके लिए आप read/write mode में file को open करेंगे।

Modes के लिए argument pass किया जाना optional है। यदि आप argument के रूप में कोई भी mode नहीं pass करते है तो file को read only mode में open किया जाता है।

ज्यादातर files को कुछ ही common modes में open किया जाता है।

- r – Read
- w = Write
- a = Append
- r+ = Both read & write
- b = Binary mode, data byte-objects की form में read और write किया जाता है।

Opening File in Context

जब open() function द्वारा file को open करते है तो एक stream open होती है जिसके द्वारा data को read और write किया जाता है। Operations

perform करने के बाद close() method द्वारा उस stream को close किया जाना आवश्यक होता है।

यदि आप उस stream को close नहीं करते है या ऐसा करना भूल जाते है तो ऐसे में वह file काफी समय तक आपके system resources को occupy करके रखती है जो की programming की दृष्टि से सही नहीं है। आखिर में वह file object garbage collector द्वारा destroy किया जाता है।

इस situation से बचने के लिए आप file को context में open कर सकते है। इसके लिए with और as keywords का प्रयोग किया है। इससे एक block का निर्माण होता है जिसमे दिए गए statements के execute होने के बाद वह file object automatically destroy हो जाता है। यह python में file को open करने का most recommended तरीका भी है।

जब आप इस प्रकार किसी file को open करते है तो उसे close() method द्वारा manually close करना आवश्यक नहीं होता है।

Python File Object Handling Methods

जैसा की पहले बताया गया है open() function file को object के रूप में return करता है। File को object के रूप में प्राप्त कर लेने पर उस object के साथ आपको कई methods automatically available हो जाते है जिनके द्वारा आप विभिन्न operations perform कर सकते है।

read()

इस method का प्रयोग file के content को read करने के लिए किया जाता है। इस method में एक argument pass किया जा सकता है जो एक number होता है जो दर्शाता है की आप कितनी bytes उस file से read करना चाहते है। यह argument optional होता है।

```
fileObj.read(size-in-bytes)    #reads given number of bytes
```

यदि आप कोई भी argument pass नहीं करते है या फिर negative argument pass करते है तो file का पूरा content read किया जायेगा।

```
fileObj.read(-1)     #reads all content of file
```

यदि file की size memory से अधिक है तो ऐसे में problem हो सकती है इसलिए यह optional argument provide किया गया है ताकि file के कुछ content को देखा जा सके।

readline()

यह method file में से एक बार में एक ही line read करता है। Line string के रूप में read की जाती है और उसके आखिर में automatically new line character (n) add कर दिया जाता है।

```
fileObj.readline()
```

यदि कोई empty line encounter होती है तो उसे सिर्फ n द्वारा show किया जाता है और file के end और empty string द्वारा represent किया जाता है।

By default readline() method सिर्फ एक ही line return करता है। सभी lines को read करने के लिए एक for loop का प्रयोग किया जा सकता है।

```
for line in fileObj          #Iterating over fileObj file
using for loop, line represents a line in file
    print(line,end=' ')      #Printing line and defining
how to end a line
```

आप f.readlines() method द्वारा list में एक file की सभी lines को भी read कर सकते है।

write()

यह method argument के रूप में string को लेता है और उसे file में write करता है। इसके बाद यह method characters की सँख्या return करता है जो की file में write किये गए है।

```
fileObj.write('content-to-write')    #write passed given
string to file
```

tell()

यह function file object की current location return करता है। उदाहरण के लिए यदि आपने readline() method द्वारा एक line read की है तो file object की location second line में होगी।

```
fileObj.tell()        #Get current location of file object
```

यह function location को bytes के रूप में दर्शाता है जो की file की शुरआत से उस location तक की दुरी को दर्शाती है।

seek()

इस method का प्रयोग file object को file में किसी location पर ले जाने के लिए किया जाता है। उदाहरण के लिए आप किसी निश्चित location से file को read करना चाहते है या फिर आप किसी location पर data enter करना चाहते है तो ऐसे में उस location तक इस method द्वारा जाया जा सकता है।

```
fileObj.seek(offset, from_what)         #Go to specific
position in file
```

यह method दो arguments लेता है। इनमें second argument optional है। First argument उस position को दर्शाता है जँहा पर आप जाना चाहते है और second argument उस file में उस जगह (beginning, end etc.) को दर्शाता है जँहा से आप जाना चाहते है।

Beginning से location पर जाने के लिए 0, current location से जाने के लिए 1 और file के end से जाने के लिए 2 का प्रयोग किया जाता है।

उदाहरण के लिए यदि आप file की beginning से 6th byte पर जाना चाहते है तो first argument के रूप में 5 और second argument के रूप में 6 pass किया जायेगा।

Python in Hindi – Iterators

March 30, 2019November 23, 2018 by Javahindi.com

- Introduction to **python iterators in Hindi**
- *Python iterable objects in Hindi*
- Creating custom python iterator in Hindi

Introduction to Python Iterators

Iterator एक tool होता है जिसके माध्यम से कुछ items की list को जल्दी और आसानी से iterate किया जा सकता है। Iterator feature कई popular programming languages द्वारा implement और support किया गया है।

हालाँकि python में iterators को अलग तरह से implement किया जाता है लेकिन ये कार्य बाकी programming languages की तरह ही करते है।

Object That Can Be Iterated Upon

Python में iterator एक object है जिसे iterate किया जा सकता है। Object के सभी items को at least एक बार access करना iterate करना कहलाता है।

Implements Iterator Protocol

Python में किसी भी object को iterator बनाने के लिए iterator protocol को implement करने की आवश्यकता होती है। यह protocol दो methods द्वारा बनता है जो iterating process को perform करते है।

- __iter__() – Returns iterator object
- __next__() – Iterate next item

Python Iterable Objects

Python में ऐसे कुछ objects है जिनमें पहले से ही iterator protocol को built in रूप से implement किया गया है। ये objects __iter__() और __next__() methods provide करते है। ऐसे objects को iterable objects कहा जाता है।

Iterable objects के लिए आपको अलग से iterator protocol नहीं implement करना होता है। इन्हे directly iterate किया जा सकता है। Python में list, tuple, dictionary, set और strings iterable objects होते है।

List

```
myList = [“Best”, “Hindi”, “Tutorials”]
list_Iterator = iter(myList)     #Returns an iterator of
myList



print(next(list_Iterator))      #/Prints first item
print(next(list_Iterator))      #Prints second item
print(next(list_Iterator))    #Prints third item
```

Tuples

```
myTuple = (“Hello”, “Reader”)
tuple_Iterator = iter(myTuple)      #Returns an iterator of
myTuple



print(next(tuple_iterator))
print(next(tuple_iterator))
```

Strings

```
myString = “Best”
string_Iterator = iter(myString)



print(next(string_Iterator))
print(next(string_Iterator))
print(next(string_Iterator))
print(next(string_Iterator))
```

Lists और tuples की ही तरह sets और dictionaries भी iterable है, जिन्हें आप समान तरीके से iterate कर सकते है।

Iterating Using For Loop

हालाँकि iterable objects default iterator protocol के साथ आते है लेकिन आप इन्हें for loop द्वारा भी iterate कर सकते है। For loop द्वारा iterate करना बहुत आसान होता है और आपको iterator object प्राप्त करने की भी जरुरत नहीं होती है।

```
myList = [“Best”,”Hindi”,”Tutorials”]     #creating myList



for x in myList:      #Iterating all myList items using for
loop
    print(x)            #print all list items
```

इसी प्रकार दूसरे python objects को भी for loop द्वारा iterate किया जा सकता है। For loop automatically iterator object create करता है और next() method को call करता है।

Custom Python Iterator

ऐसे objects जो by default iterator protocol implement नहीं करते है उनको iterate करने के लिए आपको उनमें iterator protocol implement करने की आवश्यकता होती है।

जैसा की मैने पहले बताया किसी भी object को iterator बनाने के लिए दो methods को implement करने की आवश्यकता होती है। आइये इनके बारे में detail से जानने का प्रयास करते है।

__iter__()

यदि आपने python class में define किये जाने वाले __init__() method के बारे में पढ़ रखा है तो में आपको बताना चाहूँगा की यह method init() method के जैसा ही है।

इन दोनों methods में जो main difference है वह यह की __iter__() method से आपको iterator object को return करना अनिवार्य होता है।

__next__()

यह method next item को return करता है। यह method मुख्यतः iterator operations perform करने के लिए use किया जाता है।

Example of Python Custom Iterator

```
class myTable:
    def __iter__(self):
        self.n = 2
        return self
```

```
    def __next__(self):
        x = self.n
        self.n += 2
        return x

obj = myTable()
myIterObj = iter(obj)

print(next(myIterObj))
print(next(myIterObj))
print(next(myIterObj))
```

```
2
4
6
```

Python in Hindi – Standard Library

March 30, 2019February 8, 2019 by Javahindi.com

- Introduction to **python standard library in Hindi**
- Important modules of *python standard library in Hindi*

Introduction to Python Standard Library

Python में standard library built in modules का एक set होती है जो की core programming language का हिस्सा होते है।

सामान्य तौर पर python तीन प्रकार के modules उपयोग किये जाते है।

- स्वयं आप (programmer) के द्वारा बनाये गए।
- किसी third party (programmer/organization) द्वारा बनाये गए।
- Standard library में available

Standard library में available modules python core programming team के द्वारा बनाये जाते है। ये अलग अलग प्रकार के महत्वपूर्ण tasks perform करने के लिए उपयोग किये जाते है।

Important Modules of Python Standard Library

Python की standard library बहुत ही बड़ी है। इस library में कई बहुत से modules है जिनमें बहुत से function define किये गए है।

यँहा पर कुछ ऐसे modules के बारे में बताया जा रहा है जो एक programmer के रूप में आप regularly उपयोग करेंगे।

os

Python का os module operating system से interact करने के लिए बहुत सारे functions provide करता है।

इस module में भी open() function define किया गया है। इसलिए इस module को from keyword द्वारा नहीं import करना चाहिए। ऐसा करने पर builtin open() function जो की अलग तरह से कार्य करता है उसके साथ clash हो जाता है।

इस module में dir() और help() जैसे functions है जो programmer के उपयोगी होते है।

```
>>> import os

>>> os.getcwd() # show current working directory
```

shutil

यदि आप python द्वारा regular तरीके से operating system की files और directories manage करना चाहते है तो इसके लिए आपको os module के बजाय shutil module उपयोग करना चाहिए।

Shutil module os module की अपेक्षा आसानी से उपयोग किये जाने वाला interface provide करता है।

```
>>> import shutil

>>> shutil.copyfile('data.db','archive.db')
```

glob

यह module directory searches के दौरान files की list बनाने के लिए function provide करता है।

```
>>> import glob

>>> glob.glob(*.txt)  # find txt extension files
```

sys

यह module command line arguments को process करने में अहम् भूमिका निभाता है। Command line arguments sys module के argv attribute में list के रूप में store किये जाते है।

```
>>> import sys

>>> print(sys.argv)
```

Sys module में error output redirection और program termination के लिए भी attributes होते है।

- stdin
- stdout
- stderr
-

ये attributes warnings और error messages आदि show करने के लिए उपयोगी होती है। Program termination के लिए sys module का exit() function उपयोग किया जा सकता है।

getopt

इसके अलावा python में getopt module भी available है जो की unix conventions के आधार पर command line arguments को process करता है।

argparse

Python में argparse एक और module है जो command line arguments को process करने के लिए लिए प्रयोग किया जाता है। यह दूसरे modules से अधिक powerful और flexible है।

re

यह module string processing के लिए regular expression tools provide करता है। यदि आपको सिर्फ common task perform करने है तो ऐसा आप string methods द्वारा भी कर सकते है। यह module advanced processing के लिए होता है।

Math

Python में math module में floating point math के लिए C library functions stored है। जिन्हें आप आवश्यकतानुसार उपयोग कर सकते है।

Random

यह module random selection perform करने के लिए tools provide करता है।

Statistics

यह module statistical calculations perform करने के लिए tools provide करता है। जैसे की mean, median और variance आदि properties इस module में available functions की मदद से आप calculate कर सकते ही।

urllib.request

यह module internet के साथ कार्य करने के लिए tools provide करता है। Basically यह module आपको urls से data retrieve करने की ability provide करता है जो की web applications के लिए महत्वपूर्ण होती है।

smtplib

यह module भी internet के साथ कार्य करता है। यह module email send करने के लिए tools provide करता है। इस module में available functions का उपयोग करके आप python script द्वारा emails send कर सकते है।

datetime

यह module date और time के साथ कार्य करने के लिए classes provide करता है। इन classes में available functions से आप date और time आदि को modify कर सकते है।

zlib, gzip, bz2, lzma, zipfile & tarfile

ये सभी modules data archiving और compression के लिए tools provide करते है। हर format के लिए अलग अलग modules में tools provide किये गए है।

timeit

कई बार आपको अलग अलग codes की performance measure करने की आवश्यकता पड़ सकती है। यह तब किया जाता है जब आप सोचते है की कोई एक code की performance दूसरे से बेहतर है।

Performance measure करने के लिए python में timeit module provide किया गया है। इस module में timer आदि functions available है।

doctest

यह module किसी module को scan और validate करने के लिए tools provide करता है। Validate करने के लिए tests को string में store किया जाता है।

unitest

यह module भी doctest की ही तरह module को scan और validate करने के लिए tools provide करता है। लेकिन इसमें tests को किसी string की बजाय separate file में store किया जाता है।