

Di dalam bahasa pemrograman Python pengulangan dibagi menjadi 3 bagian, yaitu :

- **While Loop**
- **For Loop**
- **Nested Loop**

Pengulangan While

Pengulangan While Loop di dalam bahasa pemrograman Python dieksesusi statement berkali-kali selama kondisi bernilai benar atau True. Dibawah ini adalah contoh penggunaan pengulangan While Loop.

```
#Contoh penggunaan While Loop
count = 0 while (count < 9):    print ('The count is:', count)
count = count + 1 print ("Good bye!")
```

5.11 Pengulangan For

Pengulangan For pada Python memiliki kemampuan untuk mengulangi item dari urutan apapun, seperti list atau string. Dibawah ini adalah contoh penggunaan pengulangan While Loop.

```
#Contoh pengulangan for sederhana
angka = [1,2,3,4,5] for x in angka:    print(x)
#Contoh pengulangan for buah = ["nanas", "apel", "jeruk"] for makanan in buah:
print("Saya suka makan", makanan)
```

5.12 Pengulangan Bersarang (Nested Loop)

Bahasa pemrograman Python memungkinkan penggunaan satu lingkaran di dalam loop lain. Bagian berikut menunjukkan beberapa contoh untuk menggambarkan konsep tersebut. Dibawah ini adalah contoh penggunaan Nested Loop.

```
#Contoh penggunaan Nested Loop
i      = 2 while(i < 100):
j      = 2   while(j <= (i/j)):
if not(i%j): break      j = j + 1
if (j > i/j) : print i, " is prime"  i = i + 1 print "Good bye!"
```

5.13 Number Python

Number adalah tipe data Python yang menyimpan nilai numerik. Number adalah tipe data yang tidak berubah. Ini berarti, mengubah nilai dari sejumlah tipe data akan menghasilkan objek yang baru dialokasikan. Objek Number dibuat saat Anda memberikan nilai pada-nya. Sebagai contoh :

```
angkaPertama = 1
```

```
angkaKedua = 33
```

Python mendukung beberapa tipe data Number diantaranya :

- Int
- Float
- Complex

Berikut ini adalah beberapa contoh dari Tipe data Number pada Python :

Int	Float	Complex
20	0.1	3.14 j
300	1.20	35 .j
-13	-41.2	3.12e-12 j
020	32.23+e123	.873 j
-0103	-92.	-.123+0 J
Int	Float	Complex
-0x212	-32.52e10	3 e+123J
0x56	60.2-E13	4.31e-4 j

Konversi Tipe Data Number Python

Pada Python Anda bisa mengkonversi tipe data dengan menggunakan fungsi. Dibawah ini adalah beberapa fungsi untuk mengkonversi tipe data number Python.

- int(x): untuk meng-konversi x menjadi plain integer.
- long(x): untuk meng-konversi x menjadi long integer.
- float(x) untuk meng-konversi x menjadi floating point number.
- complex(x): untuk meng-konversi x menjadi complex number dengan real part x dan imaginary part zero.

- `complex(x, y)` untuk meng-konversi x dan y menjadi complex number dengan real part x dan imaginary part y. x dan numeric expressions y.

Fungsi Matematika

Pada bahasa pemrograman Python terdapat fungsi untuk melakukan perhitungan matematis, berikut adalah daftarnya :

Nama	Penggunaan	Penjelasan
Absolute	abs(x)	Nilai absolut dari x:(positive) jarak antara x and 0.
Ceiling	ceil(x)	Ceiling dari x: integer terkecil yang kurang dari x.
Cmp	cmp(x, y)	-1 if $x < y$, 0 if $x == y$, or 1 if $x > y$. Tidak berlaku lagi dengan Python 3. Sebaliknya gunakan return $(x>y)-(x$
Eksponen	exp(x)	Nilai eksponen dari x: e^x
Nama	Penggunaan	Penjelasan
Fabs	fabs(x)	Nilai absolut dari x.
Floor	floor(x)	Nilai dasar dari x: internet terbesar tidak lebih besar dari x.
Log	log(x)	Logaritma dari x, untuk $x > 0$.
Log 10	log10(x)	Basis 10 logaritma dari x, untuk $x > 0$.
Max	max(x1, x2,...)	Argumen terbesar: Nilai terdekat dengan tak terhingga positif
Min	min(x1, x2,...)	Argumen terkecil: nilai yang paling mendekati tak berhingga negatif.
Modf	modf(x)	Bagian pecahan dan bilangan bulat dari x dalam tupel dua item. Kedua bagian memiliki tanda yang sama dengan x. Bagian integer dikembalikan sebagai float.
Pow	pow(x, y)	Nilai $x^{**} y$.
Round	round(x [,n])	X dibulatkan menjadi n digit dari titik desimal. Putaran Python jauh dari nol sebagai tie-breaker: round (0.5) adalah 1.0 dan round (-0.5) adalah -1.0.
Akar Kuadrat	sqrt(x)	Akar kuadrat x untuk $x > 0$.

Fungsi Nomor Acak

Nomor acak digunakan untuk aplikasi permainan, simulasi, pengujian, keamanan, dan privasi. Python mencakup fungsi berikut yang umum digunakan. Berikut adalah daftarnya :

Nama	Penggunaan	Penjelasan
Choice	choice(seq)	Item acak dari list, tuple, atau string.
RandRange	randrange ([start,] stop [,step])	Elemen yang dipilih secara acak dari jangkauan (start, stop, step).
Random	random()	A random float r, sehingga 0 kurang dari atau sama dengan r dan r kurang dari 1
Seed	seed([x])	Menetapkan nilai awal integer yang digunakan dalam menghasilkan bilangan acak. Panggil fungsi ini sebelum memanggil fungsi modul acak lainnya. Tidak ada pengembalian
Shuffle	shuffle(lst)	Mengacak daftar dari daftar di tempat. Tidak ada pengembalian
Floor	floor(x)	The floor of x: the largest integer not greater than x.
Uniform	uniform(x, y)	Sebuah float acak r, sedemikian rupa sehingga x
Nama	Penggunaan	Penjelasan
		kurang dari atau sama dengan r dan r kurang dari y.

Fungsi Trigonometri

Python mencakup fungsi berikut yang melakukan perhitungan trigonometri. Berikut adalah daftarnya :

Nama	Penggunaan	Penjelasan
Acos	acos(x)	Kembalikan kosinus x, di radian.
Asin	asin(x)	Kembalikan busur sinus x, dalam radian.
Atan	atan(x)	Kembalikan busur singgung x, di radian.

Atan 2	atan2(y, x)	Kembali atan (y / x), di radian.
Kosinus	cos(x)	Kembalikan kosinus x radian.
Hypot	hypot(x, y)	Kembalikan norma Euclidean, $\sqrt{x^2 + y^2}$.
Sin	sin(x)	Kembalikan sinus dari x radian.
Tan	tan(x)	Kembalikan tangen x radian.
Derajat	degrees(x)	Mengonversi sudut x dari radian ke derajat.
Radian	radians(x)	Mengonversi sudut x dari derajat ke radian.

Konstanta Matematika

Modul ini juga mendefinisikan dua konstanta matematika. Berikut adalah daftarnya :

Nama	Penggunaan	Penjelasan
Pi	pi	Konstanta Pi matematika
e	e	Konstanta e matematika

5.14 STRING

String adalah jenis yang paling populer di bahasa pemrograman. Kita bisa membuatnya hanya dengan melampirkan karakter dalam tanda kutip. Python memperlakukan tanda kutip tunggal sama dengan tanda kutip ganda. Membuat string semudah memberi nilai pada sebuah variabel. Dibawah ini adalah contoh sederhana dari sebuah string pada bahasa pemrograman Python.

```
print("Hello World")
```

Python tidak menggunakan tipe karakter titik koma ; Ini diperlakukan sebagai string dengan panjang satu, sehingga juga dianggap sebagai substring.

Untuk mengakses substring, gunakan tanda kurung siku untuk mengiris beserta indeks atau indeks untuk mendapatkan substring Anda. Sebagai contoh :

```
name = 'John Doe' message = "John Doe belajar bahasa python di
Belajarpython"
print ("name[0]: ", name[0]) print ("message[1:4]: ", message[1:3])
```

Bila kode diatas dieksekusi, maka akan menghasilkan hasil sebagai berikut :

name[0]: J message[1:4]: ohn

Mengupdate STRING

Anda dapat "memperbarui" string yang ada dengan (kembali) menugaskan variabel ke string lain. Nilai baru dapat dikaitkan dengan nilai sebelumnya atau ke string yang sama sekali berbeda sama sekali. Sebagai contoh

```
message = 'Hello World' print ("Updated String :- ",  
message[:6] + 'Python')
```

Bila kode diatas dieksekusi, maka akan menghasilkan hasil sebagai berikut : **Updated String :- Hello Python**

Escape Character

Dibawah ini adalah tabel dari daftar karakter escape atau karakter non-printable yang dapat diwakili/ditulis dengan awalan notasi backslash.

Notasi Backslash	Karakter Hexadecimal	Penjelasan
\a	0x07	Bell atau alert
\b	0x08	Backspace
\cx		Control-x
\C-x		Control-x
\e	0x1b	Escape
\f	0x0c	Formfeed
\M-\C-x		Meta-Control-x
\n	0x0a	Newline
\nnn		Octal notation, dimana n berada di range 0..7
\r	0x0d	Carriage return
\s	0x20	Space
Notasi Backslash	Karakter Hexadecimal	Penjelasan
\t	0x09	Tab
\v	0x0b	Vertical tab

\x		Character x
\xnn	asdafsdfsdf	Notasi Hexadecimal, dimana n berada di range 0.9, a.f, atau A.F

Operator Special String

Asumsikan variabel string adalah 'Belajar' dan variabel b adalah 'Python', lalu dibawah ini adalah operator yang bisa dipakai pada kedua string di variabel tersebut.

a =

"Belajar" b

= "Python"

Berikut adalah daftar operator spesial string pada Python :

Operator	Contoh	Penjelasan
+	a + b akan menghasilkan BelajarPython	Concatenation - Menambahkan nilai pada kedua sisi operator
*	a*2 akan menghasilkan BelajarBelajar	Pengulangan - Membuat string baru, menggabungkan beberapa salinan dari string yang sama
[]	a[1] akan menghasilkan e	Slice - Memberikan karakter dari indeks yang diberikan
[:]	a[1:4] akan menghasilkan el	Range Slice - Memberikan karakter dari kisaran yang diberikan
in	B in a akan menghasilkan 1	Keanggotaan - Mengembalikan nilai true jika ada karakter dalam string yang diberikan
not in	Z not in a akan menghasilkan 1	Keanggotaan - Mengembalikan nilai true jika karakter tidak ada dalam string yang diberikan

r/R	print r'\n' prints \n dan print R'\n' prints \n	Raw String - Menekan arti aktual karakter Escape. Sintaks untuk string mentah sama persis dengan string biasa kecuali operator string mentah, huruf "r", yang mendahului tanda petik. "R" bisa berupa huruf kecil (r) atau huruf besar (R) dan harus ditempatkan tepat sebelum tanda kutip pertama.
%		Format - Melakukan format String

Operator Format String

Salah satu fitur Python yang paling keren adalah format string operator %. Operator ini unik untuk string dan membuat paket memiliki fungsi dari keluarga printf C () C. berikut adalah contoh sederhananya :

```
print ("My name is %s and weight is %d kg!" % ('Zara', 21))
```

Berikut adalah daftar lengkap simbol yang bisa digunakan bersamaan dengan % :

Operator	Penjelasan
%c	character
%s	Konversi string melalui str () sebelum memformat
%i	Dianggap sebagai bilangan bulat desimal
%d	Dianggap sebagai bilangan bulat desimal
%u	Unsigned decimal integer
%o	Bilangan bulat oktal
%x	Bilangan bulat heksadesimal (huruf kecil)
%X	Bilangan bulat heksadesimal (huruf besar)
%e	Notasi eksponensial (dengan huruf kecil 'e')
%E	Notasi eksponensial (dengan huruf besar 'E')
%f	Bilangan real floating point
%g	Yang lebih pendek dari% f dan% e
%G	Lebih pendek dari% f dan% E

Triple Quote

Python triple quotes digunakan dengan membiarkan string untuk ditulis dalam beberapa baris, termasuk kata kerja NEWLINEs, TABs, dan karakter khusus lainnya. Sintaks untuk triple quotes terdiri dari tiga tanda kutip tunggal atau ganda ditulis berturut-turut.

Berikut adalah contohnya :

```
lutipantiga = """this is a long string that is made up of several lines and  
non-printable characters such as  
TAB (\t) and they will show up that way when displayed.  
NEWLINEs within the string, whether explicitly given like this  
within the brackets [ \n ], or just a NEWLINE within the variable  
assignment will also show up.  
"""  
print (lutipantiga)
```

String Unicode

Pada Python 3, semua string diwakili dalam Unicode. Sedangkan pada Python 2 disimpan secara internal sebagai 8-bit ASCII, maka diperlukan lampiran 'u' untuk membuatnya menjadi Unicode. Tetapi hal ini tidak lagi diperlukan sekarang.

Metode String Built-in

Python menyertakan metode built-in berikut untuk memanipulasi string

Metode	Penjelasan
capitalize()	Meng-kapitalkan huruf pertama string
center(width, fillchar)	Mengembalikan string yang dilapisi dengan fillchar dengan string asli yang dipusatkan pada total width kolom.
count(str, beg = 0,end = len(string))	Menghitung berapa kali str yang terjadi dalam string atau dalam substring string jika memulai indeks beg dan end index end diberikan.
decode(encoding = 'UTF8',errors = 'strict')	Dekode string menggunakan codec yang terdaftar untuk pengkodean. Encoding default ke pengkodean string default.

encode(encoding = 'UTF8',errors = 'strict')	Mengembalikan versi string yang dikodekan string; Pada kesalahan, default adalah menaikkan ValueError kecuali jika kesalahan diberikan dengan 'ignore' atau 'replace'.
endswith(suffix, beg = 0, end = len(string))	Menentukan apakah string atau substring string (jika memulai indeks memohon dan mengakhiri akhir indeks diberikan) berakhir dengan akhiran; Mengembalikan nilai true jika benar dan salah.
expandtabs(tabsize = 8)	Memperluas tab dalam string ke banyak ruang; Default ke 8 spasi per tab jika tabsize tidak tersedia.
find(str, beg = 0 end = len(string))	Tentukan jika str terjadi dalam string atau dalam substring string jika memulai indeks beg dan end index end diberikan return index jika ditemukan dan -1 sebaliknya.
index(str, beg = 0, end = len(string))	Sama seperti find (), namun menimbulkan pengecualian jika str tidak ditemukan.
isalnum()	Mengembalikan true jika string memiliki minimal 1 karakter dan semua karakternya alfanumerik dan false sebaliknya.
isalpha()	Mengembalikan true jika string memiliki minimal 1 karakter dan semua karakter adalah abjad dan false sebaliknya.
Metode	Penjelasan
isdigit()	Mengembalikan true jika string hanya berisi digit dan false sebaliknya.
islower()	Mengembalikan true jika string memiliki setidaknya 1 karakter casing dan semua karakter casing dalam huruf kecil dan false sebaliknya.
isnumeric()	Mengembalikan true jika string unicode hanya berisi karakter numerik dan false sebaliknya.