

Pertemuan 1-Pengenalan Python

Syntax Dasar

In [9]:	<pre>"hello world"</pre>
Out[9]:	'hello world'
In [11]:	<pre>print("Hello World")</pre> <p>Hello World</p>
In [14]:	<pre>a = 2.2 print("Tipe data", a, "adalah", type(a))</pre> <p>Tipe data 2.2 adalah <class 'float'></p>
In [17]:	<pre>b="ucup" print("Tipe data", b, "adalah", type(b))</pre> <p>Tipe data ucup adalah <class 'str'></p>
In [20]:	<pre>c=3>0 print(c) print("Tipe data", c, "adalah", type(c))</pre> <p>True Tipe data True adalah <class 'bool'></p>
In [23]:	<pre>d=10000 print("Tipe data", d, "adalah", type(d))</pre> <p>Tipe data 10000 adalah <class 'int'></p>
In [26]:	<pre>nama = "Novena Aurelia Luisma" usia = 17 print("Nama saya adalah", nama, "dan usia saya adalah ", usia)</pre> <p>Nama saya adalah Novena Aurelia Luisma dan usia saya adalah 17</p>
In [29]:	<pre>nama = "novena" usia = 17 tinggi = 1.55 benar = True print(type(nama)) print(type(usia)) print(type(tinggi)) print(type(benar)) <class 'str'> <class 'int'> <class 'float'> <class 'bool'></pre>

Pertemuan 2 - Tipe data & Operator Python

How to print

Fungsi print() digunakan untuk mencetak output program di layar.

In [39]:	<pre>print("Hello, World!")</pre> <p>Hello, World!</p>
In [41]:	<pre>print(f"Hello, World!")</pre> <p>Hello, World!</p>

Komentar

komentar merupakan sekumpulan teks yang dituliskan dalam sebuah program dan tidak akan mempengaruhi hasil dari program tersebut. Walaupun tidak mempengaruhi hasil program, komentar merupakan salah satu bagian penting dalam pengkodean, hal ini dikarenakan komentar digunakan sebagai penjelasan alur dari kode program yang kita tulis serta dapat dislipikan di antara sekumpulan statement yang telah dituliskan. komentar ini tentunya juga dapat memudahkan programmer untuk berkomunikasi dengan programmer lainnya.

In [45]:	<pre># ini namanya komentar gaes!!! ''' ini juga komentar gaes, tapi bisa banyak baris ini juga komentar gaes, tapi bisa banyak baris ini juga komentar gaes, tapi bisa banyak baris ini juga komentar gaes, tapi bisa banyak baris ''' print("Komentar tidak akan mengeluarkan output apapun")</pre> <p>Komentar tidak akan mengeluarkan output apapun</p>
----------	---

Variable

Aturan Penulisan Variabel:

- Nama variabel boleh diawali menggunakan huruf atau garis bawah _ . Contoh: nama, _nama, namaKu, nama_variabel
- Karakter selanjutnya dapat berupa huruf, garis bawah _ atau angka. Contoh: _nama, n2, nilai1
- Variabel tidak boleh dimulai dengan angka ataupun simbol.
- Variabel tidak boleh berupa kata kunci (keyword) yang ada dalam Python. Contoh: if, else, while, for, dan masih banyak lagi

In [50]:	<pre># Variabel Nama = "Novena Lluisma" umur = 18 status_pernikahan = "Belum Menikah" statuspernikahan = "Menikah"</pre> <p>Nama: novena dan Umur: 18 Nama: novena dan Umur: 18</p> <p>Novena dan Novera</p>
----------	--

MultiStatment Variabel

MultiStatement Variable dalam Python biasanya digunakan ketika ingin menyimpan atau menulis string atau kode dalam beberapa baris untuk meningkatkan keterbacaan atau menghindari batasan panjang baris.

In [70]:	<pre>nama = "Sayyid" a = 10; b = 50 kesibukan, keterangan = 'Berkuliah', 'Institut Teknologi Telkom Surabaya'</pre>
In [72]:	<pre>print(f"Nama saya {nama} dan Umur saya adalah {a+b} Tahun dan sedang {kesibukan} di {keterangan}")</pre> <p>Nama saya Sayyid dan Umur saya adalah 60 Tahun dan sedang Berkuliah di Institut Teknologi Telkom Surabaya</p>

Input dari Pengguna

Di Python, dapat meminta input dari pengguna menggunakan fungsi bawaan input(). Fungsi ini membaca input dari pengguna sebagai string secara default.

In [75]:	<pre>nama = input("Masukkan nama Anda: ") makan = 'sudah' harga = int(input("Berapa harganya?: ")) print('\n') print("Halo, aku", nama, "dan aku", makan, 'makan ayam goreng yang harganya Rp', harga)</pre> <p>Halo, aku Novena dan aku sudah makan ayam goreng yang harganya Rp 25000</p>
----------	--

Operator Aritmatika

- Penjumlahan (+): Menjumlahkan nilai dari masing-masing operan atau bilangan
- Pengurangan (-): Mengurangi nilai operan di sebelah kiri menggunakan operan di sebelah kanan
- Perkalian (*): Mengalikan operan/bilangan
- Pembagian (/): Untuk membagi operan di sebelah kiri menggunakan operan di sebelah kanan
- Sisa Bagi (%): Mendapatkan sisa pembagian dari operan di sebelah kiri operator ketika dibagi oleh operan di sebelah kanan
- Pangkat (**): Meningkatkan operan disebelah kiri operator dengan operan di sebelah kanan operator
- Pembagian Bulat (//): Sama seperti pembagian. Hanya saja angka dibelakang koma dihilangkan

In [92]:	<pre>#Operator Aritmatika #Penjumlahan print(13+2) apel = 7 jeruk = 9 buah = apel + jeruk print(buah) #Pengurangan hutang = 10000 bayar = 5000 sisaHutang = hutang - bayar print("Sisa hutang Anda adalah ", sisaHutang) #Perkalian panjang = 36 lebar = 6 luas = panjang * lebar print(luas) #Pembagian kue = 20 anak = 5 kuePerAnak = kue / anak print("Setiap anak akan mendapatkan bagian kue sebanyak ", kuePerAnak) #Sisa Bagi / Modulus bilangan1 = 14 bilangan2 = 5 hasil = bilangan1 % bilangan2 print("Sisa bagi dari bilangan ", bilangan1, " dan ", bilangan2, " adalah ", hasil) #Pangkat bilangan3 = 8 bilangan4 = 2 hasilPangkat = bilangan3 ** bilangan4 print(hasilPangkat) #Pembagian Bulat print(10//3) #10 dibagi 3 adalah 3.3333. Karena dibulatkan maka akan menghasilkan nilai 3</pre> <p>15 16 Sisa hutang Anda adalah 5000 216 Setiap anak akan mendapatkan bagian kue sebanyak 4.0 Sisa bagi dari bilangan 14 dan 5 adalah 4 64 3</p>
----------	--

Pertemuan 3 - Tipe Data dan Analisis Masalah

Tipe Data Boolean

Tipe data boolean adalah tipe data yang digunakan untuk mempresentasikan nilai kebenaran. Data ini akan erat kaitannya dengan logika matematika, dimana datanya hanya akan terdiri dari dua jenis, yaitu TRUE dan FALSE. Misalkan kita ingin menguji apakah data yang pertama lebih besar dari data yang kedua, jika benar maka Python akan menampilkan output TRUE, sementara jika salah maka akan menampilkan output FALSE.

In [84]:	<pre>a = True b = False print(a and b) print(a or b) print(not a)</pre> <p>False True False</p>
In [82]:	<pre>x = True y = False print(10 > 5) print(10 < 5)</pre> <p>True False</p>

Operasi Perbandingan

Operator perbandingan (comparison operators) digunakan untuk membandingkan suatu nilai dari masing-masing operan.

- Sama dengan (==): bernilai True Jika masing-masing operan memiliki nilai yang sama, maka kondisi bernilai benar atau True.
- Tidak sama dengan (<>) : bernilai False Akan menghasilkan nilai kebalikan dari kondisi sebenarnya.
- Lebih besar dari (>) : bernilai True Jika nilai operan kiri lebih besar dari nilai operan kanan, maka kondisi menjadi benar.
- Lebih kecil dari (<) : bernilai True Jika nilai operan kiri lebih kecil dari nilai operan kanan, maka kondisi menjadi benar.
- Lebih besar atau sama dengan (>=) : bernilai True Jika nilai operan kiri lebih besar dari nilai operan kanan, atau sama, maka kondisi menjadi benar.
- Lebih kecil atau sama dengan (<=) : bernilai True Jika nilai operan kiri lebih kecil dari nilai operan kanan, atau sama, maka kondisi menjadi benar.

In [96]:	<pre>#Operasi Perbandingan # SAMA DENGAN print(1 == 1) # Hasilnya akan bernilai True karena satu sama dengan satu print(1 != 2) # Hasilnya akan bernilai False karena satu tidak sama dengan dua # TIDAK SAMA DENGAN print(2 != 3) # Hasilnya akan bernilai False karena dua seharusnya sama dengan dua print(2 != 3) # Hasilnya akan bernilai True karena dua tidak sama dengan tiga # LEBIH BESAR DARI print(5 > 3) # Hasilnya akan bernilai True karena lima lebih besar dari tiga # LEBIH KECIL DARI print(5 < 3) # Hasilnya akan bernilai False karena lima tidak lebih besar dari tiga # LEBIH BESAR DARI SAMA DENGAN print(5 >= 3) # Hasilnya akan bernilai True karena lima lebih besar dari sama dengan tiga # LEBIH KECIL DARI SAMA DENGAN print(5 <= 3) # Hasilnya akan bernilai False karena lima tidak lebih besar dari sama dengan tiga True False False True False True True False</pre>
----------	--

Tipe Data String

Tipe data ini merupakan tipe data yang berkaitan erat dengan data yang bentuknya teks. Dalam proses analisis data, data yang kita butuhkan terkadang tidak hanya data numerik, namun juga data string. Dengan memahami tipe data ini, kita akan bisa memprosesnya dengan lebih mudah.

In [99]:	<pre>nama = "Novena Lluisma" print(nama)</pre> <p>Novena Lluisma</p>
In [101]:	<pre>sapa = "Halo" nama = "Novena Lluisma" kalimat = sapa + " " + nama print(kalimat)</pre> <p>Halo Novena Lluisma</p>

Pertemuan 4-Perulangan(dengan For-Loop)

Perulangan (Looping)

Secara umum, looping atau perulangan pada Python adalah instruksi yang program yang dieksekusi berulang kali. Fungsinya untuk memerintah komputer melakukan sesuatu secara berulang-ulang dengan jumlah tertentu selama sebuah kondisi yang telah ditentukan masih terpenuhi. Python memiliki dua jenis perulangan utama: for loop dan while loop.

1. For Loop

Digunakan untuk mengulangi elemen dalam urutan (seperti list, tuple, string, atau range).

In [135]:	<pre>#Loop Melalui List buah = ["apel", "pisang", "anggur"] for buah in buah: print(buah) print() #Loop Melalui String text = "Python" for char in text: print(char) print() # Loop dengan Range # Angka dari 0 hingga 4 for i in range(5): print(i) # Angka dari 2 hingga 5 for i in range(2, 6): print(i)</pre> <p>apel pisang anggur</p> <p>P y t h o n</p> <p>0 1 2 3 4 2 3 4 5</p>
-----------	--

2. while Loop

Digunakan untuk mengulang blok kode selama kondisi bernilai True.

In [141]:	<pre>#Bentuk umum: jawab = 'ya' hitung = 0 while(jawab == 'ya'): hitung += 1 jawab = input("Ulang lagi tidak? ") print(f"Total perulangan: {hitung}") print() #While dengan Kondisi number = 10 while number > 0: print(number - 1) number -= 2 print() #While dengan Break dan Continue count = 0 while count < 5: if count == 3: break print(count) count += 1 count = 0 while count < 5: count += 1 if count == 3: continue print(count)</pre> <p>Total perulangan: 6</p> <p>10 8 6 4 2 0 1 2 1 2 4 5</p>
-----------	---

Percabangan If Else

1. Percabangan 'If' (Single Alternative Selection):

In [148]:	<pre>umur = 18 if umur >= 18: print("Anda sudah dewasa.")</pre> <p>Anda sudah dewasa.</p>
In [150]:	<pre>nilai = 85 if nilai >= 75: print("Selamat! Anda lulus.")</pre> <p>Selamat! Anda lulus.</p>

2. Percabangan 'If-Else' (Dual Alternative Selection):

In [154]:	<pre>nilai = 82 if nilai >= 60: print("Anda lulus.") else: print("Anda belum lulus.")</pre> <p>Anda lulus.</p>
In [156]:	<pre>angka = 7 if angka % 2 == 0: print(f"{angka} adalah bilangan genap.") else: print(f"{angka} adalah bilangan ganjil.")</pre> <p>7 adalah bilangan ganjil.</p>

3. Percabangan 'If-Elif-Else' (Multiple Alternative Selection):

In [159]:	<pre>nomor_hari = 3 if nomor_hari == 1: print("Ini adalah hari Senin.") elif nomor_hari == 2: print("Ini adalah hari Selasa.") elif nomor_hari == 3: print("Ini adalah hari Rabu.") elif nomor_hari == 4: print("Ini adalah hari Kamis.") elif nomor_hari == 5: print("Ini adalah hari Jumat.") else: print("Ini adalah hari akhir pekan.")</pre> <p>Ini adalah hari Jumat.</p>
In [161]:	<pre>suhu = 32 if suhu > 30: print("Cuaca sangat panas.") elif suhu > 20: print("Cuaca hangat.") elif suhu > 10: print("Cuaca sejuk.") else: print("Cuaca dingin.")</pre> <p>Cuaca sangat panas.</p>

4. Percabangan Bersarang (Nested If)

In [165]:	<pre>usia = 25 status_pernikahan = "belum menikah" if usia >= 18: if status_pernikahan == "menikah": print("Anda sudah dewasa dan menikah.") else: print("Anda sudah dewasa dan belum menikah.") else: print("Anda masih remaja.")</pre> <p>Anda sudah dewasa dan belum menikah.</p>
In [167]:	<pre>nilai = 75 prestasi = True if nilai >= 85: if prestasi: print("Anda layak mendapatkan beasiswa penuh!") else: print("Anda layak mendapatkan beasiswa sebagian.") else: print("Maaf, Anda tidak layak mendapatkan beasiswa.")</pre> <p>Maaf, Anda tidak layak mendapatkan beasiswa.</p>