

Set Manipulation – Part 1

Sedikit berbeda dengan tipe data *list* dan *tuple*, pada tipe data set terdapat cukup banyak fitur yang disediakan oleh bahasa Python.

Tugas:

Ketikkan potongan kode pada kolom Contoh Penggunaan di *live code editor*.

Fitur	Keterangan	Contoh Penggunaan
<code>.add()</code>	Menambahkan data ke dalam set. Penting untuk kita ingat bahwa pada tipe data set tidak mengizinkan adanya duplikasi elemen di dalamnya.	mungkin dapat menghasilkan <i>output</i> : <u>Perlu diingat bahwa:</u> tipe data set merupakan <i>unordered collection</i> setiap kali perintah print dijalankan, aku mungkin akan melihat <i>output</i> yang berbeda.
<code>.clear()</code>	Menghapus seluruh elemen dalam sebuah set	akan menghasilkan <i>output</i> : atau empty set {} .
<code>.copy()</code>	Mengembalikan <i>copy</i> dari setiap elemen dalam <i>set</i>	serupa dengan tipe data <i>list</i> , melalui <i>operator assignment</i> =, penambahan pada <code>set_buah2</code> akan mempengaruhi <code>set_buah1</code> melalui penggunaan fitur <code>copy()</code> penambahan pada <code>set_buah2</code> tidak akan mempengaruhi <code>set_buah1</code> menghasilkan <i>output</i> :

		dan menghasilkan <i>output</i> :
<code>.update()</code>	Menambahkan elemen dari suatu set dengan set lainnya.	menambahkan <code>parcel1</code> dengan elemen-elemen dari <code>parcel2</code> menghasilkan <i>output</i> :
<code>.pop()</code>	Menghilangkan elemen dari sebuah set secara acak.	menghasilkan <i>output</i> : >>> <code>Fitur .pop()</code> 'Anggur' atau 'Apel' ataupun 'Jeruk' dan menghasilkan <i>output</i> : {'Apel', 'Jeruk'} atau {'Anggur', 'Jeruk'} ataupun {'Apel', 'Anggur'}
<code>.remove()</code>	Menghilangkan elemen dengan nilai tertentu	menghasilkan <i>output</i> :

Set Manipulation – Part 2

Sedikit berbeda dengan tipe data *list* dan *tuple*, pada tipe data set terdapat cukup banyak fitur manipulasi yang bisa aku gunakan.

Tugas:

Ketikkan potongan kode yang diberikan pada Contoh Penggunaan pada *live code editor*:

Fitur	Keterangan	Contoh Penggunaan
<code>.union()</code>	Mengembalikan hasil penggabungan (union) dari dua buah set.	<p>perhatikan bahwa fungsi <code>union</code> mengembalikan hasil penggabungan setiap elemen milik set <code>parcel1</code> dan <code>parcel2</code> nilai dari set <code>parcel1</code> tidak akan berubah melalui fitur <code>union()</code> tidak seperti fitur <code>update()</code></p> <p>menghasilkan <i>output</i>:</p> <p>dan</p> <p>menampilkan output:</p>
<code>.isdisjoint()</code>	Mengembalikan nilai kebenaran apakah suatu set <i>disjoint</i> (saling lepas/tidak mengandung elemen yang sama) dengan set lainnya.	<p>akan menghasilkan output <code>True</code> dikarenakan <code>parcel1</code> dan <code>parcel2</code> tidak mengandung elemen yang sama</p> <p>akan menghasilkan output <code>False</code> dikarenakan <code>parcel1</code> dan <code>parcel3</code> sama-sama memiliki <code>Apel</code></p>

<code>.issubset()</code>	Mengembalikan nilai kebenaran apakah sebuah set merupakan subset dari set lainnya. Sebuah set A merupakan subset dari set B jika seluruh elemen dari set A merupakan bagian dari set B.	akan menghasilkan output <code>True</code> dikarenakan Anggur dan Apel terdapat dalam set <code>parcel_C</code> akan menghasilkan output <code>False</code> dikarenakan Durian dan Semangka tidak terdapat dalam set <code>parcel_C</code>
<code>.issuperset()</code>	Mengembalikan nilai kebenaran apakah sebuah set merupakan superset dari set lainnya. Sebuah set A merupakan superset dari set B jika seluruh elemen dari set B terkandung dalam set A.	<code>parcel_A</code> , <code>parcel_B</code> , dan <code>parcel_C</code> seperti yang diberikan pada baris fitur <code>.issubset()</code> akan menghasilkan output <code>True</code> dikarenakan C mengandung seluruh elemen dari set A akan menghasilkan output <code>False</code> dikarenakan C tidak mengandung seluruh elemen dari set B
<code>.intersection()</code>	Mengembalikan sebuah set yang merupakan <i>intersection</i> dari dua set lainnya.	dapat menghasilkan <i>output</i> : dikarenakan Apel dan Jeruk terdapat baik pada set <code>parcel_A</code> dan <code>parcel_B</code> .

<code>.difference()</code>	<p>Mengembalikan sebuah set yang berisikan <i>difference</i> dari dua set lainnya.</p> <p><i>Difference</i> dari sebuah set A berdasarkan set B adalah setiap elemen yang terdapat di set A tetapi tidak terdapat di set B.</p>	<p>parcel_A dan parcel_B merupakan set yang sama dengan baris fitur <code>.intersection()</code></p> <p>dapat menghasilkan <i>output</i>:</p>
<code>.symmetric_difference()</code>	<p>Mengembalikan sebuah set yang berisikan <i>symmetric difference</i> dari dua set lainnya.</p> <p><i>Symmetric difference</i> dari sebuah set A dan B adalah setiap elemen dari set A yang tidak terdapat di set B digabungkan dengan (union) setiap elemen dari set B yang tidak terdapat di set A.</p>	<p>dapat menghasilkan <i>output</i>:</p>