

Syntax	Memo
<code><name> = <value></code>	Tilldelning av värde till variabel
<code><listname>.append(<value>)</code>	Tilldelning av värde till en lista
<code><value> <operator> <value></code>	Beräkningssats, där operator kan vara + - * /
<code>= , += , -= , *= , /=</code>	Tilldelningsoperatorer. För att spara nytt värde, addera ett värde, subtrahera, multiplicera ett värde, dela ett värde
<code>> < <= >= ==</code>	Jämförelseoperatorer. För att jämföra två värden
<code>float, int, str, list</code>	Vanligt förekommande datatyper
<code><typename>(<value>)</code>	Typkonvertering, för att kovertera datatypen på value till typen som angivs i typename
<code>if (<vilkor>): <commands></code>	Vilkorssats som körs en gång om vilkor är True
<code>elif (<vilkor>): <commands></code>	Vilkorsats som körs en gång om tidigare vilkor är False, och nuvarande är True
<code>else: False <commands></code>	Else-sats som körs en gång om tidigare vilkor är False
<code>while (<vilkor>): vilkor är True <commands></code>	Vilkorssats som körs flera gånger så länge som vilkor är True
<code>for (<value> in <list>): <commands></code>	List-iterator. Körs en gång för varje värde i listan. Där varje värde placeras i value

<code>def <funcname>(<param>):</code>	Funktionsdefinition. Definerar kommandon som körs
<code>för funktionen.</code>	
<code> <commands></code>	in-data till funktionen definieras i param
<code> [return <value>]</code>	optional. funktionens resultat returneras.
<code><funcname>(<param>)</code>	Funktionsanrop. Kör den angivna funktionen, med
<code>angivna parametrar.</code>	
<code>class <classname>():</code>	Klassdefinition.
<code> <datamedlemmar>:<type></code>	Definition av datamedlem, och dess typ
<code> def __init__(self):</code>	standardfunktion för klasser.
<code> <commands></code>	
<code> def __str__(self):</code>	standardfunktion för klasser.
<code> <commands></code>	
<code> return <stringvalue></code>	
<code><objectname> = <classname>()</code>	Skapande av ett objekt från en klass.