

Hilfsmittel zur Prüfung "Grundlagen der Programmierung"

<code>type(<variable>)</code>	gibt den Typ der Variablen oder des Werts zurück
<code>int(47.5) / float (3) / str(123)</code>	Casting der Werte in der Klammer auf den vorne angegebenen Typ
<code>print("text")</code>	Ausgabe von Text auf der Kommandozeile
<code>i // j</code>	Division ohne Rest
<code>i % j</code>	Modulo (arbeite mit Rest)
<code>i ** j</code>	i hoch j
<code>"HST"[1] → "S"</code> <code>"HST"[-1] → "T"</code>	Indexierung der Zeichen im String
<code>"HST"[1:2] → "S"</code> <code>"HST"[:2] → "HS"</code> <code>"HST"[1:] → "ST"</code> <code>"HST"[:] → "HST"</code> <code>"TUAS"[1:4:2] → "US"</code>	Substring von Zeichen
<code>len("HST") → 3</code>	Ausgabe der Länge des Strings
<code>3 * "hallo" → "hallohallohallo"</code>	Wiederholung des Strings
<code>"schule" in "Hochschule" → True</code>	Test, ob Zeichenkette in String vorkommt
<code>hst.count('hst')</code>	Zählt wie oft 'hst' im String hst vorkommt
<code>while <condition True>:</code> <code> break / continue	While-Schleife Abbruch in der Schleife
<code>for <variabe> in range(...):</code> <code> break / continue	For-Schleife
<code>range(<start>, <stop>, <incr>)</code>	Range-Funktion; läuft von Startwert bis Stoppwert (excl.) in der angegebenen Schrittweite
<code>if char.isnumeric():</code>	Prüft ob alle Zeichen in der Zeichenkette Buchstaben sind
<code>elif char.isalpha():</code>	Prüft ob alle Zeichen in der Zeichenkette Ziffern sind
<code>bearbeiteter_text[::-1]</code> <code>[::-1] = [start:stop:step]</code> (Slicing Methode) <code>= "txet retetiebraeB"</code>	Erstellt umgekehrte Kopie einer Originalsequenz, hier: "bearbeiteter_text"
<code>def beliebig(*args):</code>	Nimmt so viele Argumente auf wie möglich
<code>text = "Du bist ein guter Mensch"</code> <code>zusammengefügtter_text =</code> <code>"".join(text.split(" "))</code> <code>print(zusammengefügtter_text)</code> # Ausgabe: DubisteinguterMensch	Wörter eines Strings zusammenfügen
<code>range(1, breite + 1, 2)</code> <code>range(1, 11, 2)</code>	1 Startwert; breite + 1 Endwert ;2 Schritt zwischen den Werten. 1, 3, 5, 7, 9
<code>for i in reversed(wort)</code>	Rückwärtsausgabe

<code>str('texttexttextprobe-text').count('probe')</code>	Prüft, wie häufig ein String in einem String vorkommt.
<code>import random random.randint(1,10)</code>	Zufallszahl von 1-10
<code>random.choice()</code>	
<code>import string abc = string.ascii_letters</code>	Importiert ABC
<code>Indexzahl= abc.index(buchstabe)</code>	
<code>text = "Hochschule Trier" liste_aus_wörtern = text.split(" ") print(liste_aus_wörtern) # Ausgabe: ['Hochschule', 'Trier']</code>	Wörter eines Strings in eine Liste packen
<code>liste = ['2', '3', '4'] wert = list(map(int, liste)) print(wert) # Ausgabe: [2, 3, 4] liste = [2, 3, 4] wert = list(map(str, liste)) print(wert) # Ausgabe: ['2', '3', '4']</code>	String-Elemente einer Liste in integers umwandeln und andersrum
<code>Listenname.append("...")</code> <ul style="list-style-type: none"> • Bei Set: <code>Setname.add("...")</code> • Bei Tupel: es ist NICHT möglich ein Tupel zu erweitern, 	zum Erweitern der Liste um ein Element am Ende
<code>Listenname.extend([...,...])</code> <ul style="list-style-type: none"> • Set: <code>testset.update(andererSetname)</code> 	zum Anfügen einer Liste am Ende
<code>Listenname.pop(...)</code> <ul style="list-style-type: none"> • Bei Set: <code>Testset.discard("...")</code> Oder: <code>del listenname[index]</code>	Zum Entfernen eines Elementes, aus der Liste. Zum Entfernen eines bestimmten Elementes, aus der Liste.
<code>meine_liste = [10, 20, 30, 40, 50] element = 30 index = meine_liste.index(element)</code>	Index eines Elements aus einer Liste ausgeben
<code>liste2.extend(liste1)</code>	Kopieren der Werte von liste1 in liste2
<code>SortierteListe=sorted(liste,key=len)</code>	Liste sortieren
<code>del liste[index_to_remove]</code>	Ein Index mit dazugehörigen Element aus der Liste löschen
<code>SET{}</code>	
<code>Set.update Set.remove</code>	
<code>Dictionary {'...':'...'} Dictionary['Color'] = green del(Dictionary ['Color'])</code>	Add / Hinzufügen Del / Löschen
<code>Tuple = (' ', ' ')</code>	unveränderbar
<code>Class x def __init__(self,...)</code>	Klassen