

# Programming 1 – Complete Cheatsheet

Jialin Li  
2022-2023

## Inhoudstafel

<b>Basis</b> .....	<b>2</b>
<b>Format</b> .....	<b>2</b>
<b>Strings</b> .....	<b>3</b>
<b>Lists</b> .....	<b>4</b>
<b>Tuples</b> .....	<b>5</b>
<b>Sets</b> .....	<b>5</b>
<b>Classes</b> .....	<b>6</b>
<b>Tekstbestanden</b> .....	<b>6</b>
<b>Handige modules</b> .....	<b>7</b>
math module .....	7
random module .....	7
sys module .....	7
copy module .....	7
os.path module .....	8
os module .....	8

## Basis

Invoer	Uitvoer	Beschrijving
<code>print("één", 1, end="\n")</code>	'één 1'	Print waarden gescheiden door spatie. Optionele <code>end</code> -attribuut bepaalt waar volgende print komt, bv. <code>end=\n</code> of <code>end=""</code>
<code>print("one", "two", sep="X")</code>	'oneXtwo'	Print waarden gescheiden door optionele parameter <code>sep=""</code> .
<code>print('mango\'s')</code>	'mango's'	Bakslash behandelt volgend teken als string.
<code>+ - * / **</code>		Optelling, aftrekking, vermenigvuldiging, deling, machtsverheffing.
<code>5 // 2</code>	2	Integer deling: rondt af naar beneden.
<code>5 % 2</code>	1	Modulo: rest na deling.
<code>int()   float()   str()</code>		Converteert naar type int (afronden naar beneden)   float (eventueel .0 erachter plaatsen)   string.
<code>abs(-1)   round(4.56, 1)</code>	1   4.6	Absolute waarde   afronden, optionele parameter is aantal cijfers na of voor de komma.
<code>max(1, 2, 5)</code>	5	Maximum van 2 of meer waarden. Optionele parameter <code>key=func()</code> .
<code>min("x", "ab", key=len)</code>	'x'	Minimum van 2 of meer waarden. Optionele parameter <code>key=func()</code> .
<code>len("abc")</code>	3	Lengte van een object.
<code>sorted(['ab', 'b'], key=len)</code>	['b', 'ab']	Geeft een gesorteerde versie weer van een sequentie. Optionele parameter <code>key</code> .
<code>x = input("Tekst: ")</code>	Tekst:	Slaagt de input (string) van de gebruiker op in variabele <code>x</code> .
<code>a, b = b, a</code>		Wisselt de waarden van variabelen <code>a</code> en <code>b</code> om.
<code>type(a)</code>		Geeft weer tot welke datatype of klasse een object behoort, bv. <code>str</code> , <code>int</code> , <code>float</code> , ...
<code>isinstance('5', int)</code>	False	Geeft True of False. Checkt of object behoort tot het gegeven type of class.
<code>range(start=0, einde, stap=1)</code>		Iterable van optionele start (incl.) tot einde (excl.) met optionele stap.
<code># Commentaar   """Commentaar"""</code>		Commentaar op 1 lijn   over meerdere lijnen.
<code>assert var &gt; 0, "Foutmelding"</code>		Checkt in programma of aan een voorwaarde voldaan wordt. Zo niet, wordt de optionele boodschap geprint en stopt het programma.

## Format

Invoer	Uitvoer	Beschrijving
<code>"A. {}, B. {}".format(1, 2)</code>	'A. 1, B. 2'	(Oude) format.
<code>"A. {1}, B. {0}".format(1, 2)</code>	'A. 2, B. 1'	(Oude) format, optionele volgorde.
<code>var = '123'</code> <code>f"tekst: {var}"</code>	'123'	(Nieuwe) format.
<code>f"{'a':&lt;3},{'b':^3},{'c':*&gt;3}"</code>	'a , b ,**c'	Neemt een minimale aantal karakters als ruimte, wordt opgevuld met witruimte of zelfgekozen karakter. Uittlijning links, midden of rechts.
<code>f"{1.235:.2f}"</code>	'1.24'	Specificeert format getal als int (d) of float (f). Specificeert aantal cijfers na de komma, afgerond.
<code>f"{1.235:^7.2f}"</code>	' 1.24 '	Combinatie van voorgaande.

# Strings

Invoer	Uitvoer	Beschrijving
<code>s1 = "aardbei"   s2 = "bes"</code>	<code>'aardbei'   'Bes'</code>	Initialiseert een string.
<code>len(s1)</code>	7	Geeft lengte van string.
<code>max(s1)   min(s)</code>	<code>'r'   'a'</code>	Geeft grootste   kleinste karakter.
<code>ord('a')   chr(97)</code>	97   <code>'a'</code>	Geeft overeenkomstige ASCII-ordening of -teken
<code>s[2]   s[2:5:1]</code>	<code>'r'   'rdb'</code>	Substring via index of slice (excl. bovengrens, optionele stap).
<code>s[::-1]</code>	<code>'iebdraa'</code>	String inverteren.
<code>s1 + s2</code>	<code>'aardbeiBes'</code>	Strings samenvoegen (concateneren).
<i>LET OP: Onderstaande methodes veranderen de originele string niet, maar geven eventueel een bewerkte kopie.</i>		
<code>s.strip()   s1.strip('a')</code>	<code>'rdbei'</code>	Verwijdert witruime of optionele zelfgekozen karakters aan het begin en einde van de string.
<code>s.lstrip()   s.rstrip()</code>		Idem als strip(), maar enkel aan begin of einde.
<code>s2.upper()   s2.lower()</code>	<code>'BES'   'bes'</code>	Creëert een versie van een string met alle letters als hoofdletters of kleine letters.
<code>s1.title()   s1.capitalize()</code>	<code>'Aardbei'</code>	Maakt eerste letter van elk woord hoofdletter.   Maakt enkel eerste letter hoofdletter.
<code>s2.swapcase()</code>	<code>'bES'</code>	Inverteert hoofdletters en kleine letters.
<code>s1.count('a')</code>	2	Geeft weer hoe vaak een substring voorkomt.
<code>s1.find("a", 0, 5)</code>	0	Geeft laagste index van gegeven substring in de string. -1 indien niet gevonden. Optionele start en einde waar het moet zoeken (excl. einde).
<code>s1.index("a", 2, 5)</code>	ValueError	Idem als find(), maar geeft ValueError als substring niet wordt gevonden.
<code>s.rfind()   s.rindex()</code>		Idem als find() en index() maar geeft hoogste index.
<code>s1.replace("a", "X")   s1.replace("a", "X", 1)</code>	<code>'XXrdbei'   'Xardbei'</code>	Kopie van string waarbij alle instanties van een substring vervangen wordt. Optionele parameter geeft aan hoeveel instanties er vervangen moeten worden.
<code>s1 in s2</code>	False	Geeft True of False. Checkt of s1 substring is van s2.
<code>s.islower()   s.isupper()   s.istitle()</code>		Geeft True of False. Checkt of string enkel kleine letters bevat   enkel hoofdletters bevat   elke woord begint met een hoofdletter.
<code>s1.startswith('aard')   s1.endswith('ei')</code>	True	Geeft True of False. Checkt of strings start   eindigt met substring.
<code>s1.startswith(('a', 'b', 'c'))   s1.endswith(('x', 'e'))</code>	True	Geeft True of False. Checkt of string start   eindigt met één van de substrings in de tuple.
<code>s.isalpha()</code>		Geeft True of False. Check of string enkel bestaat uit letters A-Za-z (spaties of leestekens -> False).
<code>s.isnumeric()</code>		Geeft True of False. Check of string enkel bestaat uit cijfer.
<code>s.isalnum()</code>		Geeft True of False. Checkt of string enkel bestaat letters en/of cijfers.
<code>" ".join([s1, s2])   s2.join("123")</code>	<code>'aardbei bes'   '1bes2bes3'</code>	Joint string met sequentie.
<code>"a b cde".split()</code>	<code>['a', 'b', 'cde']</code>	Geeft lijst van woorden in string, gesplitst op witruimte.
<code>"a b cde".split('c')</code>	<code>['a b ', 'de']</code>	Geeft lijst van woorden in string, gesplitst op optionele parameter.
<code>s.splitlines()</code>		Geeft lijst van substrings, gesplitst op newlines.

# Lists

Invoer	Uitvoer	Beschrijving
<pre>l = ['1', '23', '456']   l = list(sequentie)</pre>	<pre>['1', '23', '456']</pre>	Initialiseert een lijst.
<pre>l_2 = [x for x in range(5) if x % 2 == 0]</pre>	<pre>[0, 2, 4]</pre>	List comprehension.
<pre>len(l)</pre>	<pre>3</pre>	Geeft lengte van de lijst (aantal elementen).
<pre>max(l)   min(l, key=len)</pre>	<pre>'456'   '1'</pre>	Geeft grootste   kleinste element. Optionele key.
<pre>sum(l_2)</pre>	<pre>6</pre>	Som van elementen in de lijst.
<pre>l[2]   l[:2:1]</pre>	<pre>'456'   ['1', '23']</pre>	Element of sublijst (slice) uit de lijst (excl. bovengrens). Optionele stap.
<pre>l[::-1]</pre>	<pre>['456', '23', '1']</pre>	Lijst inverteren.
<pre>l + l_2</pre>	<pre>['1', '23', '456', 0, 2, 4]</pre>	Lijsten samenvoegen (concateneren).
<pre>'1' in l</pre>	<pre>True</pre>	Geeft True of False. Checkt of de lijst een element bevat.
<i>LET OP: Onderstaande methodes passen de originele lijst aan, in tegenstelling tot bij strings.</i>		
<pre>l.append('7')</pre>	<pre>['1', '23', '456', '7']</pre>	Voegt nieuw element achteraan toe aan de lijst.
<pre>l.extend(l_2)</pre>	<pre>['1', '23', '456', 0, 2, 4]</pre>	Maakt list langer door een sequentie aan toe te voegen.
<pre>l.insert(1, 'a')</pre>	<pre>['1', 'a', '23', '456']</pre>	Voegt element toe aan lijst op specifieke plaats.
<pre>l.remove('23')</pre>	<pre>['1', '456']</pre>	Verwijdert eerste instantie van element uit de lijst.
<pre>var = l.pop(1)</pre>	<pre>'23'</pre>	Verwijdert laatste element of optioneel element met gegeven index uit de lijst. Retourneert het element.
<pre>del l[1]   del l[:2]</pre>	<pre>['1', '456']   ['456']</pre>	Verwijdert element of sublijst (via slice) uit lijst.
<pre>l.clear()</pre>	<pre>[]</pre>	Maakt lijst leeg
<pre>l.index('23')</pre>	<pre>1</pre>	Geeft de index van de eerste instantie van een element uit de lijst.
<pre>l.count('456')</pre>	<pre>1</pre>	Geeft weer hoe vaak een element voorkomt in de lijst.
<pre>l.sort(key=len, reverse=True)</pre>	<pre>['456', '23', '1']</pre>	Sorteert elementen in de lijst van laag naar hoog of via optionele key, of optioneel in omgekeerde volgorde.
<pre>l_2.reverse()</pre>	<pre>[4, 2, 0]</pre>	Lijst inverteren.
<pre>copy.copy(l)</pre>		(import copy) Maakt een ondiepe kopie van de lijst.
<pre>copy.deepcopy(l)</pre>		(import copy) Maakt een diepe kopie van de lijst.
<pre>" ".join(l)</pre>	<pre>'1 23 456'</pre>	Joint string met elementen uit de lijst.
<pre>"a b cde".split()</pre>	<pre>['a', 'b', 'cde']</pre>	Geeft lijst van woorden in string, gesplitst op witruimte.
<pre>"a b cde".split('c')</pre>	<pre>['a b ', 'de']</pre>	Geeft lijst van woorden in string, gesplitst op optionele parameter.
<pre>s.splitlines()</pre>		Geeft lijst van substrings, gesplitst op newlines.

# Tuples

Invoer	Uitvoer	Beschrijving
<code>t = ('1', '23', '456')   t = tuple(sequentie)</code>	<code>('1', '23', '456')</code>	Initialiseert een tuple.
<code>len(t)</code>	3	Geeft lengte van de tuple (aantal elementen).
<code>max(t)   min(t, key=len)</code>	<code>'456'   '1'</code>	Geeft grootste   kleinste element. Optionele key.
<code>sum((1, 2, 3))</code>	6	Som van elementen in de tuple.
<code>t[2]   t[:2:1]</code>	<code>'456'   ('1', '23')</code>	Element of subtuple (slice) uit de tuple (excl. bovengrens). Optionele stap.
<code>t[::-1]</code>	<code>('456', '23', '1')</code>	Tuple inverteren.

# Sets

Invoer	Uitvoer	Beschrijving
<code>s1 = {'1', '23', '456'}   s2 = set('156')</code>	<code>{'1', '23', '456'}   {'6', '1', '5'}</code>	Initialiseert een set.
<code>len(s1)</code>	3	Geeft lengte van de set (aantal elementen).
<code>max(s1)   min(s1, key=len)</code>	<code>'456'   '1'</code>	Geeft grootste   kleinste element. Optionele key.
<code>sum({1, 2, 3})</code>	6	Som van elementen in de set.
<code>s1.add('a')</code>	<code>{'a', '1', '23', '456'}</code>	Voegt element toe aan de set.
<code>s1.update('bbq')</code>	<code>{'b', '1', 'q', '23', '456'}</code>	Voegt alle elementen van een sequentie toe aan de set.
<code>s1.remove('1')   s1.discard('1')</code>	<code>{'23', '456'}</code>	Verwijders een element uit de set. Discard() negeert niet-bestaande elementen. Remove() geeft een error.
<code>s1.pop()</code>	<code>'1'</code>	Verwijdert een willekeurig element uit de set en retourneert de waarde.
<code>s.clear()</code>	<code>set()</code>	Verwijdert alle elementen uit de set.
<code>copy.copy(s)</code>		( <code>import copy</code> ) Maakt een ondiepe kopie van de set.
<code>copy.deepcopy(s)</code>		( <code>import copy</code> ) Maakt een diepe kopie van de set.
<code>s1.union(s2)</code>	<code>{'23', '456', '6', '1', '5'}</code>	Unie / alle elementen van beide set.
<code>s1.intersection(s2)</code>	<code>{'1'}</code>	Doorsnede / gemeenschappelijke elementen van beide sets.
<code>s1.difference(s2)</code>	<code>{'23', '456'}</code>	Geeft een set van alle elementen van s1, waarvan gemeenschappelijke elementen met s2 verwijderd zijn.
<code>s1.symmetric_difference(s2)</code>	<code>{'23', '456', '6', '5'}</code>	Geeft alle elementen die s1 en s2 niet gemeenschappelijk hebben.
<code>s1.isdisjoint(s2)</code>	False	Geeft True of False. Checkt of de twee sets enkel verschillende elementen hebben.
<code>s1.issubset({'1', '23', '456', '7'})</code>	True	Geeft True of False. Checkt of s1 een subset is van s2.
<code>s1.issuperset({'1'})</code>	True	Geeft True of False. Checkt of s2 een subset is van s1.
<code>s3 = frozenset(('a', 'b', 'c'))</code>	<code>frozenset({'c', 'a', 'b'})</code>	Creëert een onveranderbare set.

# Classes

Invoer	Uitvoer	Beschrijving
<code>class</code> KlasseNaam:		Initialiseert een klasse.
<code>def __init__(self, p1, p2):</code>		Initialisatie / constructor van de klasse. Optioneel parameters p1, p2, ... definiëren die je moet meegeven bij het creëren van een object.
<code>def __repr__(self):</code>		Hiermee kan je regelen wat er getoond wordt als je een object van een bepaalde klasse aanroept of print.
<code>def __str__(self):</code>		Hiermee kan je regelen wat er getoond wordt als je een object van een bepaalde klasse print.
<code>def __eq__(self, other):</code>		Hiermee kan je bepalen wanneer twee objecten hetzelfde zijn.
<code>var1 = KlasseNaam()</code>		Creëert object van een klasse.
<code>type(var1)</code>	<code>&lt;class 'KlasseNaam'&gt;</code>	Geeft weer tot welke datatype of klasse een object behoort.
<code>var1.__class__</code>	<code>&lt;class 'KlasseNaam'&gt;</code>	Geeft weer tot welke klasse een object behoort.
<code>isinstance(var1, KlasseNaam)</code>	<code>True</code>	Geeft True of False. Checkt of object behoort tot het gegeven type of klasse.
<code>var2 = copy(var1)</code>		( <code>from copy import copy</code> ) Maakt een ondiepe kopie van het object zonder onderlinge relatie.
<code>var3 = deepcopy(var1)</code>		( <code>from copy import deepcopy</code> ) Maakt diepe kopie van object zonder onderlinge relatie.

# Tekstbestanden

Invoer	Beschrijving
<code>f = open('file.txt', mode='r')</code>	Open gegeven bestand. Optionele modus: default 'r' (read), 'w' (write), 'a' (append), 'x' (create).
<code>f.close()</code>	Bestand sluiten.
<code>with open ('file.txt') as f: &lt;acties&gt;</code>	Opent bestand en voert acties uit. Hoeft geen close() te gebruiken hier.
<code>f.read()</code>	Leest een bestand en retourneert een string.
<code>f.readline()</code>	Leest een regel van een bestand en retourneert een string.
<code>f.readlines()</code>	Leest alle regels van een bestand en retourneert ze als elementen (incl. "\n") van een lijst.
<code>f.write("tekst")</code>	Schrijft een string naar een bestand.
<code>f.writelines(["a", "b"])</code>	Schrijft een lijst van strings naar een bestand.

# Handige modules (import ...)

## math module

Invoer	Uitvoer	Beschrijving
<code>sqrt(4)</code>	2.0	Vierkantswortel
<code>exp(1)</code>	2.71828...	Exponentiële / e to de macht.
<code>log(2)</code>	0.69314...	Natuurlijk logaritme.
<code>log10(100)</code>	2.0	Logartime met 10 als basis.

## random module

Invoer	Uitvoer	Beschrijving
<code>random()</code>	0.10542...	Random float in bereik [0, 1) dus excl. 1.
<code>uniform(2.5, 3.7)</code>	2.8654...	Random float in bereik incl. beide grenzen.
<code>randint(2, 5)</code>	2	Random int in bereik incl. beide grenzen.
<code>randrange(0, 50, 10)</code>	20	Random int in range excl. bovengrens. optionele stap.
<code>choice(['a', 'b', 'c'])</code>	'a'	Random element uit sequentie (lijst, tuple, string, range, ...).
<code>choices([1, 2, 3], weights=[5, 2, 2], k=2)</code>	[2, 1]	Random lijst van elementen uit gegeven sequentie. Optionele weights- en k-parameter geven gewichten van keuzes en lengte van gegenereerde lijst.
<code>shuffle([1, 2, 3])</code>	[1, 3, 2]	Shuffelt een sequentie random door elkaar. Wordt uitgevoerd op de originele sequentie!
<code>seed()</code>   <code>seed(1)</code>		Maakt de generator volledig random   Genereert telkens dezelfde waarden.

## sys module

Invoer	Uitvoer	Beschrijving
<code>exit()</code>		Stopt het programma.

## copy module

Invoer	Uitvoer	Beschrijving
<code>b = copy(a)</code>		Maakt ondiepe kopie van een object (bv. lijst, klasse, ...) zonder onderlinge relatie.
<code>b = deepcopy(a)</code>		Maakt diepe kopie van een object (bv. lijst, klasse, ...) zonder onderlinge relatie.

## os.path module

Invoer	Uitvoer	Beschrijving
<code>exists("map/file.txt")</code>	True	Geeft True of False. Checkt of het pad (bestand of folder) bestaat.
<code>isfile("map/file.txt")</code>	True	Geeft True of False. Checkt of het pad een bestand is.
<code>isdir("map/file.txt")</code>	False	Geeft True of False. Checkt of het pad een folder is.
<code>join('C:\\Usr', 'f.txt')</code>	'C:\\Usr\\f.txt'	Bouwt een pad (string) met gegeven folder en bestandsnaam.
<code>basename('C:\\Users\\f.txt')</code>	'f.txt'	Retourneert de bestandsnaam uit een gegeven pad.
<code>dirname('C:\\Users\\f.txt')</code>	'C:\\Users'	Retourneert de folder uit een gegeven pad.
<code>getsize('f.txt')</code>	1024	Retourneert de grootte in bytes van het bestand (integer).

## os module

Invoer	Uitvoer	Beschrijving
<code>getcwd()</code>	'C:\\Users\\...'	Get current working directory. Geeft huidige folder weer.
<code>listdir("./")</code>	['Documents', ...]	Retourneert een lijst met alle paden in bv. huidige folder.