

Instructie	Betekenis	Voorbeeld
Print()	Resultaat weergeven	Print("Hello, world!")
Int() → Type Casting	geeft de parameter waarde terug als integer	Int(7.5)
Float() → Type Casting	geeft de parameter waarde terug als kommagetal	Float(7)
Str() → Type Casting	geeft de parameter waarde terug als string	Str(8)
Syntax Errors	Fouten in code zodat deze niet uitgevoerd kan worden	Print("Hello, world!")
Runtime Errors	Fouten tijdens uitvoeren van code	ZeroDivisionError
Type()	Geeft het datatype van een waarde weer	Type(a)
Pow(basis, exponent, modulo)	Verheft de basis tot de exponent en neemt er eventueel de modulo van	Pow(2 ,3, 3)
Abs()	Geeft de absolute waarde van een waarde	Abs(-5)
Max()	Retourneert de grootste parameter	Max(2, 10, 5)
Min()	Retourneert de kleinste parameter	Min(2, 10, 5)
Round(getal, decimalen)	Rond een waarde af op een aantal parameters	Round(3.1356543, 3)
Len	Geeft de lengte van een parameter terug	Len("Patrick")
Input()	Vraagt de gebruiker naar input	Input("Geef een getal: ")
Print(..., sep = <string>)	Definieer een separator bij de print	Print(..., sep = "x")
Print(..., end = <string>)	Definieer een eindwaarde bij de print	Print(..., end= " ")
f"a:uitlijning ruimte .decimalen variabeletype" f"string1" if a <conditie> else "string2")'	Geeft een variabele a terug in combinatie van aanlijning, precisie en decimalen en eventueel een conditie	Print("{:>5.2f}".format(3.784)) f"Is {x} even? {"Ja" if x%2 == 0 else "Nee"}"
Import.../ From ... import ...	Importeer (een deel van) een module	Import math
Math.exp()	Retourneert 'e' tot parameter	Math.exp(3)

Math.log()	Retourneert natuurlijk logaritme van parameter	Math.log(12)
Math.log10()	Retourneert 10-logaritme van parameter	Math.log10(1000)
Math.sqrt()	Retourneert vierkantswortel van parameter	Math.sqrt(144)
Random.random()	Retourneert random float uit [0,1[Random.random()
Random.randint(x, y)	Retourneert parameter uit [x,y]	Random.randint(2, 5)
Random.seed()	Initialiseert de toevalsgenerator	Random.seed(1)
sys.exit()	Sluit het hele programma af met een 'error'	sys.exit()
Range(startgetal, eindgetal, stapgrootte)	Iteratie over [startgetal, eindgetal[met gegeven stapgrootte	Range(1, 10, 2)
Continue()	Herstart huidige loop	Continue()
Break()	Breekt uit huidige loop	Break()
Def naamfunctie(): <actie>	Initialiseert een functie	Def bereken_som(): ...
Return()	Zorgt ervoor dat de functie een waarde teruggeeft	Return(som_kwadraten)
Isinstance(parameter, datatype)	Checkt of een parameter een bepaald datatype is	Isinstance(5, int)
Global	Zorgt ervoor dat globale variabele kan gewijzigd worden binnen functie	Global x
lambda <parameters>: <actie>	Initialiseert een anonieme functie bestaande uit 1 regel code	f = lambda x: x*x
Pass	Plaatsvuller die niets doet	Pass
type() / __class__	Retourneert informatie over de klasse	p.__class__
__init__()	Zorgt dat de functie voor iedere instantie van de klasse wordt toegepast	def __init__(self):
__str__() / __repr__()	Formateert output van klasse	def __repr__(self): ...

<code>__eq__(self,other)</code>	Zorgt ervoor dat == operator gebruikt kan worden	<code>def __eq__(self,other):</code>
<code>Assert ..., 'message'</code>	Checkt of de voorwaarde voldoet en geeft anders een error message	<code>assert breedte > 0 and hoogte > 0, 'ongeldige rechthoek'</code>
<code>Copy.copy()</code>	Maakt een hard copy van een parameter	<code>Copy.copy(punt)</code>
<code>\n</code>	Teken voor nieuwe regel	<code>\n</code>
<code>\t</code>	Teken voor inspringing	<code>\p</code>
<code>String[a:b:c]</code>	Print substring af die begint bij a, eindigt bij b-1 en stapgrootte c	<code>"josefien"[2:6:2]</code>
<code>String.strip()</code>	Verwijdert alle spaties, newline tekens,... aan begin en einde van string	<code>" En nu iets heel anders \n ".strip()</code>
<code>String.find("a",b,c)</code>	Zoekt in de string naar eerste voorkomen van string a vanaf b tot c, anders -1	<code>"barabara".find("bar")</code>
<code>String.replace("a","b",c)</code>	Vervangt in de string c keer a in b	<code>"barabara".replace("bar","ber",2)</code>
<code>String.split("a", aantal splits)</code> <code>Re.split('splitter', string)</code>	Maakt een lijst van de delen van de string met a als separator	<code>"Jos eet kaas".split(" ")</code>
<code>String.isalpha()</code>	Retourneert true als alle tekens letters zijn	<code>"Jef".isalpha()</code>
<code>String.isdigit()</code>	Retourneert true als alle tekens cijfers zijn	<code>"47484".isdigit()</code>
<code>String.isspace()</code>	Retourneert true als alle tekens spaties of newlinetekens zijn	<code>" \n".isspace()</code>
<code>String.rstrip()</code>	Verwijdert alle spaties, newline tekens,... aan einde van string	<code>"En nu iets heel anders \n ".rstrip()</code>
<code>String.lstrip()</code>	Verwijdert alle spaties, newline tekens,... aan begin van string	<code>" En nu iets heel anders".lstrip()</code>
<code>String.rfind("a",b,c)</code>	Zoekt in de string naar laatste voorkomen van string a vanaf b tot c	<code>"barabara".rfind("bar")</code>
<code>"a".join(Lijst)</code>	Plakt een lijst van strings aaneen met a als separator	<code>" ".join(list)</code>
<code>Sum(list, a)</code>	Telt alle waarden van de list op met startwaarde a	<code>Sum(lijst, 10)</code>

List.append(a)	Voegt element a toe aan einde van list	Lijst.append("peer")
List.extend(a)	Voegt lijst a toe aan einde van list	Lijst.extend(["framboos", "aardbei", "aalbes"]) print(fruitlist)
List.insert(a, b)	Voegt element b toe op index a van list	Lijst.insert(3, "banaan")
List.remove(a)	Verwijdert de eerste instantie a in list, anders runtime error	Lijst.remove("appel")
List.pop(a)	Verwijdert element met index a in list en retourneert dit, anders runtime error	Lijst.pop(2)
List.del(a)	Verwijdert element of substring met index a zonder retourwaarde, anders runtime error	Del lijst[2:5]
List.clear()	Maakt list helemaal leeg	Lijst.clear()
Random.shuffle(list)	Ordent de elementen in een lijst random	Random.shuffle(lijst)
List.index(a, startwaarde, stopwaarde)	Retourneert de index van de eerste instantie van a in list, anders runtime error OOK VOOR TUPLES!	Lijst.index("ananas")
List.count(string, startwaarde, eindwaarde)	Telt hoeveel instanties er van de string in de list zijn.	['wortel', 'peer'].count('wortel')
List.sort()	Sorteert numerieke list chronologisch of stringlist alfabetisch	Lijst.sort()
List.sort(reverse=boolean)	Sorteert in omgekeerde volgorde	Lijst.sort(reverse=True)
List.sort(key=a)	Sorteert volgens de functie a	Lijst.sort(key=len)/Lijst.sort(key=str.lower)
Id(list)	Retourneert Id van de list	Id(lijst)
Is	Operator die ID's van lists vergelijkt	Lijst2 is Lijst1
Ord(a)	Geeft de ASCII-nummering van een teken	Ord("Z")
Chr(a)	Geeft het teken van de ASCII-nummering	Chr(55)
<handle> = open(bestandsnaam>)	Opent een bestand met een bepaalde handle om te lezen	Fp = open("test.txt")

open(<bestandsnaam>) as <handle>		
<handle>.read()	Leest het bestand met de handle helemaal uit	Fp.read()
<handle>.close()	Sluit het bestand met de handle	Fp.write()
<handle>.readline()	Leest 1 lijn van het bestand met de handle uit	Fp.readline()
<handle>.readlines()	Retourneert een list met als elementen de lines van bestand met deze handle	F.readlines()
<handle> = open(bestandsnaam> , "w")	Opent een bestand met een bepaalde handle om te schrijven	Fp = open("test.txt", "w")
<handle> = open(bestandsnaam> , "a", encoding="latin-1")	Opent een bestand met een bepaalde handle om te appenden	Fp = open("test.txt", "a")
Os.path.exists(a)	Retourneert True als het pad a bestaat	exists("pc_rose.txt")
Os.path.isfile(a)	Retourneert True als het pad a bestaat en a een bestand is	isfile("pc_rose.txt")
Os.path.isdir(a)	Retourneert True als het pad a bestaat en a een map is	isdir("pc_rose.txt")
Os.path.Join(map, bestand)	Retourneert een path opgebouwd uit de meegegeven parameters	join(getcwd(), naam)
Os.listdir(a)	Retourneert een list met als elementen de bestanden in de map a	Listdir(.)
Os.path.basename(a)	Retourneert de bestandsnaam uit path a	basename("/System/Home/test.txt")
Os.path.Dirname(a)	Retourneert de mapnaam uit path a	dirname("/System/Home/test.txt")
Os.path.Getsize(a)	Retourneert de bestandsgrootte in bytes van a	getsize("/System/Home/test.txt")
set.add(a)	Voegt het element a toe aan de set als het geen duplicaat is	Fruitset.add("appel")
set.update(a) set = a	Voegt de elementen van list a toe aan de set als het geen duplicaat is	fruitset.update(["appel","appel","appel","aardbei", "aardbei","appel","mango"])

Set.remove(a) / set.discard(a)	Verwijdert element a uit de set (discard geeft geen runtime error indien element niet bestaat)	fruitset.remove("appel")
Set.pop()	Verwijdert random element uit de set en retourneert het	fruitset.pop("appel")
Set.union(a) Set a	Retourneert de vereniging van de set met set a	fruit1.union(fruit2)
Set.intersection(a) Set & a	Retourneert de doorsnede van de set met set a	fruit1.intersection(fruit2)
Set.difference(a) Set – a	Retourneert de elementen van de set die niet in a zitten	fruit1.difference(fruit2)
Set.isdisjoint(a)	Retourneert True als geen enkel element van de set in a zit	fruit1.isdisjoint(fruit2)
Set.issubset(a)	Retourneert True als elk element van de set in a zit	fruit1.issubset(fruit2)
Set.issuperset(a)	Retourneert True als elk element van a in de set zit	fruit1.issuperset(fruit2)
Set.symmetric_difference(a) Set ^ a	Retourneert alle elementen van de set en a behalve de doorsnede	fruit1.symmetricdifference(fruit2)
frozenset(a)	Creëert een onveranderbare set met element(en) a	frozenset(["appel", "banaan", "kers"])
Os.getcwd()	Geeft het path van de huidige map	Os.getcwd()
Sys.Getfilesystemencoding()	Geeft de encoding van het besturingssysteem weer	Sys.Getfilesystemencoding()
U = Urllib.request.urlopen(a)	Opent de website met link a met handle U	U = urlopen("http://www.python.org")
Urllib.error.HTTPError Urllib.error.URLError	Op te vangen exceptions bij openen van een webpagina	except HTTPError as e: print("HTTP Error", e) sys.exit() except URLError as e: print("URL error", e) sys.exit()
C = collections.Counter(data) C.most_common() C.update(iterable)	Counter object met bijhorende functies	from collections import Counter data = ["appel", "banaan", "appel", "banaan", "appel", "kers"]

		<pre> c = Counter(data) print(c) print(c.most_common(1)) data2 = ["mango", "kers", "kers", "kers", "kers"] c.update(data2) print(c) print(c.most_common()) </pre>
Dq = collections.deque(list) Append(left)() Extend(left)() Pop(left)()	Deque object met bijkomende functies	<pre> from collections import deque dq = deque([1, 2, 3]) dq.appendleft(4) dq.extendleft([5, 6]) print(dq) </pre>
Glist = glob.glob(regex)	Creëert list met alle filenames die aan de meegegeven regex voldoen	<pre> from glob import glob glist = glob("*.pdf") for name in glist: print(name) </pre>
Glob.iglob(regex)	Creëert iterator voor alle filenames die aan de meegegeven regex voldoen	<pre> for name in glob.iglob("*.pdf"): print(name) </pre>
Statistics.mean() Statistics.median() Statistics.(multi)mode() Statistics.stdev() Statistics.variance()	De statistics module met zijn functies	<pre> from statistics import mean, median, mode, stdev, variance, \ StatisticsError data = [4, 5, 1, 1, 2, 2, 2, 3, 3, 3] print("gemiddelde:", mean(data)) print("mediaan:", median(data)) try: print("modus:", mode(data)) except StatisticsError as e: print(e) print("st.dev.: {:.3f}".format(stdev(data))) print("variantie: {:.3f}".format(variance(data))) </pre>
Statistics.StatisticsError	Op te vangen exception bij het gebruik van statistics-functions, bv. wanneer er geen unieke modus is	<pre> except StatisticsError as e: print(e) </pre>
Tijd =datetime.datetime(year, month, day, uur = 0, minute = 0, second = 0, microsecond = 0, tzinfo = 0)	De datetime module en zijn 2 classes	<pre> ditjaar = datetime.now().year xmasditjaar = datetime(ditjaar, 12, 25, 23, 59, 59) dezedag = datetime.now() dagen = xmasditjaar - dezedag </pre>

Delta = datetime.timedelta(day = 0, second = 0, microsecond = 0)		<pre> if dagen.days < 0: print("Kerst komt volgend jaar weer.") elif dagen.days == 0: print("Het is Kerst!") else: print("Slechts", dagen.days, "dagen tot Kerst!") </pre>
--	--	---

Conditie/iteratie:

- ➔ If (+else/elif)
- ➔ While (+else)
- ➔ For (+else)

Exceptions:

- ZeroDivisionError:
 - Delen door nul
- IndexError
 - Het benaderen van een list of tuple met een index die niet binnen het legale bereik valt
- NameError
 - Onbekende variabelenaam
- KeyError
 - Het benaderen van een dictionary met een key die onbekend is
- IOError
 - Iedere fout die kan optreden als je een bestand benadert (deze exception is een alias voor OSError)
- FileNotFoundError
 - Het proberen te openen van een niet-bestaand bestand om eruit te lezen
- ValueError
 - Het optreden van een fout bij het “casten” van een waarde naar een andere waarde
- TypeError
 - Het gebruiken bij een operatie van een waarde met een data type dat niet ondersteund wordt door de operatie
- errno.ENOENT
 - Dit bestand of deze directory bestaat niet. Dit krijg je als je een bestand benadert dat niet bestaat.
- errno.EACCESS
 - Toestemming geweigerd. Je kunt deze melding in verschillende omstandigheden krijgen, zoals wanneer je probeert te lezen uit een gesloten bestand, of als je in een bestand probeert te schrijven dat voor alleen lezen bedoeld is, of als je een directory probeert te openen alsof het een bestand is.

- `errno.ENOSPC`
 - Onvoldoende ruimte. Je krijgt deze fout als je een bestand probeert te schrijven en er geen ruimte voor het bestand beschikbaar is, bijvoorbeeld als je probeert te schrijven naar een USB-stick die vol is.

Termen:

- Text Editor \leftrightarrow Word Processor
- Pure functies \leftrightarrow Modifiers
- Formele parameter \leftrightarrow Actuele parameter