

CodeTeam

Beginnerscursus Programmeren voor Volwassenen

Concepten Python Codecademy



1 Python Syntax

Output naar de console:

```
print "Dit komt in de console te staan"
```

Variabele aanmaken:

```
mijn_variabele = 10
```

Datatypes:

Naam	Soort data	Voorbeeld
Integer	Gehele getallen	mijn_integer = 55
Float	Decimale getallen	mijn_float = 55.0
String	Tekst	mijn_string = "hallo"
Boolean	Waar of Niet waar	mijn_bool = True

Strings kunnen met 'enkele' en met "dubbele" aanhalingstekens

Comments:

```
# dit is een comment
```

Rekenen:

Operatie	Operator	Voorbeeld
Optellen	+	5 + 5
Aftrekken	-	4 - 2
Vermenigvuldigen	*	5 * 6
Delen	/	10 / 2
Machtsverheffen	**	10 ** 3
Rest berekenen (modulo)	%	22 % 7

Variabelen updaten:

De volgende expressies zijn equivalent:

```
mijn_variabele = mijn_variabele + 5
```

```
mijn_variabele += 5
```

```
mijn_variabele = mijn_variabele - 5
```

```
mijn_variabele -= 5
```

```
mijn_variabele = mijn_variabele * 5
```

```
mijn_variabele *= 5
```

```
mijn_variabele = mijn_variabele / 5
```

```
mijn_variabele /= 5
```

2a Strings and Console output

Variabelen printen naar de console:

```
mijn_string = "hallo!"  
print mijn_string
```

Strings met aanhalingstekens:

```
"deze string heeft een ' erin"      'deze string heeft een " erin'  
'deze string heeft een \' erin'     "deze string heeft een \" erin"
```

String index:

```
"MONTY"[0] geeft "M"
```

String-functies:

```
mijn_string = "Hallo"  
len(mijn_string) geeft 5  
mijn_string.lower() geeft "hallo"  
mijn_string.upper() geeft "HALLO"  
mijn_string + " " + "Pietje" geeft "Hallo Pietje"
```

String-parameters:

```
"Ik heb %s en %s" ("eieren", "wortels") geeft "Ik heb eieren en wortels"
```

Input van de gebruiker:

```
naam = raw_input("Wat is je naam?: ")
```

Veranderen van datatype:

Functie	Resultaat	Voorbeeld
str()	String	str(100) geeft "100"
int()	Integer	int(10.3) geeft 10 int("123") geeft 123
float()	Float	float(3) geeft 3.0

2b Date and time project

Module importeren:

```
from datetime import datetime
```

Huidige datum en tijd:

```
nu = datetime.now()
```

Opbreken in jaren/maanden/dagen/uren/minuten/seconden:

```
nu = datetime.now()  
nu.year  
nu.month  
nu.day  
nu.hour  
nu.minute  
nu.second
```

3a Conditionals and Control Flow

Vergelijkingen:

Operatie	Operator	Voorbeeld
Gelijk aan	<code>==</code>	<code>2 == 2</code>
Ongelijk aan	<code>!=</code>	<code>2 != 3</code>
Groter dan	<code>></code>	<code>3 > 2</code>
Groter of gelijk aan	<code>>=</code>	<code>2 >= 2</code>
Kleiner dan	<code><</code>	<code>2 < 3</code>
Kleiner of gelijk aan	<code><=</code>	<code>3 <= 3</code>

Boolean operaties:

Operator	Voorbeeld
and	True and True
or	True or False
not	not False

Volgorde: not > and > or

Control flow:

```
if 1+1==2:
    print "dit is waar!"
elif 1+1==3:
    print "dit gebeurt nooit!"
elif 1+1==4:
    print "dit dus ook nooit!"
else:
    print "anders niet!"
```

3b PygLatin Project

Isalpha

`mijn_string.isalpha()` geeft True als `mijn_string` alleen letters bevat

`"tekst".isalpha()` geeft True

`"tekst123".isalpha()` geeft False

String slices

`"een stukje tekst"[4:10]` geeft "stukje"

4a Functions

Functies definiëren:

```
def square(n):  
    squared = n ** 2  
    print "%d squared is %d" % (n, squared)  
  
def power(base, exponent):  
    return base ** exponent
```

Floats als parameters in strings:

```
"Ik heb een %f gehaald voor %s!" % (6.0, "wiskunde")
```

Modules importeren:

```
import math (importeert hele module)  
math.sqrt(16) geeft 4
```

```
from math import sqrt (importeert één functie uit de module)  
from math import * (importeert alle functies uit de module)  
sqrt(16) geeft 4
```

Functies met variabele hoeveelheid argumenten:

```
def mijn_grootste_functie(*args):  
    return max(args)
```

```
mijn_grootste_functie(1, 2, 3, 4.0, 5, 6) geeft 6
```

Ingebouwde functies:

Functie	Resultaat	Voorbeeld
min(*args)	Kleinste argument	min(3, 6, 2, 4) geeft 2
max(*args)	Grootste argument	max(3, 6, 2, 4) geeft 6
abs(arg)	Absolute waarde	abs(-5) geeft 5
type(arg)	Datatype argument	type(3.0) geeft float

5a Lists and Dictionaries

Lijsten maken:

```
lege_lijst = []  
lijst_met_strings = ["boter", "kaas", "eieren"]  
lijst_met_ints = [1, 2, 3]  
gemengde_lijst = [1, "boter", 2.0, "kaas", True, "eieren"]
```

Lijsten veranderen:

```
mijn_lijst = [1, 2, 3]  
mijn_lijst[2] = 4 geeft [1, 2, 4]  
mijn_lijst.append(4) geeft [1, 2, 3, 4]  
mijn_lijst.insert(2, 7) geeft [1, 2, 7, 3]  
mijn_lijst.remove(2) geeft [1, 3]
```

Index opvragen:

```
mijn_lijst = ["boter", "kaas", "eieren"]  
mijn_lijst.index("kaas") geeft 1
```

Lengte van een lijst:

```
mijn_lijst = ["boter", "kaas", "eieren"]  
len(mijn_lijst) geeft 3
```

List slices:

```
mijn_lijst = ["eerste", "tweede", "derde", "vierde", "vijfde"]  
mijn_lijst[0:2] geeft ["eerste", "tweede"]  
mijn_lijst[2:4] geeft ["derde", "vierde"]  
mijn_lijst[3:6] geeft ["vierde", "vijfde"]  
mijn_lijst[:] geeft ["eerste", "tweede", "derde", "vierde", "vijfde"]  
mijn_lijst[:3] geeft ["eerste", "tweede", "derde"]  
mijn_lijst[3:] geeft ["vierde", "vijfde"]
```

Lijsten sorteren:

```
mijn_lijst = [1, 4, 2, 5, 3]  
mijn_lijst.sort() geeft [1, 2, 3, 4, 5]
```

Loops op lijsten:

```
mijn_lijst = [1, 2, 3]  
for getal in lijst:  
    print "en dat is " + str(getal)  
geeft:  
en dat is 1  
en dat is 2  
en dat is 3
```

Dictionaries maken:

```
mijn_dict = {"boter" : 2.50, "kaas" : 4.00, "eieren" : 1.99}  
mijn_dict["kaas"] geeft 4.0
```

Dictionaries veranderen:

```
mijn_dict = {"boter" : 2.50, "kaas" : 4.00, "eieren" : 1.99}
```

```
mijn_dict["kaas"] = 4.40 geeft  
{"boter" : 2.50, "kaas" : 4.40, "eieren" : 1.99}
```

```
del mijn_dict["kaas"] geeft {"boter" : 2.50, "eieren" : 1.99}
```

Lengte van een dictionary:

```
mijn_dict = {"boter" : 2.50, "kaas" : 4.00, "eieren" : 1.99}  
len(mijn_dict) geeft 3
```