

Python



Python

Python is een krachtige programmeertaal die makkelijk te leren is en waar je van alles mee kunt maken.

De eerste versie van Python is in 1991 gelanceerd door Guido van Rossem.



Python versies

- Python 3.X
 - Te gebruiken op Windows, Mac, Linux
- MicroPython
 - Beschikbaar op diverse developerboards
 - Voor ons is dit de Raspberry Pi Pico



Waarom Python

- Makkelijk te leren. ...
- Werkt bijna overal. ...
- Het is een hele populaire taal. ...
- A.I. ...
- Zo veel (hulp)programma's. ...
- Je kan het (bijna) overal voor gebruiken. ...

Python is

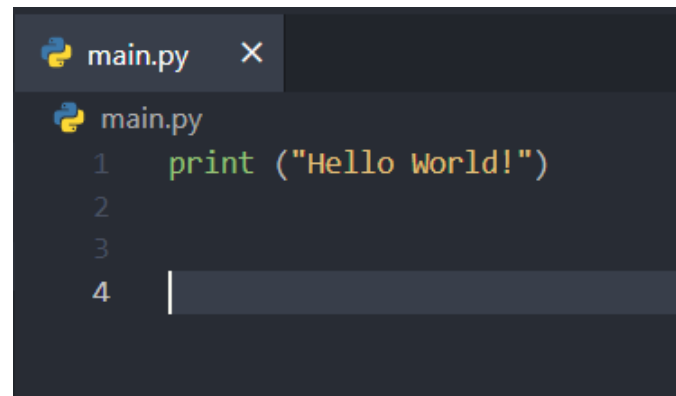
- Case-sensitive (variabele = 0 is niet hetzelfde als Variabele = 0)
- Werkt met Indentatie om de diverse blokken aan te geven
- Inmiddels bij versie 3.x (Gebruik 2.x niet meer)
- Veel gebruikt bij data science (Anaconda)

Installeren.....

- Python (<https://www.python.org/downloads/>)
- Text editor
 - Visual studio code (<https://code.visualstudio.com/download>)
 - Thonny (<https://thonny.org/>)
 - Notepad++ (<https://notepad-plus-plus.org/downloads/>)
- Anaconda (optioneel)
(<https://www.anaconda.com/products/individual>)

Hello World!

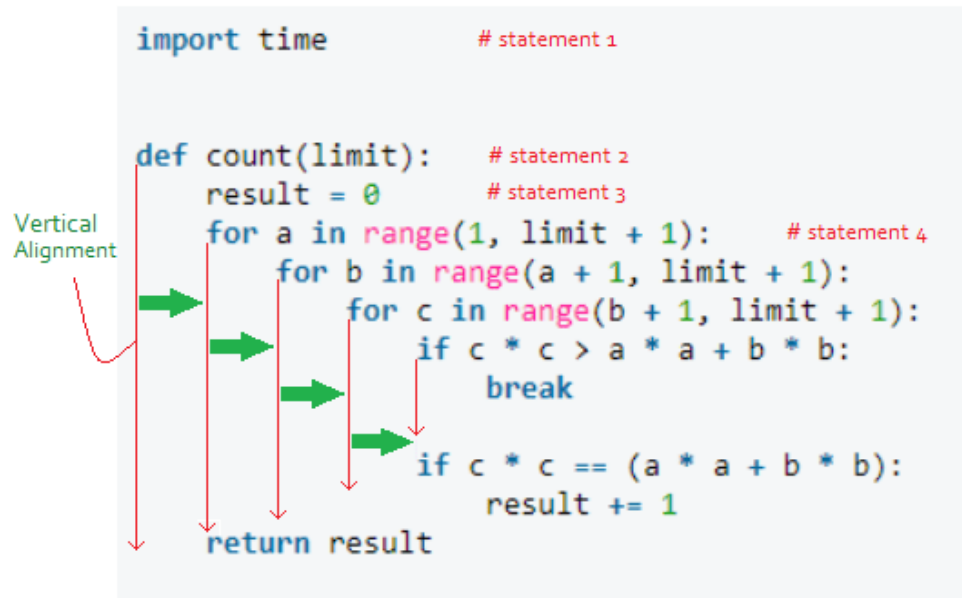
- Natuurlijk starten we met „Hello World“
 - Start je editor
 - Maak een nieuw bestand aan bijvoorbeeld main.py
 - Zet in dit bestand: `print("Hello world")`
 - Start het programma (meestal F5)
- DEMO



```
main.py x
main.py
1 print ("Hello World!")
2
3
4 |
```

Python Indentation

- Een codeblock begint altijd met `:`
 - Daarna altijd een indentation met 4 spaties of TAB



The diagram shows a Python code block with four levels of indentation. Red vertical arrows point from the start of each line to a common vertical line on the left, labeled "Vertical Alignment". Green horizontal arrows point from the left towards the start of each line, showing the increasing indentation levels. The code is as follows:

```
import time # statement 1

def count(limit): # statement 2
    result = 0 # statement 3
    for a in range(1, limit + 1): # statement 4
        for b in range(a + 1, limit + 1):
            for c in range(b + 1, limit + 1):
                if c * c > a * a + b * b:
                    break
            if c * c == (a * a + b * b):
                result += 1
    return result
```


Python commentaar in je code

```
#Dit is een commentaar  
print("Hello, World!")
```

```
""  
  
Dit is een  
multiline  
commentaar  
""  
  
print("Hello, World!")
```



Python Variables

```
x = 5
```

```
y = "John"
```

```
print(x)
```

```
print(y)
```

```
boolean = True / False
```

```
integer = 10
```

```
float = 10.01
```

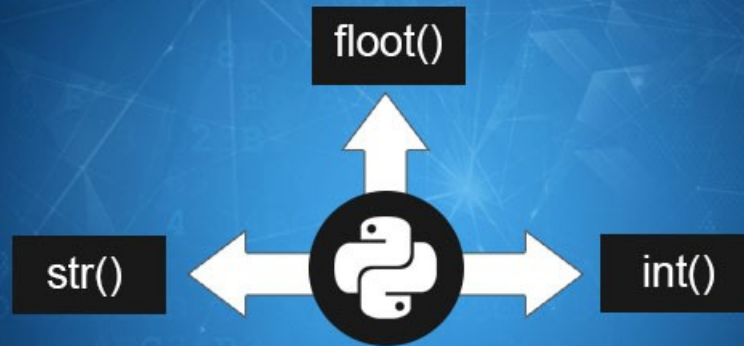
```
string = "123abc"
```

```
list = [ value1, value2, ... ]
```

```
dictionary = { key1:value1, key2:value2, ... }
```

Python type casting


Type Casting in Python



Python Variables

Data Types in Python

Some compound data types



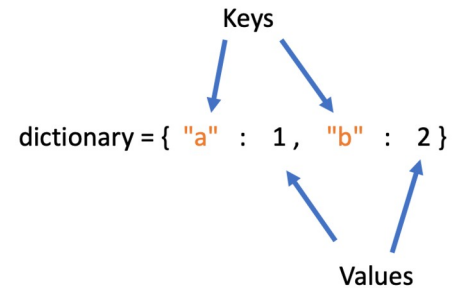
delimiters

["A", "C", "G", "T"] ← list

("A", "C", "G", "T") ← tuple

{ "A": "T", "C": "G",
 "G": "C", "T": "A" } ← dict

A tuple is essentially an immutable list
whereas a dict is like a hash



Python gebruikersinvoer

```
x = input('Enter your name:')  
print('Hello, ' + x)
```

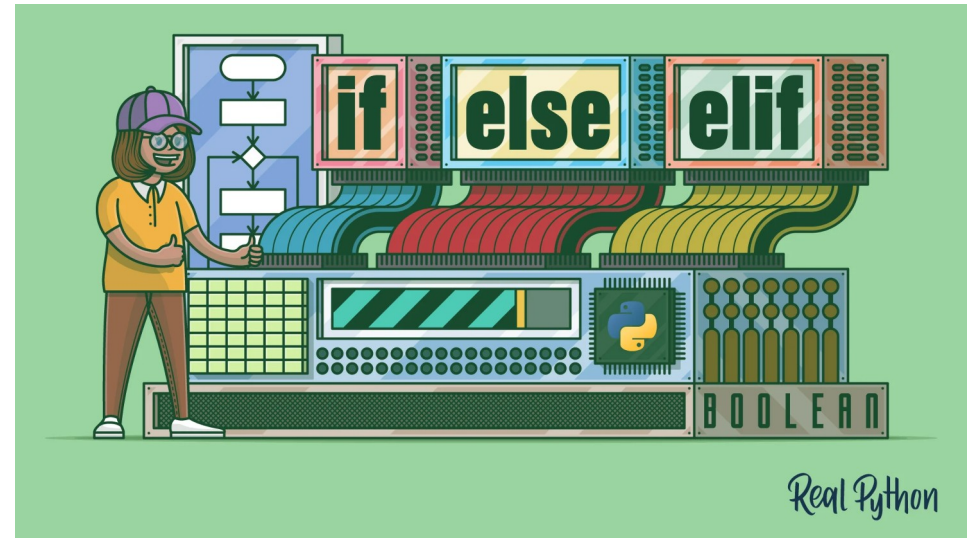
Python functions

```
def volume_kubus(a):  
    return a * a * a  
  
print(volume_kubus(8))
```

The diagram illustrates the components of a Python function definition using the example: `def add(x, y):`
 `print(f'arguments are {x} and {y}')`
 `return x + y`
Annotations with red arrows point to specific parts of the code:
1. `def` keyword
2. `add` function name
3. `(x, y)` function arguments inside parentheses
4. `:` colon ends the function definition
5. The block of code inside the function (from `print` to `return`) is labeled as function code.
6. `return x + y` is labeled as the function return statement.

If then else

```
a = 400
b = 93
if b > a:
    print("b is greater than a")
elif a == b:
    print("a and b are equal")
else:
    print("a is greater than b")
```



Denk ook aan: == <= >= !=

While

Python while True



tutorial.eyehunts.com

```
while True:  
    progress = do_something()  
    if progress is done:  
        break.
```


Opdracht

Een raadspel programmeren

In deze eerste Python app moet de gebruiker een geheim getal raden. De programmeur heeft in dit voorbeeld al een geheim getal ingegeven.

Mogelijke oplossing

```
Geheim_getal = 4
```

```
gok = 0
```

```
while True:
```

```
    gok=input("Raad het getal: ")
```

```
    if int(gok) == Geheim_getal:
```

```
        print("Gefeliciteerd! het geheime getal was "+gok)
```

```
        break
```

```
    else:
```

```
        print("Helaas, dat was niet het geheime getal. Probeer het nog eens.")
```

Python 3 Beginner's Reference Cheat Sheet

Alvaro Sebastian
<http://www.sixthresearcher.com>

Main data types

boolean = `True / False`
integer = `10`
float = `10.01`
string = `"123abc"`
list = `[value1, value2, ...]`
dictionary = `{ key1:value1, key2:value2, ... }`

Numeric operators

+ addition
- subtraction
***** multiplication
/ division
****** exponent
% modulus
// floor division

Comparison operators

== equal
!= different
> higher
< lower
>= higher or equal
<= lower or equal

Boolean operators

and logical AND
or logical OR
not logical NOT

Special characters

coment
\n new line
\<char> scape char

String operations

string[i] retrieves character at position `i`
string[-1] retrieves last character
string[i:j] retrieves characters in range `i` to `j`

List operations

list = [] defines an empty list
list[i] = x stores `x` with index `i`
list[i] retrieves the item with index `i`
list[-1] retrieves last item
list[i:j] retrieves items in the range `i` to `j`
del list[i] removes the item with index `i`

Dictionary operations

dict = {} defines an empty dictionary
dict[k] = x stores `x` associated to key `k`
dict[k] retrieves the item with key `k`
del dict[k] removes the item with key `k`

String methods

string.upper() converts to uppercase
string.lower() converts to lowercase
string.count(x) counts how many times `x` appears
string.find(x) position of the `x` first occurrence
string.replace(x,y) replaces `x` for `y`
string.strip(x) returns a list of values delimited by `x`
string.join(L) returns a string with `L` values joined by string
string.format(x) returns a string that includes formatted `x`

List methods

list.append(x) adds `x` to the end of the list
list.extend(L) appends `L` to the end of the list
list.insert(i,x) inserts `x` at `i` position
list.remove(x) removes the first list item whose value is `x`
list.pop(i) removes the item at position `i` and returns its value
list.clear() removes all items from the list
list.index(x) returns a list of values delimited by `x`
list.count(x) returns a string with list values joined by `S`
list.sort() sorts list items
list.reverse() reverses list elements
list.copy() returns a copy of the list

Dictionary methods

dict.keys() returns a list of keys
dict.values() returns a list of values
dict.items() returns a list of pairs (key,value)
dict.get(k) returns the value associated to the key `k`
dict.pop() removes the item associated to the key and returns its value
dict.update(D) adds keys-values (`D`) to dictionary
dict.clear() removes all keys-values from the dictionary
dict.copy() returns a copy of the dictionary

Legend: `x,y` stand for any kind of data values, `s` for a string, `n` for a number, `L` for a list where `i,j` are list indexes, `D` stands for a dictionary and `k` is a dictionary key.