

Files, Lists en Tuples



BIN-OWE1

Studiewijzer

Les	Onderwerp	
	Algemeen	Python
1	Linux	
2	Git	Input/output
3	Pseudocode	If/elif/else Booleans
4		For loop
5		Lists and tuples Files (CSV bestanden)
6	Flowcharts	Functies
7		Strings CSV bestanden

Studiemateriaal

- Boek: “Starting Out with Python, third edition”
- Reader: “Linux voor Bio-informatici”
- Onderwijs Online
- Kom je er niet uit? → Google



Doelstellingen

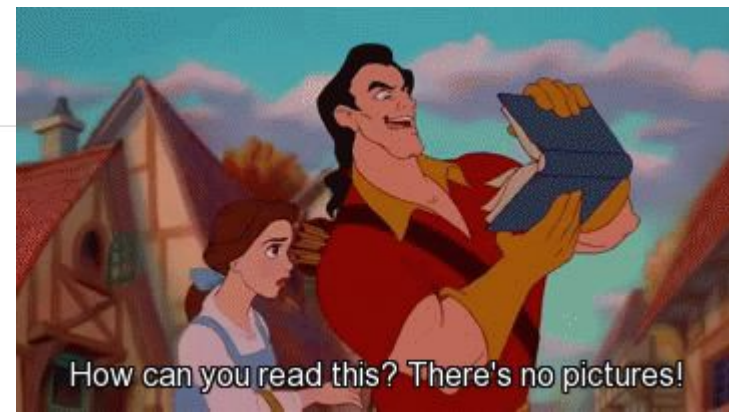
- Aan het eind van deze week kan je
 - Bestanden openen in Python en deze parsen
 - Lijsten uitlezen en vullen
 - Tuples aanmaken
- Aan het eind van deze week begrijp je
 - Het verschil tussen lists en tuples
 - Hoe je een bestand kunt inlezen met behulp van Python

Inhoud

- Openen van bestanden
- Lezen
- Schrijven
- Aanvullen

Introductie

- Python leesbaar
 - Bestanden met tekstuele informatie, zogenaamde platte tekst
 - ASCII of Unicode
- Binary files
 - Bestanden met binaire informatie
 - Plaatjes, muziek, video
 - Maar ook .docx en .odt, .xlsx, etc.
 - En ook python datastructuren, maar daarover later meer...
 - Niet standaard Python leesbaar, maar zijn opties voor

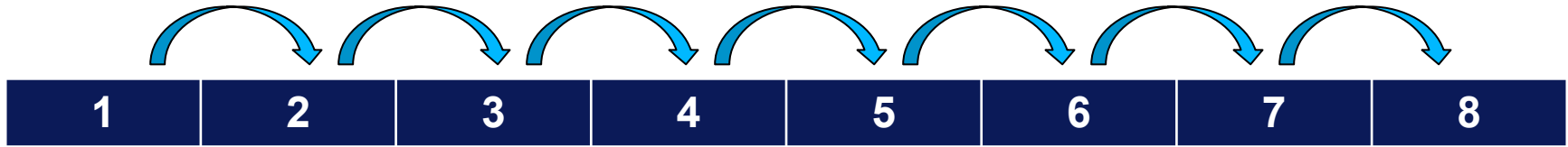


Bestands toegang

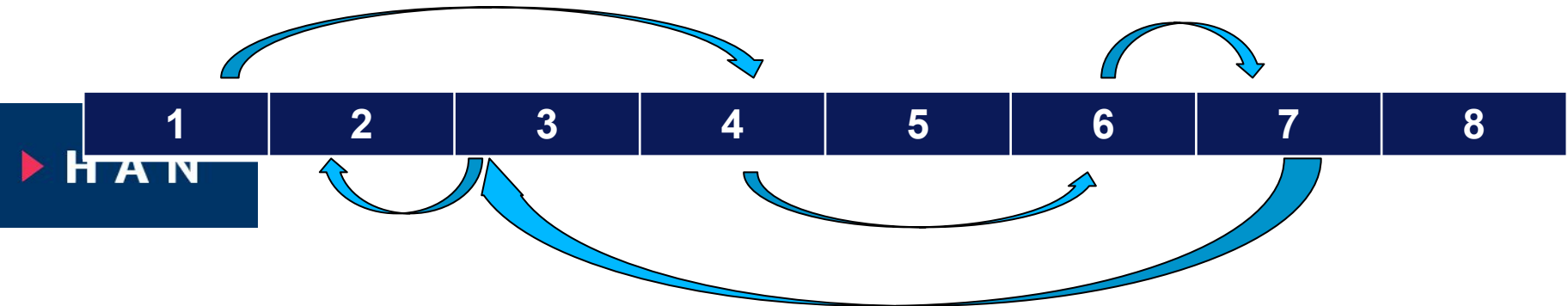
- Sequentieel
 - Iedere byte wordt achter elkaar gelezen
- Direct
 - Iedere byte is op elk willekeurig moment toegankelijk

Sequentieel vs Direct

Sequentieel Access: muziek op de radio



Direct Access: muziek in je mp3 lijst

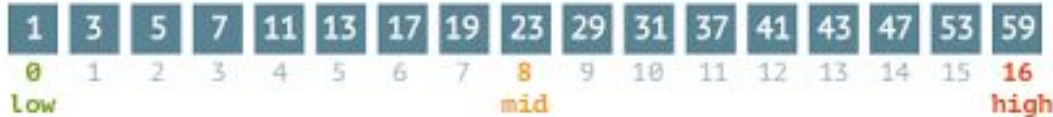


Voorbeeld met een zoekalgoritme

Binary search

steps: 0

37



Sequential search

steps: 0

37



Stappen

1. Open bestand
2. Lees bestand
3. Sluit bestand



Bestand Openen

```
file_var = open(filename, mode)
```

- `file_var` is de naam die je de variabele geeft
- `filename` is de volledige naam van het bestand, als string
- `mode` is de wijze van toegang
 - `r` is lezen
 - `w` is schrijven
 - `a` is aanvullen

Sluiten van een bestand

```
file.close()
```

- Dit doe je om het bestand weer vrij te geven voor de buitenwereld
- Dit gebeurt ook impliciet op het moment dat het programma afgelopen is, maar is wel netjes om te doen tijdens het draaien van het programma

Inhoud

- Openen van bestanden
- Lezen
- Schrijven
- Aanvullen

Bestand lezen

```
bestand = open ("demo.txt", "r")  
regel = bestand.readline ()  
print regel
```

Methodes om bestanden te lezen

```
file.read()
```

```
file.readline()
```

```
file.readlines()
```

- Wat zou de 'zuinigste' manier zijn?
- Geen methode, maar wel een manier:
 - Looping (for of while)

Extra String Methodes

`.split()`

`.replace()`

`.strip()`

Deze zijn heel handig om bestanden te parsen

Inhoud

- Openen van bestanden
- Lezen
- **Schrijven**
- Aanvullen

Schrijven van een bestand

```
bestand = open("demo.txt", "w")  
bestand.write("Hello World!")  
bestand.close()
```

Write modus overschrijft de inhoud van het bestand. Indien het bestand niet bestaat maakt hij een nieuwe aan.

Noem een voorbeeld van een proces waarvoor schrijven heel handig zou zijn

Inhoud

- Openen van bestanden
- Lezen
- Schrijven
- **Aanvullen**

Inhoud

```
bestand = open("demo.txt", "a")  
bestand.write("Hello World!")  
bestand.close()
```

Append modus overschrijft de inhoud van het bestand niet, maar vult het aan.

Bedenk een voorbeeld van een proces waarbij appenden heel handig zou zijn.

Samenvatting

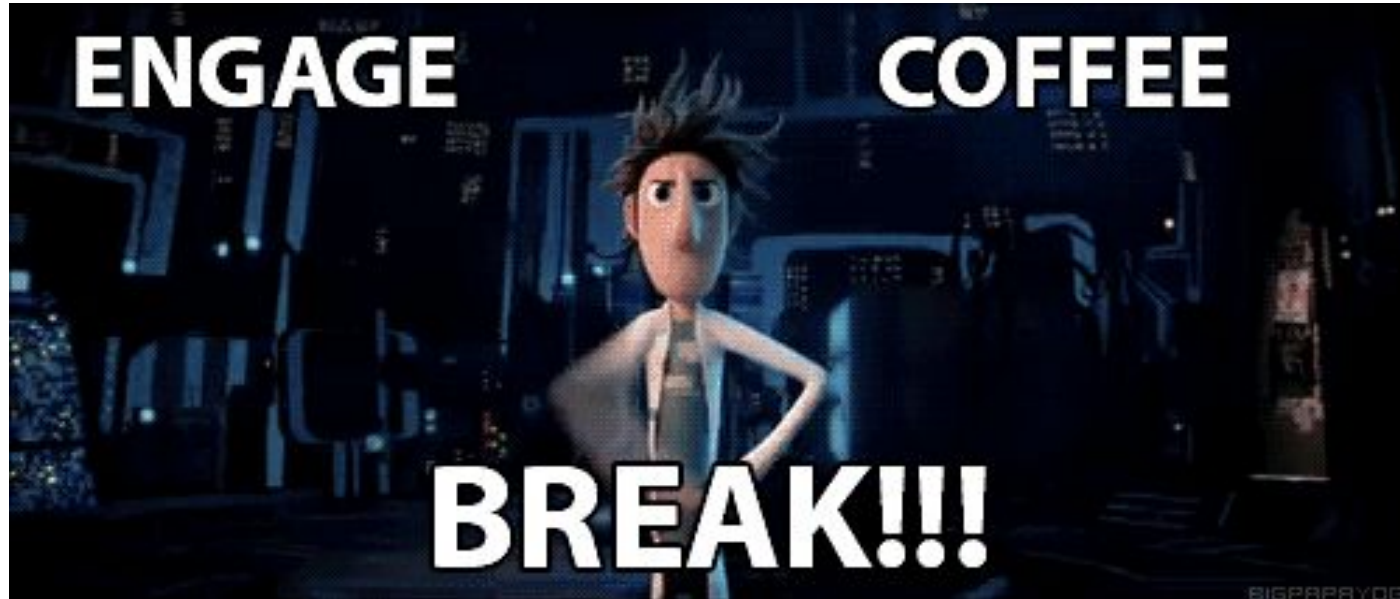
- Gebruik `open()` om een bestand te openen
- Gebruik `file.close()` om een bestand te sluiten
- Het lezen (`read`) van een bestand kan op verschillende manieren, ieder met zijn eigen doel voor ogen
- Schrijven (`write`) overschrijft de volledige inhoud van het bestand
- Aanvullen (`append`) vult vanaf het einde van het bestand aan wat je schrijft.
- We kennen ook `a/w + modus`, waarbij je ook mag lezen.
- Lees de documentatie eens na wat een file object allemaal mag:

<https://docs.python.org/2/library/stdtypes.html#builtin-file-objects>

Opdracht

- Ga naar OnderwijsOnline
- Maak Afvinkopdracht 5
- Bij problemen:
 - Google
 - Klasgenoten
 - Docent
- Lever deze in bij de praktijkdocent





Verantwoording

- In deze uitgave is géén auteursrechtelijk beschermd werk opgenomen
- Alle teksten © Martijn van der Bruggen/Esther Kok/HAN tenzij expliciet externe bronnen zijn aangegeven
- Screenshots op basis van eigen werk auteur en/of vernoemde sites en/of fair use
- Eventuele images zijn opgenomen met vermelding van bron