



Week 4A

Doelstellingen

Kunnen werken met Logging mechanisme aanwezig in Python
Kunnen werken met het threading mechanisme in Python
In staat zijn om network programming in Python te implementeren

Afspraken

Alle programmeeroefeningen van de verschillende labo's dienen op Github geplaatst te worden. Om jouw oefeningen eenvoudig te kunnen vragen we om per week een afzonderlijke map aan te maken.

Oefeningen

Oef 1: we laten twee (of meerdere threads) *samen* 20 willekeurig verschillende getallen selecteren. We houden het resultaat continue in een dictionary bij waarin elke thread zijn eigen list aanmaakt (indien nog niet aanwezig) en daarin zijn gegenereerde getallen bijhoudt.

Voorbeeld:

```
{'Thread-1': [27, 30, 86, 0, 18, 5, 59, 74, 68, 78, 81, 34], 'Thread-2': [60, 55, 7, 100, 90, 22, 20, 63]}
```

Maak gebruik van logging-mechanisme om op elk moment te kunnen opvolgen welk getal door wie gegenereerd wordt.

Integreer het lock-mechanisme zodat geen enkele thread gelijktijdig de dictionary kan bewerken.

Nadat een thread een getal gekozen en aan zijn list toegevoegd heeft, wordt hij voor een willekeurige tijd in slaap gelegd: tussen 0 en 1 sec.

Print nadien het volledige resultaat af. Voorbeeld:

```
INFO:root:Thread-1 - Number selected: 27  
INFO:root:Thread-2 - Number selected: 60  
INFO:root:Thread-1 - Number selected: 30...
```

```
{'Thread-1': [27, 30, 86, 0, 18, 5, 59, 74, 68, 78, 81, 34], 'Thread-2': [60, 55, 7, 100, 90, 22, 20, 63]}  
Thread-1 -> numbers selected: 12  
Thread-2 -> numbers selected: 8
```

Oef 2: bestudeer de gegeven code. De methode ping_website gaat na of een website down is (status_code >= 400) of als er een timeout plaats vindt. In beide gevallen wordt een exception terug gegeven.

Gegeven een lijst met te controleren websites.

Gegeven: de seriële benadering gaat elke website één voor één controleren. De nodige tijd wordt hiervoor geregistreerd en afgeprint.

Maak een alternatieve benadering door gebruik te maken van

- queue (<https://docs.python.org/3/library/queue.html>). Vul de queue op met alle gegeven websites.
- threading: 4 threads die via de queue de verschillende websites gaan controleren.

Via de methodes get() kan een volgend item uit de queue opgehaald worden, via task_done() geef je aan dat het item verwerkt is.

Via de methode join() op de queue wacht de verdere uitvoering totdat de queue leeg is (en dus de aanwezige items verwerkt zijn).

Meet ook hier de tijd op. Vergelijk met de seriële benadering.



Oef 3: bestudeer de gegeven code. Ga na hoeveel tijd je nodig hebt om een testmethode (versie1 en versie2) ofwel seriëel in één proces, ofwel via threads, ofwel via verschillende afzonderlijke processen te laten uitvoeren.