# week 1/2

Deze opdracht geeft je oefening met Input/Output in verwerken in Python. Maak de volgende opdrachten uit het boek, Hoofdstuk 2, Programming Exercises:

1. (1) Personal Information
2. (2) Sales Prediction
3. (4) Total Purchase
4. (5) Distance Traveled
5. (7) Miles per Gallon
6. (10) Ingredient Adjuster

Maak de volgende oefeningen uit het boek.

Opdracht: Programming excersises Hoofdstuk 3

1. 1. Number Analyser
2. 2. Areas of Rectangles
3. 9. Roulette Wheel Colors
4. 17. WI-Fi Diagnostic Tree
5. 18. Restaurant Selector
6. - Schrijf een python script dat het aantal nucleotiden van een sequentie kan tellen.
7. - Schrijf een python script dat het gc% van een sequentie kan berekenen.

# week 3

Er bestaan erg veel verschillende restrictie enzymen. Hiermee kunnen we een methode opzetten om DNA te analyseren. Schrijf een script dat op basis van de invoer van de naam van een restrictie enzym bepaalt of er een knip gezet wordt in de DNA reeks gegeven in het bestand startOpgave3.py, deze kun je vinden als je de github classroom assignment accepteert. Je mag in dit bestand in de github repo verder werken.

Restrictie enzymen worden ingezet voor de diagnose van genetische defecten maar zijn ook bijvoorbeeld veel gebruikt in [forensisch onderzoek](https://sciencing.com/restriction-enzymes-used-forensic-science-5724559.html).

1. Accepteer de opdracht in de classroom ([link](https://classroom.github.com/a/7wEHqk2U))map start opgave 3
2. Bekijk de code en maak een bijbehorende flowchart uit in. Not really usefull for u
3. Werk het script verder uit. Doe dit voor alle restrictie enzymen gegeven in het bestand.
4. Voorzie je bestand van commentaar en schrijf deze weg als Afvinkopdracht3.py.
5. Draai het script in dezelfde map als startOpgave3.txt, anders kan het script het bestand niet vinden. Of geef de precieze bestandslocatie door aan python.
6. Push het geheel naar de opdracht repo.
7. Optioneel: Geef per restrictie enzym op welke positie(s) van het DNA hij geknipt heeft. Dit is een moeilijke opgave!

Schrijf een python script dat:

* toetst of de ingevoerde sequentie DNA kan zijn
* toetst of de ingevoerde sequentie eiwit kan zijn
* de lengte van een sequentie bepaald.
* de lengte van een eiwit sequentie en van de bijbehorende mRNA sequentie met elkaar vergelijkt.
* het aantal aminozuren D, E, R, K in de eiwitsequentie telt.

# Week 4

Maak de volgende Programming Exercises uit het boek (Hoofdstuk 4):

1. (1) Bug Collector
2. (3) Lap Times
3. (12) Calculating the Factorial of a Number

Hieronder vind je de sequenties van een sikkelcel-gen met die mutatie en van het normale gen zonder de mutatie. Als blijkt dat die mutatie resulteert in het verdwijnen of juist verschijnen van een restrictiesite, kan voor toekomstige prenatale diagnostiek hiervan gebruik worden gemaakt. Men hoeft dan alleen nog maar het betreffende gebied te vermeerderen m.b.v. PCR, gevolgd door een knip met dit restrictie-enzym en analyse op een agarose-gel. Kortom: je opdracht is om een aantal restrictie enzymen te testen tegen onderstaande sequenties. Zoek uit of een van de restrictie enzymen geschikt is om een onderscheid te maken tussen beide sequenties.

**Sikkelcel gen**

GAGCCATCTATTGCTTACATTTGCTTCTGACACAACTGTGTTCACTAGCAACCTCAAACAGA

CACC**ATG**GTGCACCTGACTCCTG**T**GGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAAGGTGA

ACGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAGGTTGGTATCAAGGTTACAAGACAGGTTT

AAGGAGACCAATAGAAACTGGGCATGTGGAGACAGAGAAGACTCTTGGGTTTCT

**Normaal gen**

GAGCCATCTATTGCTTACATTTGCTTCTGACACAACTGTGTTCACTAGCAACCTCAAACAGA

CACC**ATG**GTGCACCTGACTCCTG**A**GGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAAGGTGA

ACGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAGGTTGGTATCAAGGTTACAAGACAGGTTT

AAGGAGACCAATAGAAACTGGGCATGTGGAGACAGAGAAGACTCTTGGGTTTCT

**Stappenplan**

1. Initieer een nieuwe repo op je laptop
2. Accepteer de opdracht op GitHub Classroom ([link](https://classroom.github.com/a/VY7-Tyii)). Hierin vind je net als de vorige opdracht de bestanden die je nodig hebt. Start opgave 4
3. Pull de repo.
4. Open het script startOpgave4.py
5. Creëer een script dat door alle enzymen loopt uit het bestand enzymen.txt
6. Vergelijk de restrictiepatronen van beide sequenties. Welk restrictie-enzym is geschikt voor bovengenoemde diagnostiek?
7. Commit tussentijdse veranderingen
8. Voorzie je bestand van commentaar
9. Push de repo terug naar de remote en laat je praktijkdocent weten als je klaar bent
10. Optioneel: Geef van beide sequenties het stuk coderend DNA op (ofwel, het stuk dat begint met ATG en eindigd bij een stopcodon). Zorg dat dit in-frame is.

- Bereken met een python script het molecuulgewicht (MW, eenheid = Da) van de 5 eiwitsequenties (het gekozen eiwit in 5 organismen) aan de hand van de molecuul gewichten van de verschillende aminozuren.

# Week 5

Maak uit hoofdstuk 5 de volgende opgaven:

1. Kilometer Converter (1)
2. Test Average and Grade (15)
3. Prime Numbers (17)
4. Random Number Guessing Game (20)
5. Rock, Paper, Scissors Game (21)

Vergeet bij deze opgaven niet dat je alles in functie vorm moet zetten. Je moet dus gaan nadenken over hoe je de verschillende onderdelen van het probleem uit elkaar haalt om ieder van deze in zijn eigen functie te zetten. Maak er meer als je het gevoel hebt meer oefening nodig te hebben.

Functies zijn erg handig om te kunnen hergebruiken. Deze week ga je aan de slag met verschillende functies die handig zijn om te kunnen herhalen. Een functie om een fasta bestand ([lamaseq.fasta](https://onderwijsonline.han.nl/manage/content/lessonfile/QN4pn4NZ/eyJpdiI6IjFMR3Z4SFRZVFZMTXIyaGdmSkt0aWc9PSIsInZhbHVlIjoiYmJuaEJcL095VDBYa0lXZFdNb01TT3hVbUk0YXppSElkZDBKaEpIT29HYUk9IiwibWFjIjoiZTIyMWRmYjEwZDAxMjYwNTUyNzc1NDg2YWY5NDQ4ODI5ZjgzMzE4YzhjMzliMmI0MDdiNjFkZjkxMDYxODliMSJ9)) in te lezen, een controle voor dna en een functie die controleert of een restrictie enzym in de sequentie knipt. In je main zorg je voor nette relevante output.

Het programma bevat de volgende functies:

* lees\_inhoud(): Een functie die de header en de sequentie retourneert. Als bonus mag je de functie aanpassen zodat deze het fasta bestand inleest en deze scheidt op header en sequentie. Het resultaat zijn twee strings. De functie retourneert dus de header en de sequentie.
* is\_dna(): Een functie die controleert of alle nucleotiden werkelijk in DNA voorkomen. Deze functie retourneert een boolean die indiceert of alle letters bestaan uit ATGC.
* knipt(): Een functie die de restrictiesite van een restrictie enzym en een sequentie accepteert. Controleert of dit restrictie-enzym de sequentie knipt en geeft relevante output (return naar main).
* main(): Deze functie roept alle andere functies aan en print relevante output.

Maak een python script dat de aminozuren van twee sequenties per positie met elkaar vergelijkt en vervolgens het percentage identity berekent. Met andere woorden vergelijk per positie of het aminozuur hetzelfde is.

# Week 6

Maak de volgende Programming Excercises uit het boek, hoofdstuk 7:

1. Lottery Number Generator (2)
2. Number Analysis Program (4)
3. Dice Rolling Function (6)

Maak de volgende Programming Excercises uit het boek, hoofdstuk 6:

1. File Head Display (2)
2. Sum of Numbers (5)
3. Word List File Writer (7)

De bedoeling van de afvinkopdracht is dat jullie, met behulp van wat je van de weektaak geleerd hebt over alignments, aan de slag gaan met het vizualiseren van een alignment. Volg hierbij de volgende stappen:

1. Accepteer de opdracht op GitHub Classroom ([link](https://classroom.github.com/a/4dzolScA)).(start opgave 6)
2. Initialiseer een repo in een nieuwe folder, koppel de remote en pull de inhoud van de remote.
3. Ga uit van het bestand enzymen.txt, bijgevoegd in de remote repo.
4. Maak een nieuw Python script die de volgende functies bevat:
   1. lees\_inhoud(): leest het bestand in met de restrictie enzymen en zet deze in een lijst en retourneer deze terug naar de main.
   2. match(): deze accepteert een sequentie en enzymenlijst als parameters en checkt met behulp van een for-loop of de enzymen knippen in de gegeven sequentie.
5. De bedoeling is dat de visualisering toont op het moment dat er een match gevonden is. Voel je vrij om hiervoor de theorie van de weektaken te gebruiken, om bijvoorbeeld een deel match ook weer te geven.
6. Maak tussendoor voldoende commits en push je uitwerking naar de remote als je klaar bent.

Maak gebruik van de volgende sequentie voor de vergelijking:

ACTAGCAACCTCAAACAGACACCATGGTGCACCTGACTCCTGTGGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAAGGTGAACGTGGATGAAGTTGGTGGTGAGGCC

De uiteindelijke output kan er ongeveer zo uitzien:

--------------------------------------------------------------------------------

match met ApaLI op positie 26

ACTAGCAACCTCAAACAGACACCATGGTGCACCTGACTCCTGTGGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAAGGTGAACGTGGATGAAGTTGGTGGTGAGGCC

GTGCAC

Verbeter de scripts van de weektaken door gebruik te maken van functies

# Week 7

Voor diegene die het gevoel hebben dat ze bepaalde onderdelen nog niet voldoende in de vingers hebben zijn hier een aantal oefenopgaven op basis van thema. Deze oefeningen kan je uiteraard ook gebruiken in voorbereiding op de toets.

1. **Variabelen (H2)**

* Maak van de volgende types een variabele aan en print per type de inhoud van de variabele en het type.
  + Boolean
  + Float
  + String
  + Integer
* Opgaven uit het boek.
  + Male and Female Percentages (11)
  + Tip, Tax, and Total (8)

1. **Beslisstructuren (H3)**

* Opgaven uit het boek.
  + Color Mixer (7)
  + Hot Dog Cookout Calculator (8)
  + Time Calculator (15)

1. **Herhaalstructuren (H4)**

* Schrijf een for-loop die op zoek gaat naar een start-codon in een sequentie op basis van indexen.
  + Check dus [0:3] voor het eerste codon en schuif je index daarna iedere keer 1 positie op.
  + Maak gebruik van range().
  + Als het codon een start codon is, print dan zinvolle informatie voor de gebruiker, bijvoorbeeld de positie waar deze gevonden is.
  + N.B. Omdat dit een oefening herhaalstructuren is mag je geen gebruik maken van .index() o.i.d.
* Opgaven uit het boek
  + Draw patterns (13 & 14)

1. **Lijsten (H7)**

* Opgaven uit het boek.
  + Driver's Licence Exam (7)
  + Lo Shu Magic Square  (11)

1. **Functies (H5)**

* Schrijf een functie die in staat is om een sequentie te accepteren en als resultaat een lijst teruggeeft
  + Gebruik een return voor de lijst
  + In de lijst staat [aantalA, aantalT, aantalC, aantalG]
  + Print het resultaat van de functie in een nette output buiten de functie om
* Schrijf een functie die een sequentie accepteert en bepaald op basis van aantal ATCG of het om een DNA sequentie gaat.
  + Retourneer een boolean
  + Print buiten de functie om of het DNA betreft of niet
* Opgaven uit het boek.
  + Maximum of Two Values (12)
  + Paint Job Estimator (8)

We gaan op basis van een CSV bestand een tweetal kleine dingen berekenen, die we ook met Excel zouden kunnen uitrekenen. Het CSV bestand bevat gen expressie metingen. Het doel van deze opdracht is niet dat je de data begrijpt, maar dat je leert werken met CSV bestanden. Het geeft dus niet als je niet snapt wat al die getallen betekenen.

Volg de volgende stappen:

1. Maak een map aan op je computer waar je in wilt gaan werken.
2. Accepteer de opdracht op GitHub ([link](https://classroom.github.com/a/3F7dTZC6)) en pull de bestaande repo met daarin het bestand dat je mag gaan analyseren.
3. Open het bestand eerst eens in Excel (of Libre Office Spreadsheet). Merk het volgende op:
   * Kolom A bevat een gen ID, ofwel, welk gen er gemeten wordt
   * De eerste rij bevat kolomnamen voor ieder verschillend meetpunt, zoals IFFABF, SASSCC, etc, maar niet voor kolom A.
4. Open het bestand daarna in een text editor zoals kladblok of gedit. Merk ook hier op:
   * De informatie is hetzelfde, alleen nu niet netjes onder elkaar gestructureert, maar gescheiden op komma's.
5. Maak een Python programma aan dat de volgende functies heeft:
   * Accepteert een string bestandsnaam en leest het bestand in. Retourneert de informatie uit de CSV in een 2D lijst.
     + Gebruik de code van de les over herhaling (week 7) als je even niet meer weet hoe het ook alweer moet.
     + Hou er rekening mee dat alleen de gegevens uit kolom A een string behoren te zijn, de rest zijn floats.
     + Als je een test print wilt doen, print dan alleen de eerste 10 rijen om te testen of het gelukt is.
   * Accepteert een lijst en berekent het gemiddelde over kolom B. Retourneert dit gemiddelde.
     + Je kan er dus voor kiezen om alleen kolom B aan de functie te voeren, of het gehele CSV en binnen de functie de extractie van de gegevens van kolom B te doen.
   * Accepteert een lijst en filtert alle ID's waarvoor de waarde van kolom B groter of gelijk is aan 50. Retourneert deze ID's in een lijst.
     + Print de eerste 10 gen ID's als resultaat voor de gebruiker.
6. Push je uitwerking naar de remote en laat je praktijkdocent weten dat je klaar bent. Maak per functie een commit.