

Docentenhandleiding

bij

Crash course programmeren in Python



ISBN 978 90 5905 674 9

 *Visual Steps™*
www.visualsteps.nl

1. Introductie

De docentenhandleiding gaat uit van een lessenschema dat bestaat uit negen punten. Deze punten zijn in drie groepen verdeeld. In principe ziet een les er als volgt uit:

- A Inleiding**
1. Zorgen dat de studenten aandacht hebben voor de les.
 2. Uitleggen wat de leerdoelen van deze les zijn.
 3. Noodzakelijke voorkennis in herinnering brengen.
- B Instructie**
1. De informatie (leerstof) aanbieden.
 2. De studenten helpen bij het leren.
 3. Studenten de gevraagde handelingen laten tonen.
 4. Terugkoppeling geven over de getoonde prestaties.
- C Afsluiting**
1. Het eindresultaat vaststellen van het leren (toetsing).
 2. Herhaling en wijzen op toepassingen.

Het schema wordt van begin tot eind stap voor stap doorlopen.

De stappen B1 tot B4 worden steeds herhaald, dat wil zeggen dat de leerstof steeds in kleine delen wordt aangeboden (B1).

Er zullen dus vele momenten zijn waarop de leerling hulp nodig kan hebben (B2).

Daaruit volgt ook dat tijdens een les studenten meerdere handelingen aanleren die ze kunnen demonstreren (B3), waarop de docent kan reageren met goedkeuring of eventueel noodzakelijke corrigerende opmerkingen (B4).

Op de volgende pagina vindt u nogmaals het **schema** aangevuld met extra informatie om de inhoud en het waarom van elke stap te verduidelijken. De uitwerking van de lessen wordt namelijk binnen dit schema gegeven.

Reserveer voor:

- de Inleiding 10-15%
- de Instructie 60-70%
- de Afsluiting 20-25% van de beschikbare tijd.

Lesschema

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

De docent kan om stilte verzoeken, de deur dichtdoen, een aankondiging op het bord schrijven of op een andere manier de aandacht op zijn persoon en de komende les richten.

2. Uitleggen wat de leerdoelen van deze les zijn.

Voor de student en leraar is het van belang precies te weten wat de gewenste resultaten voor een les zijn.

Alleen maar noemen van onderwerpen die behandeld gaan worden is onvoldoende. De docent moet dit in begrijpelijke termen doen.

3. Noodzakelijke voorkennis in herinnering brengen.

De docent moet in het begin van de les zorgen dat de nieuwe informatie aansluit op en voortbouwt op eerder verworven begrippen en regels. De betreffende kennis van de studenten moet dan eerst opgehaald (geactiveerd) worden.

B Instructie

1. De informatie (leerstof) aanbieden.

De kern van het onderwijsleerproces is het aanbieden van de informatie door de docent of door het studieboek.

2. De studenten helpen bij het leren.

De docent blijft tijdens het leerproces niet op zijn plaats maar is bezig de verrichtingen van de studenten te observeren en waar nodig te ondersteunen. De toegepaste didactische opvatting kan worden gekenschetst als begeleid ontdekkend leren.

3. Studenten de gevraagde handeling laten tonen.

Het gaat hier niet om het controleren of de studenten hun huiswerk hebben gedaan, maar om een oefenmogelijkheid voor het uitvoeren van de gewenste (eind)handeling. Een gerichte vraag geeft de student de mogelijkheid om zichzelf en de docent te tonen dat hij of zij het weet of kan.

4. Terugkoppeling geven over de getoonde prestaties.

Het geven van informatie over de getoonde prestaties aan de studenten is informatief en versterkend. Terugkoppeling van informatie over de uitvoering van de handeling van de lerende gaat in vele gevallen vanzelf: het werkt of het werkt niet. Terugkoppeling moet in ieder geval informatie geven over de correctheid van het uitvoeren van de handelingen die tijdens het leren zichtbaar zijn.

- C Afsluiting**
- 1. Het eindresultaat vaststellen van het leren (toetsing).**
De bedoeling van deze gebeurtenis is vast te stellen of de student het leerdoel heeft bereikt en of hij of zij met zekerheid de gewenste eindhandelingen kan uitvoeren. Voor dit doel zijn verschillende toetsingsvormen mogelijk: bijvoorbeeld een opdracht uitvoeren of een opgave maken. Ze hoeven niet allemaal op hetzelfde tijdstip te worden afgenomen. Uiteraard behoort de toets afgestemd te zijn op de gestelde leerdoelen.
 - 2. Herhaling en wijzen op toepassingen.**
Deze laatste stap is gericht op het bevorderen van het bekijken op de wat langere termijn. Daarom dienen deze aanwijzingen lesoverstijgend te zijn en gericht op het zichtbaar maken van lesverbanden.

2. Voorkennis

Voor deze lessen wordt verondersteld dat de studenten beschikken over handigheid met de computer.

3. Algemeen

Technische zaken als het op de juiste wijze geïnstalleerd zijn van *Windows 10* of een Mac-computer zijn essentieel voor het geven van deze lessen.

In hoofdstuk 1 wordt uitgelegd hoe het programma op de computer kan worden gezet. U kunt er voor kiezen het programma aan het begin van les 1 op de computer te laten installeren door de studenten, maar u kunt er ook voor kiezen dit voor aanvang van les 1 te doen, ofwel door uzelf ofwel door de studenten.

Er is een internetverbinding nodig.

4. Overzicht van de lessen

De lesopzet bestaat uit 19 bijeenkomsten, waarbij gebruik wordt gemaakt van hoofdstuk 1 tot en met 20 uit **Crash course programmeren in Python**.

Les 1 Hoofdstuk 1 Aan de slag en Hoofdstuk 2 Variabelen en eenvoudige datatypes

Les 2 Hoofdstuk 3 Introductie van lijsten

Les 3 Hoofdstuk 4 Werken met lijsten

Les 4 Hoofdstuk 5 If-instructies

Les 5 Hoofdstuk 6 Woordenboeken

Les 6 Hoofdstuk 7 Invoer door de gebruiker en while-lussen

Les 7 Hoofdstuk 8 Functies

Les 8 Hoofdstuk 9 Classes

Les 9 Hoofdstuk 10 Bestanden en uitzonderingen

Les 10 Hoofdstuk 11 Je code testen

Les 11 Hoofdstuk 12 Een schip dat kogels afvuurt

Les 12 Hoofdstuk 13 Buitenaardse wezens!

Les 13 Hoofdstuk 14 Scores

Les 14 Hoofdstuk 15 Data genereren

Les 15 Hoofdstuk 16 Data downloaden

Les 16 Hoofdstuk 17 Werken met api's

Les 17 Hoofdstuk 18 Aan de slag met django

Les 18 Hoofdstuk 19 Gebruikersaccounts

Les 19 Hoofdstuk 20 Een toepassing vormgeven en implementeren

Indien gewenst kunt u naar eigen inzicht lessen samenvoegen of anders indelen.

Les 1

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

In deze les voer je jouw **eerste Python-programma** uit: `hello_world.py`. Eerst moet je nagaan of **Python** op je computer is geïnstalleerd. Als dat niet het geval is, moet je het **installeren**. Je gaat ook een **teksteditor installeren** om samen te werken met je Python-programma's. Teksteditoren herkennen Python-code en markeren gedeeltes terwijl je schrijft, waardoor het makkelijk wordt de **structuur van je code** te begrijpen.

In deze les leer je over de **verschillende soorten data** waarmee je in je Python-programma's kunt werken. Je leert ook hoe je je **data in variabelen opslaat** en hoe je die variabelen in je programma's gebruikt.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- instellen van je programmeeromgeving;
- Python op verschillende besturingssystemen;
- problemen bij installatie oplossen;
- Python-programma's vanuit een terminal uitvoeren;
- wat er daadwerkelijk gebeurt als je `hello_world.py` uitvoert;
- over variabelen;
- over getallen;
- opmerkingen maken;
- over The Zen of Python.

3. Noodzakelijke voorkennis in herinnering brengen.

Omdat dit de eerste bijeenkomst is, kunt u niet terugwijzen naar de vorige bijeenkomst.

B Instructie

1. De informatie (leerstof) aanbieden.

Pagina's 33-67.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.

Zie hiervoor het algemene schema.

C Afsluiting**1. Het eindresultaat vaststellen van het leren (toetsing).**

Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 47-48, 53, 59, 63, 65 en 67. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door.

Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 2

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

In deze en de volgende les leer je wat **lijsten** zijn en hoe je met de **elementen in een lijst** begint te werken. Met lijsten kun je **informatieverzamelingen op één plek opslaan**, of je nu maar een paar items of miljoenen items hebt. Lijsten zijn één van krachtigste functies van Python die direct toegankelijk zijn voor nieuwe programmeurs en die veel belangrijke programmeerconcepten verbinden.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- wat een lijst is;
- elementen wijzigen, toevoegen en verwijderen;
- een lijst organiseren;
- index-fouten voorkomen wanneer je met lijsten werkt.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering.

- instellen van je programmeeromgeving;
- Python op verschillende besturingssystemen;
- problemen bij installatie oplossen;
- Python-programma's vanuit een terminal uitvoeren;
- wat er daadwerkelijk gebeurt als je `hello_world.py` uitvoert;
- over variabelen;
- over getallen;
- opmerkingen maken;
- over The Zen of Python.

B Instructie

1. De informatie (leerstof) aanbieden.

Pagina's 69-84.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.

Zie hiervoor het algemene schema.

C Afsluiting**1. Het eindresultaat vaststellen van het leren (toetsing).**

Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 72, 78-79, 82 en 84. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door.

Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 3

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

In de vorige les heb je geleerd om een eenvoudige lijst te maken en met de afzonderlijke elementen in een lijst te werken. In deze les leer je hoe je **met een lus door een hele lijst gaat**, met slechts een paar regels code, onafhankelijk van de lengte van de lijst. Met een lus kun je **dezelfde actie of reeks van acties uitvoeren op elk item in een lijst**. Daardoor kun je efficiënt werken met lijsten van elke lengte, zelfs als die duizenden of zelfs miljoenen items bevatten.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- door een hele lijst lussen;
- fouten met inspringen voorkomen;
- numerieke lijsten maken;
- werken met een deel van een lijst;
- over tupels;
- je code opmaken.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- wat een lijst is;
- elementen wijzigen, toevoegen en verwijderen;
- een lijst organiseren;
- index-fouten voorkomen wanneer je met lijsten werkt.

B Instructie

1. De informatie (leerstof) aanbieden.

Pagina's 85-107.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Kinderen het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.

Zie hiervoor het algemene schema.

C Afsluiting **1. Het eindresultaat vaststellen van het leren (toetsing).**
Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 92-93, 96-97, 101-102, 104 en 107. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door. Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 4

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

Bij programmeren gaat het vaak om het **bekijken van een reeks voorwaarden en bepalen welke actie er op basis daarvan ondernomen moet worden**. Met de **if-instructie** van Python kun je de actuele status van een programma bekijken en daar adequaat op reageren.

In deze les leer je **voorwaardelijke tests** te schrijven, waarmee je elke gewenste voorwaarde kunt controleren. Je leert eenvoudige if-instructies te schrijven en hoe je een complexere reeks if-instructies maakt om te bepalen wanneer de exacte gewenste voorwaarden aanwezig zijn. Daarna pas je dit concept toe op **lijsten**, zodat je een **for-lus** kunt schrijven die de meeste items in een lijst op de ene manier behandelt, maar bepaalde items met specifieke waarden op een andere manier.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- over een eenvoudig voorbeeld van if-instructies;
- over voorwaardelijke tests;
- over if-Instructies;
- if-instructies gebruiken bij lijsten;
- je if-instructies opmaken.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- door een hele lijst lussen;
- fouten met inspringen voorkomen;
- numerieke lijsten maken;
- werken met een deel van een lijst;
- over tupels;
- je code opmaken.

B Instructie

1. De informatie (leerstof) aanbieden.

Pagina's 109-129.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.

Zie hiervoor het algemene schema.

C Afsluiting**1. Het eindresultaat vaststellen van het leren (toetsing).**

Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 116, 123-124, 127-128 en 129. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door.

Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 5

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

In deze les leer je de **woordenboeken** van Python te gebruiken, waarmee je stukjes gerelateerde informatie met elkaar kunt verbinden. Je leert hoe je de **informatie benadert** als die eenmaal in een woordenboek zit en hoe je die informatie kunt **aanpassen**. Omdat woordenboeken een haast eindeloze hoeveelheid informatie kunnen bevatten, laten we je zien hoe je door de data in een woordenboek lust. Daarnaast leer je om woordenboeken in lijsten, lijsten in woordenboeken en zelfs woordenboeken in andere woordenboeken te **nesten**.

Als je het principe van woordenboeken begrijpt, kun je diverse **objecten uit de praktijk nauwkeuriger modelleren**. Je kunt een woordenboek maken dat een persoon voorstelt en er dan zoveel informatie als je maar wilt over die persoon in opslaan. Je kunt hun naam, leeftijd, locatie, beroep en elk ander beschrijvend kenmerk van een persoon opslaan. Je kunt twee soorten informatie opslaan die bij elkaar horen, zoals een lijst met woorden en hun betekenissen, een lijst met namen van personen en hun geluksgetallen, een lijst van bergen en hun hoogte, enzovoort.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- een eenvoudig woordenboek;
- werken met woordenboeken;
- door een woordenboek lussen;
- over nesten.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- over een eenvoudig voorbeeld van if-instructies;
- over voorwaardelijke tests;
- over if-Instructies;
- if-instructies gebruiken bij lijsten;
- je if-instructies opmaken.

-
- B Instructie**
- 1. De informatie (leerstof) aanbieden.**
Pagina's 131-153.
 - 2. De studenten helpen bij het leren.**
Zie hiervoor het algemene schema.
 - 3. Studenten het gevraagde gedrag laten tonen.**
Zie hiervoor het algemene schema.
 - 4. Terugkoppeling geven over de getoonde prestaties.**
Zie hiervoor het algemene schema.
- C Afsluiting**
- 1. Het eindresultaat vaststellen van het leren (toetsing).**
Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 139, 145-146 en 152-153. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython
 - 2. Herhaling en wijzen op toepassingen.**
Keer terug naar de doelstellingen en lees ze nog eenmaal door.
Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 6

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

De meeste programma's worden geschreven om een probleem van een eindgebruiker op te lossen. Om dat te doen heb je meestal informatie nodig van de gebruiker. Een simpel voorbeeld is iemand die wil weten of hij oud genoeg is om te stemmen. Als je een programma schrijft om deze vraag te beantwoorden, moet je de leeftijd van de gebruiker weten voordat je een oplossing kunt bieden. Het programma moet de gebruiker vragen om **input**, oftewel om zijn leeftijd in te voeren.

Zodra deze informatie ingevoerd is, kan het programma die vergelijken met de kiesgerechtigde leeftijd om te bepalen of de gebruiker oud genoeg is, en vervolgens het **resultaat doorgeven**. In deze les leer je hoe je **invoer van de gebruiker kunt accepteren**, zodat je programma daar dan mee kan werken. Wanneer je programma een naam nodig heeft, kun je de gebruiker om zijn naam vragen. Wanneer je programma een lijst met namen nodig heeft, kun je de gebruiker om een reeks namen vragen. Hiervoor gebruik je de **input()-functie**.

Je leert ook hoe je je programma's zo lang laat draaien als de gebruikers dat willen, zodat ze zoveel informatie kunnen invoeren als nodig is. Daarna kan jouw programma met die informatie aan de slag. Je maakt daarbij gebruik van de **while-lus** in Python om programma's te laten draaien zolang aan bepaalde voorwaarden voldaan wordt. Met de mogelijkheid om met invoer van de gebruiker te werken en de mogelijkheid om te bepalen hoe lang je programma's moeten blijven draaien, kun je volledig interactieve programma's schrijven.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- hoe de input()-functie werkt;
- een introductie van while-lussen;
- een while-lus gebruiken bij lijsten en woordenboeken.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- een eenvoudig woordenboek;
- werken met woordenboeken;
- door een woordenboek lussen;
- over nesten.

B Instructie**1. De informatie (leerstof) aanbieden.**

Pagina's 155-170.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.

Zie hiervoor het algemene schema.

C Afsluiting**1. Het eindresultaat vaststellen van het leren (toetsing).**

Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 160, 166 en 170. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door.

Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 7

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

In deze les leer je hoe je **functies** schrijft. Dat zijn blokken code met een naam, die zijn ontworpen om **één specifieke opdracht uit te voeren**. Wanneer je een bepaalde taak wilt uitvoeren die je hebt gedefinieerd in een functie, ga je de **functie** die daarvoor verantwoordelijk is **aanroepen**. Als je die taak meerdere keren in je programma moet uitvoeren hoef je niet steeds opnieuw alle code voor diezelfde taak te typen. Je roept gewoon de functie aan die voor het uitvoeren van die taak is opgesteld, en de aanroep geeft Python de opdracht om de code binnen de functie uit te voeren. Je zult merken dat het gebruik van functies ervoor zorgt dat je programma's makkelijker te schrijven, lezen, testen en repareren zijn.

In deze les leer je ook manieren om **informatie aan functies door te geven**. Je gaat aan de slag met het schrijven van functies waarvan het de belangrijkste taak is om informatie weer te geven, en met andere functies die zijn ontworpen om gegevens te verwerken en een waarde of serie waarden te retourneren. Tot slot leer je om **functies op te slaan in aparte bestanden, modules genaamd**, om je te helpen je belangrijkste **programmabestanden te ordenen**.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- een functie definiëren;
- argumenten doorgeven;
- retourwaarden;
- een lijst doorgeven;
- een willekeurig aantal argumenten doorgeven;
- je functies in modules opslaan;
- functies opmaken.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- hoe de input()-functie werkt;
- een introductie van while-lussen;
- een while-lus gebruiken bij lijsten en woordenboeken.

-
- B Instructie**
- 1. De informatie (leerstof) aanbieden.**
Pagina's 171-199.
 - 2. De studenten helpen bij het leren.**
Zie hiervoor het algemene schema.
 - 3. Studenten het gevraagde gedrag laten tonen.**
Zie hiervoor het algemene schema.
 - 4. Terugkoppeling geven over de getoonde prestaties.**
Zie hiervoor het algemene schema.
- C Afsluiting**
- 1. Het eindresultaat vaststellen van het leren (toetsing).**
Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 173-174, 179-180, 185, 190, 193 en 198. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython
 - 2. Herhaling en wijzen op toepassingen.**
Keer terug naar de doelstellingen en lees ze nog eenmaal door.
Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 8

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

Objectgeoriënteerd programmeren is een van de effectiefste manieren om software te schrijven. Bij objectgeoriënteerd programmeren schrijf je **classes die dingen en situaties uit het echte leven voorstellen en maak je objecten die op deze classes zijn gebaseerd**. Wanneer je een class schrijft, definieer je het algemene gedrag dat een hele categorie van objecten vertoont. Wanneer je **vanuit de class afzonderlijke objecten maakt**, wordt elk object automatisch voorzien van het algemene gedrag. Daarna kun je elk object alle door jou gewenste **unieke eigenschappen geven**. Je zult versteld staan hoe goed situaties uit het echte leven gemodelleerd kunnen worden door middel van objectgeoriënteerd programmeren.

Het maken van een object vanuit een class wordt bij programmeren **instantiation** genoemd. Je werkt dan met **instances van een class**. In het Nederlands wordt de term **exemplaren** gebruikt. In deze les **schrijf je classes en maak je exemplaren van die classes**. Je **specificeert wat voor informatie in de exemplaren** kan worden **opgeslagen** en je **definieert acties** die met deze exemplaren ondernomen kunnen worden. Je gaat ook classes schrijven die de functionaliteit van bestaande classes uitbreiden, zodat vergelijkbare classes code efficiënt kunnen delen. Je **slaat de classes op in modules en importeert classes** die door andere programmeurs geschreven zijn in je eigen programmabestanden.

Met kennis van objectgeoriënteerd programmeren kun je de wereld zien zoals een programmeur dat doet. Het helpt je om je code echt te kennen, niet alleen wat er regel voor regel gebeurt, maar ook het grotere idee erachter. De logica achter classes traint je om logisch na te denken, zodat je programma's kunt schrijven die bijna alle problemen die je ondervindt effectief aanpakken.

Classes maken het leven makkelijker voor jou en de andere programmeurs met wie je samen moet werken wanneer je steeds complexere uitdagingen aanpakt. Wanneer jij en de andere programmeurs code schrijven die op dezelfde logica gebaseerd is, begrijp je elkaars werk. Je programma's zijn logisch voor andere collega's, zodat iedereen meer kan bereiken.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- een class maken en gebruiken;
- werken met classes en exemplaren;
- overerven;
- classes importeren;
- de standaardbibliotheek van Python;
- classes opmaken.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- een functie definiëren;
- argumenten doorgeven;
- retourwaarden;
- een lijst doorgeven;
- een willekeurig aantal argumenten doorgeven;
- je functies in modules opslaan;
- functies opmaken.

B Instructie 1. De informatie (leerstof) aanbieden.

Pagina's 201-229.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.

Zie hiervoor het algemene schema.

C Afsluiting 1. Het eindresultaat vaststellen van het leren (toetsing).

Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 207, 212, 219-220, 226 en 227-228. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door.

Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 9

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

Nu je de basisvaardigheden voor het schrijven van gebruiksvriendelijke en goed georganiseerde programma's beheerst, is het tijd om na te denken over hoe je jouw programma's nog relevanter en bruikbaarder kunt maken. In deze les leer je met **bestanden** werken, zodat jouw **programma's snel grote hoeveelheden data kunnen analyseren**.

Je leert hoe je met **fouten om kunt gaan**, zodat je programma's niet crashen wanneer ze onverwachte situaties tegenkomen. Je komt meer te weten over **uitzonderingen**: speciale objecten die Python maakt om fouten af te handelen die zich voordoen als een programma wordt uitgevoerd. Je maakt ook kennis met de **json-module**, waarmee je **gebruikersgegevens kunt opslaan**, zodat ze niet verloren gaan wanneer je programma stopt.

Als je leert werken met bestanden en weet hoe je gegevens kunt opslaan, worden je programma's makkelijker in gebruik. Gebruikers kunnen kiezen welke gegevens ze willen invoeren en wanneer ze die invoeren. Ze kunnen je programma uitvoeren, er wat in werken, het programma sluiten en later weer verdergaan waar ze gebleven waren. Doordat je leert om **uitzonderingen op te lossen**, kun je omgaan met situaties waardoor je programma's zouden kunnen crashen, zoals niet-bestaande bestanden en andere kwesties. Zo worden je programma's sterker voor het geval ze worden geconfronteerd met slechte data, of dat nu komt door onschuldige vergissingen of door kwaadaardige pogingen om in je programma's in te breken. Met de vaardigheden die je in deze les leert, maak je jouw programma's beter toepasbaar, bruikbaar en stabiel.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- een bestand lezen;
- naar een bestand schrijven;
- uitzonderingen;
- data opslaan.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- een class maken en gebruiken;
- werken met classes en exemplaren;
- overerven;
- classes importeren;
- de standaardbibliotheek van Python;
- classes opmaken.

B Instructie 1. De informatie (leerstof) aanbieden.

Pagina's 231-258.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.

Zie hiervoor het algemene schema.

C Afsluiting 1. Het eindresultaat vaststellen van het leren (toetsing).

Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 239, 242, 251, 257-258. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door.

Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 10

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

Wanneer je een functie of class schrijft, kun je voor die code ook **tests schrijven**. Een test bewijst dat je code naar behoren werkt in reactie op alle soorten invoer waarvoor het programma ontworpen is. Wanneer je tests schrijft, kun je er zeker van zijn dat je code goed werkt als meer mensen je programma's gaan gebruiken. Je kunt nieuwe code ook testen terwijl je die toevoegt, om ervoor te zorgen dat je wijzigingen het bestaande gedrag van je programma niet verstoren. Elke programmeur maakt fouten, dus moet elke programmeur zijn code vaak testen om zo problemen te ondervangen voordat gebruikers die tegenkomen.

In deze les leer je hoe je jouw code test met hulpmiddelen uit de **unittest-module van Python**. Je leert een testcase bouwen en te controleren of een bepaalde invoer resulteert in de gewenste uitvoer. Je ziet ook hoe een geslaagde test en een falende test eruitziet, en je leert hoe een falende test je kan helpen jouw code te verbeteren. Je leert hoe je functies en classes test en krijgt inzicht in het aantal tests dat je voor een project moet schrijven.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- een functie testen;
- het testen van een class.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- een bestand lezen;
- naar een bestand schrijven;
- uitzonderingen;
- data opslaan.

B Instructie

1. De informatie (leerstof) aanbieden.

Pagina's 259-273.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.

Zie hiervoor het algemene schema.

C Afsluiting**1. Het eindresultaat vaststellen van het leren (toetsing).**

Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 266 en 272. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door.

Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 11

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

We gaan een spel bouwen! Hiervoor ga je **Pygame** gebruiken, een verzameling leuke, krachtige **Python-modules** voor het **beheren van graphics, animaties en zelfs geluid**, waarmee je eenvoudiger geavanceerde spellen kunt bouwen. Omdat Pygame taken als het tekenen van afbeeldingen op het scherm afhandelt, kun jij veel van het vervelende, moeilijke programmeren van code overslaan en je richten op de bovenliggende logica van de speldynamiek.

In deze les stel je Pygame in en vervolgens maak je een **schip dat naar links en rechts beweegt en kogels afvuurt als reactie op de invoer van de speler**. In de volgende twee lessen maak je een vloot buitenaardse wezens om te vernietigen en daarna verfijn je het spel verder, zoals door het vaststellen van een limiet voor het aantal schepen dat je kunt gebruiken en het toevoegen van een scorebord. In deze les kom je ook meer te weten over het **beheren van grote projecten** die uit meerdere bestanden bestaan. **Je schoont een heleboel code op en beheert de bestandsinhoud** om het project georganiseerd en de code efficiënt te houden.

Het maken van spellen is de perfecte manier om plezier te hebben tijdens het leren van een taal. Het geeft veel voldoening om anderen spellen te zien spelen die jij geschreven hebt, en het schrijven van een eenvoudig spel helpt je te begrijpen hoe professionele spellen worden geschreven. Voer tijdens het doorwerken van deze les de code in en voer die ook uit om te begrijpen hoe elk blok code bijdraagt aan de algehele gameplay. Experimenteer met verschillende waarden en instellingen om een beter inzicht te krijgen in hoe je interacties in je eigen spellen kunt verfijnen.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- je project plannen;
- Pygame installeren;
- beginnen met het spelproject;
- de afbeelding van het schip toevoegen;
- opschonen: de `game_functions`-module;
- het schip besturen;
- hoe je kogels laat afvuren.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- een functie testen;
- het testen van een class.

B Instructie 1. De informatie (leerstof) aanbieden.

Pagina's 279-310.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.

Zie hiervoor het algemene schema.

C Afsluiting 1. Het eindresultaat vaststellen van het leren (toetsing).

Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 294, 302 en 310. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door.

Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 12

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

In deze les **voeg je buitenaardse wezens toe** aan Alien Invasion. Eerst wordt één buitenaards wezen toegevoegd bovenin het scherm, en daarna maak je een hele vloot buitenaardse wezens. Je laat die vloot zijwaarts en naar beneden oprukken en je verwijdert elk buitenaards wezen dat door een kogel wordt geraakt. Tot slot beperk je het aantal schepen dat een speler tot zijn beschikking heeft, en beëindig je het spel wanneer de speler geen schepen meer over heeft.

In deze les leer je meer over Pygame en over het beheren van een groter project. Je leert ook hoe je **botsingen tussen spelelementen detecteert**, bijvoorbeeld tussen kogels en buitenaardse wezens. Door het detecteren van botsingen kun je interactie tussen elementen in je spel definiëren: je kunt een personage binnen de muren van een doolhof houden of een bal doorgeven van het ene personage naar het andere. Bovendien blijven we werken aan de hand van een plan dat we er af en toe bijpakken om doelgericht code te blijven schrijven.

Voordat je begint met het schrijven van nieuwe code om de vloot buitenaardse wezens aan het scherm toe te voegen, bekijken we eerst het project en werk je jouw plan bij.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- je project doornemen;
- het eerste buitenaardse wezen maken;
- een hele vloot buitenaardse wezens bouwen;
- de vloot laten bewegen;
- buitenaardse wezens neerschieten;
- het spel beëindigen;
- identificeren welke delen van het spel uitgevoerd moeten worden.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- je project plannen;
- Pygame installeren;

- beginnen met het spelproject;
- de afbeelding van het schip toevoegen;
- opschonen: de `game_functions`-module;
- het schip besturen;
- hoe je kogels laat afvuren.

B Instructie **1. De informatie (leerstof) aanbieden.**
Pagina's 311-337.

2. De studenten helpen bij het leren.
Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.
Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.
Zie hiervoor het algemene schema.

C Afsluiting **1. Het eindresultaat vaststellen van het leren (toetsing).**
Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 322, 326, 331 en 337. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.
Keer terug naar de doelstellingen en lees ze nog eenmaal door. Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 13

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

In deze les rond je het Alien Invasion-spel af. Je voegt een **Play-knop** toe om een spel op afroep te starten of het spel te herstarten als het afgelopen is. Ook pas je de **snelheid van het spel** aan, zodat het steeds sneller gaat wanneer de speler een niveau hoger bereikt. Bovendien voeg je een **scoresysteem** toe.

Aan het eind van deze les weet je genoeg om te beginnen met het schrijven van spellen die steeds moeilijker worden naarmate een speler vordert, en die ook scores laten zien.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- de Play-knop toevoegen;
- een niveau hoger toevoegen;
- scores implementeren.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- je project doornemen;
- het eerste buitenaardse wezen maken;
- een hele vloot buitenaardse wezens bouwen;
- de vloot laten bewegen;
- buitenaardse wezens neerschieten;
- het spel beëindigen;
- identificeren welke delen van het spel uitgevoerd moeten worden.

B Instructie

1. De informatie (leerstof) aanbieden.

Pagina's 339-366.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.
Zie hiervoor het algemene schema.

C Afsluiting

1. Het eindresultaat vaststellen van het leren (toetsing).

Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 346, 349 en 365. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door.

Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 14

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

Bij **datavisualisatie** gaat het om het **analyseren van data door die visueel weer te geven**. Dit is nauw verbonden met **datamining**, waarbij **code gebruikt wordt om patronen en verbanden in een dataset te ontdekken**. Een dataset kan een kleine lijst met getallen zijn die op één regel code past, maar ook vele gigabytes aan data. Een mooie weergave van data draait om meer dan mooie plaatjes alleen. Wanneer je een dataset eenvoudig en visueel aantrekkelijk weergeeft, wordt de betekenis ervan duidelijk voor de kijkers.

Gebruikers zullen dan patronen in en het belang van je datasets ontdekken, waarvan ze het bestaan niet kenden.

Gelukkig heb je geen supercomputer nodig om complexe data te visualiseren. Met de efficiency van Python kun je, gewoon op een laptop, snel datasets van miljoenen afzonderlijke **gegevenspunten analyseren**.

De gegevenspunten hoeven ook **niet alleen maar getallen te zijn**. Met de basisprincipes die je in het eerste deel van dit boek geleerd hebt, kun je ook **niet-numerieke data analyseren**.

Python wordt gebruikt voor gegevensintensief werk in genetica, klimaatonderzoek, politieke en economische analyses en nog veel meer. Datawetenschappers hebben een indrukwekkende reeks hulpmiddelen voor visualisatie en analyse in Python geschreven, waarvan er veel ook voor jou beschikbaar zijn. Een van de populairste hulpmiddelen is **matplotlib**, een **bibliotheek voor wiskundige plotting**. Met plotting wordt in kaart brengen bedoeld.

Je gaat matplotlib gebruiken voor eenvoudige plots, zoals **lijngrafieken** en **spreidingsdiagrammen**. Daarna maak je **interessantere datasets**, gebaseerd op het concept van een willekeurige wandeling: een **visualisatie die gegenereerd wordt op basis van een reeks willekeurige beslissingen**.

We gebruiken ook een pakket met de naam **Pygal**, dat gericht is op het maken van visualisaties die goed werken op digitale apparaten. Je kunt Pygal gebruiken voor het benadrukken van elementen en voor het wijzigen van de afmetingen daarvan terwijl de gebruiker bezig is met je visualisatie en je kunt de hele weergave makkelijk van afmeting veranderen, zodat die past op een kleine smartwatch en op een gigantische monitor. Je gaat Pygal gebruiken om te ontdekken wat er gebeurt wanneer je dobbelstenen op verschillende manieren laat rollen.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- matplotlib installeren;
- een eenvoudige lijngrafiek plotten;
- data genereren voor willekeurige 'wandelingen';
- een dataset te genereren die het gooien van dobbelstenen nabootst.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- de Play-knop toevoegen;
- een niveau hoger toevoegen;
- scores implementeren.

B Instructie

1. De informatie (leerstof) aanbieden.

Pagina's 369-397.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.

Zie hiervoor het algemene schema.

C Afsluiting

1. Het eindresultaat vaststellen van het leren (toetsing).

Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 380, 388 en 396. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door.

Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 15

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

In deze les ga je **datasets downloaden van onlinebronnen** en daar werkende **visualisaties van maken**. Er kan online een onvoorstelbare verscheidenheid aan data gevonden worden, waarvan een groot gedeelte nog niet grondig onderzocht is. Als je weet hoe je deze data analyseert, kun je patronen en verbanden ontdekken die nog niemand anders gevonden heeft.

We zullen data benaderen en visualiseren die is opgeslagen in twee veelvoorkomende indelingen: **CSV** en **JSON**. Je gebruikt de **csv-module van Python** om weerdata te verwerken die in de CSV-indeling (kommagescheiden waarden) zijn opgeslagen, en je analyseert de minimumen maximumtemperaturen gedurende een bepaalde periode op twee verschillende locaties. Vervolgens gebruik je **matplotlib om een grafiek te genereren** op basis van de gedownloade data om temperatuurvariatie in twee heel verschillende omgevingen weer te geven: Sitka in Alaska en Death Valley in Californië. Verderop in deze les gebruik je de **json-module** om bevolkingsgegevens te benaderen die zijn opgeslagen in de JSON-indeling en gebruik je Pygal om per land een kaart van de bevolking te tekenen.

Aan het eind van deze les ben je er klaar voor om met verschillende soorten en indelingen van datasets te werken en heb je meer inzicht in het bouwen van complexe visualisaties. De vaardigheid om onlinedata van verschillende typen en indelingen te benaderen en te visualiseren, is essentieel voor het werken met een breed scala aan datasets uit de praktijk.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- over de CSV-bestandsindeling;
- globale datasets in kaart brengen: JSON-indeling.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- matplotlib installeren;
- een eenvoudige lijngrafiek plotten;

- data genereren voor willekeurige ‘wandelingen’;
- een dataset te genereren die het gooien van dobbelstenen nabootst.

B Instructie **1. De informatie (leerstof) aanbieden.**

Pagina's 399-426.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.

Zie hiervoor het algemene schema.

C Afsluiting **1. Het eindresultaat vaststellen van het leren (toetsing).**

Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 412 en 425. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door.

Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 16

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

In deze les leer je hoe je een zelfstandig programma schrijft om een **visualisatie te genereren op basis van gegevens die het ophaalt**. Je programma gaat een **web application programming interface (API) gebruiken om automatisch specifieke informatie van een website op te vragen** in plaats van complete pagina's. Het zal deze informatie dan gebruiken om een **visualisatie te genereren**. Omdat programma's die op deze manier zijn geschreven altijd recente data gebruiken om een visualisatie te genereren, zal deze altijd up-to-date zijn, zelfs wanneer die data snel verandert.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- een web-API gebruiken;
- archieven visualiseren met Pygal;
- over de Hacker News API.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- over de CSV-bestandsindeling;
- globale datasets in kaart brengen: JSON-indeling.

B Instructie

1. De informatie (leerstof) aanbieden.

Pagina's 427-444.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.

Zie hiervoor het algemene schema.

C Afsluiting **1. Het eindresultaat vaststellen van het leren (toetsing).**
Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 444. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door.
Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 17

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

Achter de schermen zijn de websites van vandaag eigenlijk rijke toepassingen die fungeren als volledig ontwikkelde desktop-toepassingen. Python bevat een geweldige set hulpmiddelen voor het bouwen van **webtoepassingen**.

In deze les leer je om **Django** (<http://djangoproject.com/>) te gebruiken om een project met de naam Learning Log te bouwen: een onlinedagboekstelsel waarmee je de informatie die je over bepaalde onderwerpen geleerd hebt bij kunt houden.

We gaan een **specificatie voor dit project schrijven** en dan **modellen definiëren voor de data** waarmee de toepassing zal werken. We gebruiken het **admin-systeem** van Django om wat eerste data in te voeren en leren daarna om weergaven en sjablonen te schrijven zodat Django de pagina's van onze site kan bouwen.

Django is een **web framework**: een set hulpmiddelen die is ontworpen om je te helpen **interactieve websites te bouwen**.

Django kan reageren op paginaverzoeken, het makkelijker maken om een database te lezen en er naartoe te schrijven, gebruikers beheren en nog veel meer. In de volgende lessen verfijnen we het Learning Log-project en implementeren het vervolgens op een live-server zodat jij en je vrienden het kunnen gebruiken.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- een project opzetten;
- een toepassing starten;
- pagina's maken: de homepage van Learning Log;
- extra pagina's bouwen.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- een web-API gebruiken;
- archieven visualiseren met Pygal;
- over de Hacker News API.

-
- B Instructie**
- 1. De informatie (leerstof) aanbieden.**
Pagina's 447-477.
 - 2. De studenten helpen bij het leren.**
Zie hiervoor het algemene schema.
 - 3. Studenten het gevraagde gedrag laten tonen.**
Zie hiervoor het algemene schema.
 - 4. Terugkoppeling geven over de getoonde prestaties.**
Zie hiervoor het algemene schema.
- C Afsluiting**
- 1. Het eindresultaat vaststellen van het leren (toetsing).**
Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 453, 463 of 476. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython
 - 2. Herhaling en wijzen op toepassingen.**
Keer terug naar de doelstellingen en lees ze nog eenmaal door.
Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 18

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

De spil van een webtoepassing is dat elke gebruiker, overal ter wereld, een **account** in je toepassing kan **maken** en het kan **gebruiken**. In deze les bouw je **formulieren** zodat gebruikers hun eigen onderwerpen en vermeldingen kunnen toevoegen en bestaande vermeldingen kunnen bewerken. Je leert ook hoe Django zich wapent tegen veelvoorkomende aanvallen op pagina's met formulieren, zodat jij niet te veel tijd hoeft te besteden aan het denken over het beveiligen van je toepassingen.

We maken ook een systeem om gebruikers te verifiëren. Je maakt een **registratiepagina** voor gebruikers om accounts te maken en **bepert** dan **de toegang tot bepaalde pagina's tot ingelogde gebruikers**. Daarna passen we enkele **weergavefuncties** aan zodat gebruikers alleen hun eigen data kunnen zien. Je leert de data van je gebruikers **veilig en beschermd te houden**.

**In deze les leer je het volgende:
(zet ook dit ergens goed leesbaar neer)**

- gebruikers toestaan om data in te voeren;
- gebruikersaccounts instellen;
- gebruikers toestaan om hun eigen data te beheren.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- een project opzetten;
- een toepassing starten;
- pagina's maken: de homepage van Learning Log;
- extra pagina's bouwen.

B Instructie

1. De informatie (leerstof) aanbieden.

Pagina's 479-509.

2. De studenten helpen bij het leren.

Zie hiervoor het algemene schema.

3. Studenten het gevraagde gedrag laten tonen.

Zie hiervoor het algemene schema.

4. Terugkoppeling geven over de getoonde prestaties.

Zie hiervoor het algemene schema.

C Afsluiting**1. Het eindresultaat vaststellen van het leren (toetsing).**

Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 491, 500 en 508. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython

2. Herhaling en wijzen op toepassingen.

Keer terug naar de doelstellingen en lees ze nog eenmaal door.

Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

Les 19

A Inleiding

1. Zorgen dat de studenten aandacht hebben voor de les.

Zie hiervoor het algemene schema.

2. Uitleggen wat de leerdoelen van deze les zijn.

Geef eerst onderstaande informatie; schrijf de vet gedrukte woorden ergens in grote letters zodat iedereen ze goed kan zien (bijvoorbeeld op een bord).

Learning Log is nu volledig functioneel, maar het heeft nog geen vormgeving en draait alleen op je lokale machine. In deze les zullen we het project op een eenvoudige maar professionele manier **vormgeven** en het dan op een **live-server implementeren**, zodat iedereen in de wereld een account kan maken.

Voor het vormgeven gebruiken we de **Bootstrap-bibliotheek**, een verzameling hulpmiddelen voor het vormgeven van webtoepassingen, zodat ze er professioneel uitzien op alle moderne apparaten, van een groot, plat beeldscherm tot een smartphone. Hiervoor gebruiken we de **djangobootstrap3-toepassing**, wat je ook meteen laat oefenen met toepassingen die door andere Django-ontwikkelaars gemaakt zijn.

We gaan Learning Log implementeren met **Heroku**, een site waarmee je jouw project op een van de **servers** kunt zetten en het beschikbaar maakt voor iedereen met een internetverbinding. We gaan ook een **versiebeheersysteem** met de naam **Git** gebruiken om **wijzigingen aan het project bij te houden**.

Wanneer je klaar bent met Learning Log ben je in staat om eenvoudige webtoepassingen te ontwikkelen, ervoor te zorgen dat ze er goed uitzien en kun je ze op een live-server te implementeren. Daarnaast ben je in staat om meer geavanceerde leermiddelen te gebruiken naarmate je jouw vaardigheden ontwikkelt.

In deze les leer je het volgende:

(zet ook dit ergens goed leesbaar neer)

- Learning Log vormgeven;
- Learning Log implementeren.

3. Noodzakelijke voorkennis in herinnering brengen.

Breng in het kort de leerstof van de vorige bijeenkomst in herinnering:

- gebruikers toestaan om data in te voeren;
- gebruikersaccounts instellen;
- gebruikers toestaan om hun eigen data te beheren.

- B Instructie**
- 1. De informatie (leerstof) aanbieden.**
Pagina's 511-540.
 - 2. De studenten helpen bij het leren.**
Zie hiervoor het algemene schema.
 - 3. Studenten het gevraagde gedrag laten tonen.**
Zie hiervoor het algemene schema.
 - 4. Terugkoppeling geven over de getoonde prestaties.**
Zie hiervoor het algemene schema.
- C Afsluiting**
- 1. Het eindresultaat vaststellen van het leren (toetsing).**
Bespreek in het kort eventuele veel voorkomende problemen die zich tijdens B Instructie voordeden.

Laat de studenten de Probeer het zelf-oefeningen doen op pagina 522 en 540. De oplossingen voor de oefeningen zijn te vinden op de webpagina www.visualsteps.nl/crashcoursepython
 - 2. Herhaling en wijzen op toepassingen.**
Keer terug naar de doelstellingen en lees ze nog eenmaal door.
Moedig de studenten aan de opgedane kennis en vaardigheden thuis te herhalen.

5. Ander cursusmateriaal

De populaire boeken van Visual Steps zijn uitstekend geschikt voor gebruik als cursusmateriaal. Een overzicht van alle boeken vindt u op **www.visualsteps.nl**

Bij een groot aantal boeken zijn docentenhandleidingen beschikbaar. Kijk voor meer informatie op **www.visualsteps.nl/docent**
Na registratie krijgt u gratis toegang tot alle aanvullende materialen en handleidingen.