

Opleiding : Campus Atlas - Python, objectgeoriënteerd programmeren

Praktijkcursus - 5d - 35u00 - Ref. PYE

Prijs : 2610 € V.B.

NEW

Na afloop van de cursus kunnen de deelnemers de belangrijkste functies van de Python-taal gebruiken om multiplatformtoepassingen te ontwikkelen. Dit trainingsprogramma is gericht op medewerkers in de professionele branches die onder de OPCO Atlas vallen.

Pedagogische doelstellingen

Aan het einde van de training is de deelnemer in staat om:

- ✓ De basisbeginselen van de Python-taal en het bijbehorende ecosysteem begrijpen
- ✓ De principes van objectprogrammering aanleren
- ✓ Functies en modules begrijpen en gebruiken
- ✓ Grafische interfaces ontwerpen
- ✓ Tools gebruiken om de kwaliteit van een Python-programma te testen en te evalueren

Doelgroep

Voor OPCO Atlas leden: ontwikkelaars, ingenieurs, projectmanagers met nauwe banden met ontwikkeling.

Voorafgaande vereisten

Basis programmeervaardigheden (bij voorkeur in objectgeoriënteerde talen).

Praktische modaliteiten

Praktisch werk

Praktische oefeningen en/of casestudies.

Leer methodes

60% pratique – 40% théorie. Pour optimiser le parcours d'apprentissage, des modules e-learning peuvent être fournis avant et après la session présentielle ou la classe virtuelle, sur simple demande du participant.

DEELNEMERS

Voor OPCO Atlas leden:
ontwikkelaars, ingenieurs,
projectmanagers met nauwe banden
met ontwikkeling.

VOORAFGAANDE VEREISTEN

Basis programmeervaardigheden (bij voorkeur in objectgeoriënteerde talen).

VAARDIGHEDEN VAN DE CURSUSLEIDER

De deskundigen die de cursus leiden zijn specialisten op het betreffende vakgebied. Zij werden geselecteerd door onze pedagogische teams zowel om hun vakkennis als hun pedagogische vaardigheden voor elke cursus die zij geven. Zij hebben minstens vijf tot tien jaar ervaring in hun vakgebied en oefenen of oefenden verantwoordelijke bedrijfsfuncties uit.

BEOORDELINGSMODALITEITEN

De cursusleider beoordeelt de pedagogische vooruitgang van de deelnemer gedurende de gehele cursus aan de hand van meerkeuzevragen, praktijksituaties, praktische opdrachten, ...
De deelnemer legt ook van tevoren en naderhand een test af ter bevestiging van de verworven kennis.

Opleidingsprogramma

1 Algoritmen, redeneren vóór ontwerpen - Vooropleiding digitale leerinhoud

- Inleiding tot algoritmen.
- Basisinstructies in pseudocode.

Digitale activiteiten

In deze e-learningcursus leer je nadenken voordat je een programma ontwerpt door de basisprincipes van algoritmiëk te behandelen. De fundamentele instructies in pseudocode komen ook aan bod.

2 Inleiding tot Python

- Installatie en configuratie van de omgeving.
- Fundamentele gegevenstypes.
- Basisgegevensstructuren.
- Variabelen en toepassingsgebied.
- Operatoren en uitdrukkingen.
- Python stijl en conventies (PEP 8).

Praktisch werk

Setup de l'environnement. Types et structures. Style et bonnes pratiques.

3 Besturingsstructuren en -functies

- Geavanceerde besturingsstructuren.
- Functies definiëren en gebruiken.
- Meerdere parameters en retours.
- Fout- en uitzonderingsafhandeling.
- Python modules en pakketten.
- Goede structureringspraktijken.

Praktisch werk

Structures de contrôle. Fonctions et modularité. Gestion des erreurs.

4 Objectgeoriënteerde benadering

- Fundamentele principes van OOP.
- Klassen en instanties.
- Erfgoed en samenstelling.
- Polymorfisme en abstractie.
- Interfaces en abstracte klassen.
- Ontwerppatronen in Python.

Praktisch werk

Introduction à la programmation orientée objet (POO). Création de classes. Design patterns.

PEDAGOGISCHE EN TECHNISCHE MIDDELEN

- De gebruikte pedagogische middelen en cursusmethoden zijn voornamelijk: audiovisuele hulpmiddelen, documentatie en cursusmateriaal, praktische oefeningen en correcties van de oefeningen voor praktijkstages, casestudies of reële voorbeelden voor de seminars.
- Na afloop van de stages of seminars verstrekt ORSYS de deelnemers een evaluatievragenlijst over de cursus die vervolgens door onze pedagogische teams wordt geanalyseerd.
- Na afloop van de cursus wordt een presentielijst per halve dag verstrekt, evenals een verklaring van de afronding van de cursus indien de stagiair alle sessies heeft bijgewoond.

TOEGANGSMODALITEITEN EN TERMIJNEN

De inschrijving dient 24 uur voor aanvang van de cursus plaatsgevonden te hebben.

TOEGANKELIJKHEID VOOR MINDERVALIDEN

Is voor u speciale toegankelijkheid vereist? Neem contact op met mevr. FOSSE, contactpersoon voor mindervaliden, via het adres psh-accueil@ORSYS.fr om uw verzoek en de haalbaarheid daarvan zo goed mogelijk te bestuderen.

5 Geavanceerde verwerking van objecten

- Eigenschappen en descriptors.
- Magische methoden.
- Metaklassen en decoratoren.
- Introspectie en reflectie.
- Geheugenbeheer.
- Best practice voor OOP.

Praktisch werk

Properties et descripteurs. Méthodes spéciales. Métaprogrammation.

6 Gevorderde OOP

- Geavanceerde speciale Python-methoden.
- Meervoudige overerving en method order resolution (MRO).
- Contextmanagers en Python-protocollen.
- Geavanceerde ontwerppatronen.
- Mixins en compositie.
- Anti-patronen en veelvoorkomende valkuilen.

Praktisch werk

Méthodes spéciales avancées. Héritage et composition. Design patterns avancés.

7 Geavanceerde foutafhandeling

- Volledige hiërarchie van uitzonderingen.
- Aangepaste uitzonderingen maken.
- Geavanceerde logging en configuratie.
- Debuggen en problemen oplossen.
- Asynchrone foutafhandeling.
- Goede robuustheidspraktijken.

Praktisch werk

Exceptions personnalisées. Logging avancé. Debugging et tests.

8 StdLib gebruiken

- Argumenten en opties op de opdrachtregel (argparse).
- Reguliere uitdrukkingen (re).
- Bestandssysteemmanipulatie (os, pathlib).
- Belangrijke modules (datetime, collections, itertools).
- Databases met sqlite3.

Praktisch werk

Manipulation de fichiers. Expressions régulières. Base de données SQLite.

9 Standaard Python-bibliotheek

- sys en os modules voor systeeminteractie.
- Geavanceerde bestandsmanipulatie met pathlib.
- Gespecialiseerde collecties (collections, heapq).
- Reguliere expressies met re.
- Serialisatie met pickle en json.
- Concurrentie met threading en multiprocessing.

Praktisch werk

Manipulation système. Collections avancées. RegEx et sérialisation.

10 CLI-tools en hulpprogramma's

- Argumenten en opties met argparse.
- Configuratie met configparser.
- Inloggen in CLI.
- Uitgebreide console-interfaces.
- Systeemautomatisering.
- Distributie van CLI-tools.

Praktisch werk

Parsing d'arguments. Interface utilisateur. Packaging et distribution.

11 Testen en documentatie

- Unit testen met unittest en pytest.
- Testgestuurde ontwikkeling (TDD).
- Documentatie met docstrings en typehints.
- Documentatie genereren met Sphinx.
- Code coverage meting.
- Continue integratie.

Praktisch werk

Tests unitaires. Documentation. Couverture et CI.

12 Code kwaliteit

- Hulpmiddelen voor statische analyse (pylint, flake8).
- Stijlconventies (PEP 8).
- Refactoring-technieken.
- Codecontrole en peer programming.
- Maten van complexiteit.
- Geautomatiseerde controles.

Praktisch werk

Analyse statique. Refactoring. Revue de code.

13 Grafische interfaces met Tkinter

- Inleiding tot Tkinter en zijn alternatieven.
- Basis widgets en lay-outs.
- Beheer van evenementen.
- MVC-patroon in GUI.
- Stijlen en thema's.
- Goede interfacepraktijken.

Praktisch werk

Fondamentaux Tkinter. Widgets avancés. Architecture MVC.

14 Geavanceerde interface en GUI testen

- Aangepaste widgets.
- Geavanceerde stijlen en thema's.
- Animatie en visuele effecten.
- Grafische interface testen.
- Toegankelijkheid.
- Internationalisering.

Praktisch werk

Widgets custom. Tests GUI. Internationalisation.

15 Interfacing met C

- Inleiding tot ctypes en de alternatieven.
- C-gegevenstypen en hun mapping.
- C-functies aanroepen vanuit Python.
- Geheugenbeheer en pointers.
- Bibliotheken inpakken C.
- Prestaties en debuggen.

Praktisch werk

Introduction à ctypes. Appels de fonctions. Performance et debug.

16 Optimalisatie met Cython

- Inleiding tot Cython en het ecosysteem.
- Cython syntaxis en statische types.
- Compilatie- en bouwproces.
- Optimalisatie van bestaande Python-code.
- Integratie met C/C++.
- Profileren en benchmarking.

Praktisch werk

Fondamentaux Cython. Optimisation. Integration C/C++

17 Profileren en prestaties

- Profilerings tools (cProfile, line_profiler).
- Analyse van CPU- en geheugenprestaties.
- Algoritmische optimalisatie.
- Benchmarking en metriek.
- Geheugenlekken en afvalverzameling.
- Monitoring in productie.

Praktisch werk

Profilage CPU. Mémoire et GC. Benchmarking.

18 Beste praktijken voor optimalisatie

- Python optimalisatiepatronen.
- Krachtige architecturen.
- Monitoring en observeerbaarheid.
- Geoptimaliseerde inzet.
- Horizontale en verticale schaalbaarheid.
- Belasting- en stresstests.

Praktisch werk

Architecture performante. Tests de charge. Monitoring production.

19 Start van het samenvattingsproject

- Presentatie van projectonderwerpen.
- Ontwikkelingsmethodologie.
- Teamorganisatie.
- Planning en mijlpalen.
- Architectuur en ontwerp.
- Kwaliteitsnormen.

Praktisch werk

Initialisation projet. Architecture. Setup projet.

20 Ontwikkeling - Fase 1

- De basisstructuur opzetten.
- Implementatie van kernfuncties.
- Unit testen en documentatie.
- Git workflow en samenwerking.
- Codecontrole en kwaliteit.
- Voortgangscontrole.

Praktisch werk

Core features. Git workflow. Point d'avancement.

21 Ontwikkeling - Fase 2

- Afwerking van functionaliteiten.
- Integratietesten.
- Gebruikersdocumentatie.
- Voorbereiden op uitrol.
- Optimalisatie en refactoring.
- De demo voorbereiden.

Praktisch werk

Features avancées. Documentation. Préparation démo.

22 Presentatie en beoordeling

- Projectpresentaties.
- Technische demonstraties.
- Feedback.
- Beoordeling van eerdere leerervaringen.
- Perspectieven voor discussie.
- Feedback over training.

Praktisch werk

Présentations. Revue technique. Perspectives. Clôture de la formation.

23 Python 3, de basis van de taal - Post-training digitale leerinhoud

- Inleiding.
- Gegevenstypen.
- Algoritmen.
- Gegevensmanipulatie.

Digitale activiteiten

Deze online cursus introduceert de essentiële basisprincipes van de Python-taal, zodat je efficiënt kunt leren programmeren. Deelnemers leren over programmastructuur, datatypes, functies en algoritmische concepten, voordat ze verder gaan met datamanipulatie en objectgeoriënteerd programmeren. Het gebruik van een database met SQLAlchemy en de toepassing van goede praktijken om schone en onderhoudbare Python-code te ontwikkelen komen ook aan bod.

24 Python 3, concepten voor gevorderden - Digitale leerinhoud na de training

- Objectmodel.
- Typische objecten.
- Testen.
- XML.
- Documenten maken.

Digitale activiteiten

Deze online cursus introduceert het Python objectmodel en getypeerde objecten, een van de moderne ontwikkelassen van Python. Deelnemers zullen in staat zijn om krachtige, moderne toepassingen en veilige gegevensverwerking te bouwen. Ze ontdekken de beste methodes om hun code te testen en de kwaliteit ervan te garanderen. Ze leren ook hoe ze XML kunnen manipuleren met Python en hoe ze PDF-, openDocument- en afbeeldingsdocumenten kunnen genereren.

Data en plaats

KLAS OP AFSTAND

2026 : 9 maa., 1 juni, 14 sep., 23 nov.

PARIS LA DÉFENSE

2026 : 18 mei, 7 sep., 16 nov.

LILLE

2026 : 9 maa., 1 juni, 14 sep., 23 nov.