

Python, Object-georiënteerd programmeren

Praktijkcursus van 5 dagen - 35h

Ref : PYT - Prijs 2025 : 3 020 excl. BTW

Python is een multiplatform-programmeertaal waarmee zeer uiteenlopende toepassingen kunnen worden ontwikkeld. U leert alles over de syntax, de belangrijkste mechanismen en het Object-paradigma. U maakt kennis met de functionaliteiten van de bibliotheek van standaardmodules, leert grafische interfaces implementeren, krijgt toegang tot gegevens uit een database en leert hulpmiddelen gebruiken om de kwaliteit van de geproduceerde code te testen en te beoordelen.

PEDAGOGISCHE DOELSTELLINGEN

Na afloop van de opleiding kan de cursist:

Beheersing van de syntax van de Python-taal

De essentiële kennis over object-georiënteerd programmeren verwerven

Kennis en toepassing van de verschillende Python-modules

Grafische interfaces ontwerpen

Tools implementeren voor het testen en beoordelen van de kwaliteit van een Python-programma

HET PROGRAMMA

laatste update: 02/2024

1) Syntax van de Python-taal

- Identifiers en referenties. Coderingsconventies en naamgevingsregels.
- Blokken, opmerkingen.
- De beschikbare gegevenstypes.
- Variabelen, geformatteerde weergave, lokaal en globaal bereik.
- Bewerking van digitale types, bewerking van strings.
- Bewerking van dynamische arrays (lijst), statische arrays (tuple) en dictionary's.
- Gebruik van bestanden.
- De voorwaardelijke structuur if/elif/else.
- Logische operatoren en vergelijkingsoperatoren.
- De iteratielussen while en for. Onderbreking van iteraties break/continue.
- De range-functie.
- Functies schrijven en documenteren.
- De lambda-uitdrukking.
- Generatoren.
- Structurering van de code in modules.

Installatie van de Python-interpreter en aan de slag ermee.

2) Objectgeoriënteerde aanpak

- De principes van het Object-paradigma.
- De definitie van een object (toestand, gedrag, identiteit).
- Het begrip klasse, attributen en methoden.
- Inkapseling van gegevens.
- Communicatie tussen objecten.
- Overerving, overdracht van de kenmerken van een klasse.
- Het begrip polymorfisme.

DEELNEMERS

Ontwikkelaars, ingenieurs, projectmanagers die dicht bij ontwikkeling staan.

VOORAFGAANDE VEREISTEN

Basiskennis van programmeren.

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.

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.

- Koppeling tussen klassen.
- De interfaces.
- Presentatie van UML.
- Klasse-, sequentie-, activiteitendiagrammen enz.
- Begrip ontwerppatroon (Design Pattern).

Modellering van een eenvoudige casestudy in UML.

3) Object-georiënteerd programmeren in Python

- Bijzondere kenmerken van het Python Object-model.
- Klassen schrijven en instantiëren.
- Bouwers en slopers.
- Toegangsbeveiliging van attributen en methoden.
- De noodzaak van de parameter Self.
- Enkelvoudige overerving, meervoudige overerving, polymorfisme.
- De grondbeginselen van zichtbaarheid.
- Speciale methoden.
- Introspectie.
- Implementatie van interfaces.
- Goede praktijken en gangbare ontwerppatronen.
- Gebruik van het uitzonderingsmechanisme voor het beheer van fouten.

Oefenen van de verschillende Object-concepten door uitvoering van de casestudie.

4) Gebruik van StdLib

- De argumenten die zijn doorgegeven op de opdrachtregel.
- Gebruik van de engine met reguliere uitdrukkingen van Python met de "re"-module, speciale tekens, kardinaliteiten.
- Bewerking van het bestandssysteem.
- Presentatie van enkele belangrijke modules van de standaardbibliotheek: module "sys", "os", "os.path".
- Inpakking en installatie van een Python-bibliotheek.
- Toegang tot relationele databases, werking van de DB API.

Implementatie van Python-modules: reguliere uitdrukkingen, toegang tot een database,

5) QA-tools

- Statische codeanalysetools (Pylint, Pychecker).
- Analyse van analyserapporten (soorten berichten, waarschuwingen, fouten).
- Automatische extractie van documentatie.
- De Python-debugger (stapsgewijze uitvoering en post-mortem analyse).
- Testgestuurde ontwikkeling.
- Python unit test modules (Unittest...).
- Automatisering van tests, samenvoeging van tests.
- Code coverage tests, profiling.

Gebruik van de tools pylint en pychecker om een Python-code te verifiëren. Uitvoering van unit tests.

6) Creatie van TkInter HMI

- Programmeerprincipes van grafische interfaces.
- Presentatie van de TkInter-bibliotheek.
- De belangrijkste containers.
- Presentatie van de beschikbare widgets (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...).
- De vensterbeheerder.
- De plaatsing van de componenten, de verschillende lay-outs.
- Gebeurtenissenbeheer, het gebeurtenisobject.
- Toepassingen met meerdere vensters.

Ontwerpen van een grafische interface met de Tkinter-bibliotheek.

7) Python/C interfacing

- Presentatie van de module Ctypes.
- Laden van een C-bibliotheek.
- Een functie aanroepen.
- Herschrijven van een Python-functie in C met de Python/C API.
- Aanmaken van C-modules voor Python.
- De Python interpreter in C.
- Gebruik van de code profiler.

Aanroepen van in C geschreven functies vanuit Python. Aanmaken van C-modules voor Python met Pyrex.

8) Conclusie

- Kritische analyse van Python.
- De evolutie van de taal.
- Webografie- en bibliografie-elementen.

DATA

KLAS OP AFSTAND
2025 : 18 août, 03 nov.

BRUSSEL
2025 : 18 août, 03 nov.