

Opleiding : Ontwikkelen in objectgeoriënteerde Python, API Society-certificering

Praktijkcursus - 4d - 28u00 - Ref. YPT

Prijs : 2630 € V.B.



4,8 / 5

Python is een programmeertaal die veel gebruikt wordt in webapplicaties, softwareontwikkeling, data science, financiën, cartografie, kunstmatige intelligentie en machine learning. In deze cursus leer je de basissyntaxis, verschillende gegevenstypen, voorwaardelijke en herhalende structuren, functies, modules, pakketten, bestandsmanipulatie en het afhandelen van uitzonderingen.



Pedagogische doelstellingen

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

- ✓ De syntaxis van de taal Python kennen
- ✓ Fundamentele gegevenstypen kennen en manipuleren
- ✓ Functies, modules, pakketten, uitzonderingen gebruiken en definiëren en bestanden manipuleren
- ✓ De theorie van objectgeoriënteerd programmeren begrijpen en deze toepassen in Python
- ✓ De essentiële modules van de standaardbibliotheek kennen en gebruiken

Doelgroep

Ontwikkelaars, ingenieurs, projectmanagers die dicht bij de ontwikkeling staan.

Voorafgaande vereisten

Kennis van een programmeertaal.

Certificatie

Het certificeringsexamen vindt online, offline en in het Frans plaats in de maand volgend op de cursus. Het bestaat uit een theoretische test van 20 minuten - 40 waar/onwaar MCQ-vragen en in te voeren informatie, en een praktische programmeertest (code-oefening).

DEELNEMERS

Ontwikkelaars, ingenieurs, projectmanagers die dicht bij de ontwikkeling staan.

VOORAFGAANDE VEREISTEN

Kennis van een programmeertaal.

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.

Praktische modaliteiten

Praktisch werk

Individueel en groepspracticum, collectieve reflectie.

Leer methodes

Actieve lesmethoden die persoonlijke betrokkenheid en uitwisselingen tussen deelnemers aanmoedigen.

Opleidingsprogramma

1 Inleiding

- Geschiedenis (auteur, datum eerste versie).
- Versies van Python (takken 2 en 3).
- Taaleigenschappen (multi-paradigma, sterke dynamische typografie, duidelijke syntaxis).
- Overzicht van de standaardbibliotheek.
- Uitbreidingsmodules en pip control.
- Hoe de PYC bytecode-interpreter werkt.
- Officiële CPython-interpreter en andere interpreters (micropython, brython, pypy, numba).
- Hulbronnen (python.org website, toegang tot documentatie).
- Help() functie en documentketens.
- Principe van inspringen om instructieblokken af te bakenen.
- Commentaar.
- Gereserveerde sleutelwoorden.
- Naamgevingsconventies.
- Interactieve tolk.
- Op zichzelf staand programma.
- Basis ingebouwde functies: print(), type(), input(), len().

2 Niet-aanpasbare gegevenstypen

- Het nut van niet-wijzigbare types (geheugenoptimalisatie), id() en hash() functies, is operator.
- Principe van geordende reeksen (str, tuple en list) en verzamelingen (dict, set).
- Booleaanse (bool), True en False objecten.
- Getallen (int, float, complex), constructoren, operatoren >>, <<, |, &, ^, //, % en **.
- Exponentiële, binaire, octale en hexadecimale notatie, hex(), oct(), bin(), chr(), ord() functies.
- Unicode string (str), definitie met enkele en dubbele aanhalingstekens, multiline strings, raw mode.
- Positieve en negatieve signalering, slice, +, * en in operatoren, iteratie.
- Essentiële str-methodes: split(), replace(), lower(), upper(), strip(), join().
- Opgemaakte tekenreeks (%s, %d, %f), methode format() en f-string.
- Principe van variabele depacking.
- Array van bytes, fabrikant.
- Tuple (tuple), constructor, indexering, iteratie, +, * en in operatoren, methoden count() en index().
- Geen object en repr() functie.

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.

3 Aanpasbare gegevenstypen

- Lijsten, constructor, indexering, integratie, +, * en in operatoren.
- Append() en insert() methoden, del() functie, sort(), reverse(), remove(), extend(), pop() en clear() methoden.
- Manipulatie van pointers, oppervlakkig kopiëren met de methode copy() of [:] markers.
- Studie van diep kopiëren met de functie deepcopy() in de kopieermodule.
- Gesorteerde functie.
- Hoe iterabele objecten werken.
- Omgekeerde() en bereik() functies.
- Woordenboeken (dict), constructor, indexering, in operator, del() functie.
- Manipulatie van pointers, oppervlakkig kopiëren met de methode copy();
- Diepe kopieeranalyse met de deepcopy()-functie van de kopieermodule.
- Set, constructor, operatoren - | & en ^.

4 Voorwaardelijke en repetitieve structuren

- Voorwaardelijke structuur if ... elif ... else.
- Ternary en morse operator.
- Repetitieve structuur.
- Repetitieve structuur voor.
- Instructies pauzeren en voortzetten.
- Functie opsommen.
- Blok anders op repetitieve structuur.
- Intensie lijst (comprehension list), intentie woordenboek (comprehension dict).

5 Functies, modules en pakketten

- Een functie definiëren en aanroepen.
- Lokale, globale en voorgedefinieerde naamruimtes (__builtins__) en dir() functie.
- Retourwaarden, , retourinstructie.
- Generieke functies (eend typen).
- Standaardwaarden.
- Doorgeven door label.
- Willekeurig aantal argumenten (*args, **kwargs).
- Anonieme (lambda) functies.
- eval(), exec(), map() en filter() functies.
- Modules importeren.
- Een module maken.
- Blok als __name__ == "__main__".
- Pakket importeren.
- Aanmaken van een pakket (__init__.py).
- Instructie opbrengst.

6 Bestandsverwerking

- Open() functie en sluit() methode.
- Readline() en readlines() methoden.
- Iterable object.
- Instructie met bestanden.
- Lees() en schrijf() methoden.
- Tell() en seek() methoden.
- Methode Schrijfregels().
- Aanvullende modules: struct, csv, json, xml.
- Serialisatie met de pickle-module.
- Serialisatie met de rekkenmodule.

7 Objectgeoriënteerd programmeren

- Fundamentele concepten van OOP (scheiding van code, inkapseling, overerving).
- Begrippen objectklasse, object (instantie), attribuut en methode.
- Definitie van een objectklasse.
- Objectinstantie, functie `isinstance()`.
- Constructor (`__init__`).
- Eigenschappen en methoden.
- De zelfparameter.
- Overbelasting weergeven (`__str__`).
- Operator overload (`__eq__`, `__add__`).
- Property (speciale eigenschapfunctie), accessor en mutator.
- Globale, object en klasse namespaces.
- Klassevariabele.
- Constructor met willekeurig aantal argumenten (`*args`, `**kwargs`).
- Aggregatie / Samenstelling.
- Klasse-overerving (generalisatie), `issubclass()`, `super()` functies en `mro()` methode.

8 Uitzonderingen

- Hoe het werkt.
- Vooraf gedefinieerde uitzonderingen en overervingsstructuur.
- Instructies proberen ... behalve ... anders ... tot slot.
- Voortplanting van uitzonderingen.
- Uitzonderingen activeren.
- Een uitzondering definiëren.

9 Standaardbibliotheekmodules

- Interactie met de interpreter: module `sys`.
- Interactie met het besturingssysteem: `os` en `pathlib` modules...
- Interactie met het bestandssysteem: `os.path` module.
- Reguliere uitdrukkingen: `re` module, `search()`, `match()`, `split()` en `sub()` functies.
- Unit tests: `assert` statement, `unittest` module.
- Tour d'horizon de modules de la bibliothèque standard : `datetime`, `math`, `timeit`, `urllib`, `collections`, `csv`, `json`, `sqlite3`.
- Introductie tot `datetime`, `subprocess`, `shutil`, `collections`, `timeit`, `urllib`, `sqlite3`...

Data en plaats

KLAS OP AFSTAND

2026 : 16 juni, 29 sep., 15 dec.

PARIS LA DÉFENSE

2026 : 9 juni, 22 sep., 15 dec.