# Python

Python este un limbaj de programare dinamic multi-paradigmă, creat în 1989 de programatorul olandez Guido van Rossum. Van Rossum este și în ziua de astăzi un lider al comunității de dezvoltatori de software care lucrează la perfecționarea limbajul Python și implementarea de bază a acestuia, CPython, scrisă în C. Python este un limbaj multifuncțional folosit de exemplu de către companii ca Google sau Yahoo! pentru programarea aplicațiilor web, însă există și o serie de aplicații științifice sau de divertisment programate parțial sau în întregime în Python. Popularitatea în creștere, dar și puterea limbajului de programare Python au dus la adoptarea sa ca limbaj principal de dezvoltare de către programatori specializați și chiar și la predarea limbajului în unele medii universitare. Din aceleași motive, multe sisteme bazate pe Unix, inclusiv Linux, BSD și Mac OS X includ din start interpretatorul CPython.  
Python pune accentul pe curățenia și simplitatea codului, iar sintaxa sa le permite dezvoltatorilor să exprime unele idei programatice într-o manieră mai clară și mai concisă decât în alte limbaje de programare ca C.  
În ceea ce privește paradigma de programare, Python poate servi ca limbaj pentru software de tipul object-oriented, dar permite și programarea imperativă, funcțională sau procedurală. Sistemul de tipizare este dinamic iar administrarea memoriei decurge automat prin intermediul unui serviciu „gunoier” (garbage collector). Alt avantaj al limbajului este existența unei ample biblioteci standard de metode.  
Implementarea de referință a Python este scrisă în C și poartă deci numele de CPython. Această implementare este software liber și este administrată de fundația Python Software Foundation.  
  
  
== Sintaxă și Implementare ==  
  
  
=== Tipizare ===  
Limbajele dinamice sunt deseori caracterizate prin tipizare dinamică, sunt interpretate, au management de memorie automatizat (garbage collecting) și au un nivel înalt de abstractizare.  
Tipul variabilelor nu este fixat la momentul declarării acestora (de fapt, declararea variabilelor nu este necesară ca în C), ci este determinat de interpretator după conținutul lor sau după operațiile efectuate. Într-un asemenea caz se vorbește de tipizare dinamică.  
Avantajul acestei metode de a lucra cu tipuri este că codul produs este mai clar și mai intuitiv. De exemplu:  
  
În ciuda tipizării dinamice, Python este puternic tipizat, adică interpretatorul nu permite operații cu obiecte de tip diferit dacă acestea nu sunt bine definite, cum ar fi operațiile de tip cast în C. Există totuși unele excepții logice, cum ar fi în cazul numerelor întregi și celor raționale (float), între care sunt permise operații ca adunarea sau împărțirea, tipul rezultatului ținând cont de natura operației, cum se poate vedea mai jos:  
  
Alt concept important în Python este cel al tipurilor mutabile și nemutabile. Așa cum implică numele, datele cu un tip mutabil pot fi alterate după inițializare, în vreme ce la date nemutabile lucrul acesta este imposibil. Luând ca exemplu o listă (mutabilă), constatăm că putem adăuga elemente la listă sau le putem modifica fără probleme. În cazul unui șir de caractere (nemutabil), acest lucru nu este permis:  
  
  
=== Structuri de date ===  
  
Python oferă tipuri tradiționale de date, cum ar fi numărul întreg (integer, int) sau cel rațional (float), dar introduce totodată și concepte noi.  
De exemplu, un grad mare de flexibilitate îl oferă listele (tablourile) în Python. Acestea nu sunt statice, ci pot conține orice tipuri de date (în cadrul aceleiași liste) și pot fi modificate pe loc adăugând și eliminând elemente fără a declara sau utiliza funcții de manipulare a memoriei.  
  
Alte structuri de date sunt tuplurile/perechile (tuples) și dicționarele (dictionaries, mappings). Tuplurile sunt liste care au un număr prestabilit de elemente, și nu pot fi modificate parțial. Tuplurile pot fi utilizate în cazuri în care este nevoie de o anumită structură de date specializată, de exemplu coordonate în spațiul cartezian. Dicționarele sunt liste neordonate în care fiecare element are asociat o cheie, care poate fi număr sau șir de caractere. Dicționarele au foarte multe aplicații, inclusiv crearea structurilor de tip hash-tables.  
  
  
=== Stil ===  
  
Python este un limbaj multi-paradigmă, concentrându-se asupra programării imperative, orientate pe obiecte și funcționale, ceea ce permite o flexibilitate mai mare în scrierea aplicațiilor.  
Din punctul de vedere al sintaxei, Python are un număr de construcții și cuvinte cheie cunoscute oricărui programator, dar prezintă și un concept unic: nivelul de indentare are semnificație sintactică. Blocurile de cod sunt delimitate prin simplă indentare.  
În C un astfel de blocuri sunt deseori desemnate prin acolade, {<cod>}, dar în Python nu este nevoie de astfel de construcții. Nivelele de indentare îndeplinesc această funcție. Această importanță a indentării este foarte suprinzătoare pentru mulți utilizatori noi ai limbajului Python, chiar dacă sunt programatori cu experiență. Dar o astfel de utilizare a indentării permite codului să fie mai ușor de citit și mai compact. Programatorii cu experiență vor indenta implicit codul sursă, oricare ar fi limbajul, fiindcă astfel se permite structurarea codului sursă și evidențierea funcționalității. Python face din această deprindere folositoare în acest sens o cerință strictă.  
O impunere similară există și în limbajul de programare Java, care forțează programatorii să delimiteze clasele în fișiere aparte, din motive de organizare și sporire a eficienței de scriere a softului în echipe.  
  
  
=== Biblioteci și Extindere ===  
  
Includerea tuturor acestor structuri, precum și a funcțiilor ce permit manipularea și prelucrarea lor, precum și multe alte biblioteci de funcții sunt prezente datorită conceptului “Batteries Included”, ce poate fi explicat prin faptul că Guido van Rossum și comunitatea ce s-a format în jurul limbajului cred că un limbaj de programare nu prezintă utilitate practică dacă nu are un set de biblioteci importante pentru majoritatea dezvoltatorilor.  
Din acest motiv Python include biblioteci pentru lucrul cu fișiere, arhive, fișiere XML și un set de biblioteci pentru lucrul cu rețeaua și principalele protocoale de comunicare pe internet (HTTP, Telnet, FTP). Un număr mare de platforme Web sunt construite cu Python. Abilitățile limbajului ca limbaj pentru programarea CGI sunt în afara oricăror dubii. De exemplu YouTube, unul din site-urile cu cea mai amplă cantitate de trafic din lume, este construit pe baza limbajului Python.  
Totuși, Python permite extinderea funcționalității prin pachete adiționale programate de terți care sunt axate pe o anumită funcționalitate. De pildă, pachetul wxPython conține metodele și structurile necesare creării unei interfețe grafice.  
Popularitatea limbajului este în creștere începând cu anul 2000, datorită faptului că Python permite crearea mai rapidă a aplicațiilor care nu cer viteze înalte de procesare a datelor. De asemenea este util ca limbaj de scriptare, utilizat în cadrul aplicațiilor scrise în alte limbaje. Modulele (bibliotecile) Python pot fi de asemenea scrise în C, compilate și importate în Python pentru a mări viteza de procesare.  
  
  
== Vezi și ==  
Listă alfabetică de limbaje de programare  
Platforma LAMP  
  
  
== Legături externe ==  
python.org:  
Tutorial oficial Python în română  
MITx - Introduction to Computer Science and Programming Using Python