**PYTHON**

Python – uniwersalny język programowania wysokiego poziomu , z szeroką gamą standardowych pakietów bibliotek, główną ideą jest czytelność i przejrzystość kodu źródłowego. Jego składnia jest jasna i zwięzła. Python obsługuje różne paradygmaty programowania: zorientowane obiektowo, imperatywne i mniej funkcjonalne. Posiada w pełni dynamiczny system typów i automatyczne zarządzanie pamięcią. Python został opracowany jako projekt open source i zarządzany przez Python Software Foundation, która jest organizacją non-profit. Standardową implementacją języka jest CPython (napisany w C), ale istnieją inne implementacje, takie jak Jython (napisany w Javie), CLPython napisany w Common Lisp, IronPython (napisany w .NET) i PyPy (napisany w Pythonie) .

**Rozwój języka**

Python został stworzony przez Guido van Rossuma na początku lat 90. jako następca języka ABC. Van Rossum jest głównym twórcą Pythona, chociaż wiele wkładów w jego rozwój pochodzi od innych. Ze względu na kluczową rolę Van Rossuma w podejmowaniu ważnych decyzji projektowych, często nazywany jest „ Benevolent Dictator for Life”. Nazwa języka pochodzi z serialu komediowego BBC emitowanego w latach 70. „Monty Python’s Flying Circus”. Projektant był fanem serii, szukał krótkiej, niepowtarzalnej i nieco tajemniczej nazwy i uznał ją za świetną. Wersja 1.2 to ostatnia wersja wydana przez CWI. Począwszy od 1995 roku Van Rossum kontynuował pracę nad badaniami Pythona w National Research Projects Corporation (CNRI) w Reston w stanie Wirginia, gdzie wydał wiele wersji Pythona, w tym 1.6. W 2000 roku van Rossum i zespół programistów Pythona przenieśli się do BeOpen.com, tworząc zespół BeOpen PythonLabs.

Pierwszą i jedyną wersją wydaną przez BeOpen.com jest Python 2.0. Wraz z wydaniem wersji 1.6 zaczęto wykorzystywać do programowania komercyjnego, van Rossum miał nadzieję, że Python będzie mógł być używany z oprogramowaniem na licencji GPL. CNRI i Free Software Foundation współpracowały, aby odpowiednio zmodyfikować licencję Pythona. Wersja 1.6.1 jest zasadniczo taka sama jak wersja 1.6, z wyjątkiem kilku drobnych poprawek i licencji, aby wyższe wersje były zgodne z GPL. Python 2.1 wywodzi się z wersji 1.6.1 i 2.0. Po wydaniu przez BeOpen.com Pythona 2.0, Guido van Rossum i inni programiści PythonLabs zwrócili się w stronę Digital Creations. Od tego czasu wszystkie prawa własności intelektualnej dodane z Pythona 2.1 należą do Python Software Foundation, organizacji non-profit wzorowanej na Apache Software Foundation.

**Filozofia Pythona**

Python implementuje jednocześnie kilka paradygmatów. Podobny do C++, ale w przeciwieństwie do Smalltalk, nie wymusza jednego stylu programowania, ale pozwala na różne style programowania. W Pythonie możliwe jest programowanie obiektowe, strukturalne i funkcjonalne. Typy są sprawdzane dynamicznie, a do zarządzania pamięcią używane jest wyrzucanie elementów bezużytecznych. Chociaż nacisk kładziony jest na różnice w stosunku do Perla pod względem popularności, Python jest podobny do Perla pod wieloma względami. Jednak projektanci Pythona porzucili złożoną składnię Perla na rzecz bardziej zwięzłej i czytelnej składni w ich oczach. Chociaż podobny do Perla, Python jest czasami klasyfikowany jako język skryptowy, ale służy do tworzenia dużych projektów, takich jak serwer aplikacji Zope, system udostępniania plików Mojo Nation, a nawet oprogramowanie Odoo ERP.

**Typy i struktury danych**

Python jest językiem dynamicznie typowanym, podobnie jak Lisp, w przeciwieństwie do np.: Java. W przeciwieństwie do wielu języków, wartości nie są przekazywane ani przez wartość, ani przez referencję, ale przez przypisanie. W porównaniu z innymi językami programowania Python jest silnie typowany. Nie jest ani tak wolny jak Perl, ani tak surowy jak OCaml. Reguły gramatyczne Pythona pozwalają na wyrażanie pojęć bez pisania dodatkowego kodu. Automatyczna konwersja jest zdefiniowana dla typów liczbowych, więc liczbę zespoloną można pomnożyć przez liczbę całkowitą typu long bez rzutowania. Ale w przeciwieństwie do Perla, na przykład, nie ma automatycznej konwersji między ciągami a liczbami; liczba nie jest prawidłowym parametrem dla operacji na ciągach. Python zapewnia szeroki zakres podstawowych typów danych, w tym typy numeryczne (liczby całkowite, zmiennoprzecinkowe, liczby zespolone) i zbiory.

**Składnia**

Python dąży do tego, aby kod źródłowy był jak najbardziej czytelny. Ma prosty układ tekstu, używa wcięć lub angielskich słów tam, gdzie inne języki używają interpunkcji, i ma mniej struktury składniowej niż wiele języków strukturalnych (takich jak C, Perl czy Pascal). Dla czytelności w Pythonie są tylko dwie pętle: for, która iteruje przez elementy listy (takie jak foreach Perla) i dopóki warunek logiczny jest prawdziwy, powtarza się. Python nie ma gramatyki w stylu C, **do...while** ani perlowego **until**.

**Inne cechy**

Interpreter Pythona ma również tryb interaktywny, w którym można wprowadzać wyrażenia z terminala i natychmiast uzyskiwać wyniki. Zgodnie z założeniami twórców Pythona ma to ułatwić naukę programowania, ponieważ pozwala od razu wypróbować fragmenty kodu. Jednak standardowa powłoka nie jest zbyt wygodna i nie ma dużej funkcjonalności - IPython, który jest częścią pakietu SciPy, nie ma tych wad.