

C++ vs Python

1) Kompilering

C++ er et kompilert språk. C++ kompilator genererer en objektkode fra C++ kildekoden og blir deretter utført for å produsere utdata.

Python er et tolket språk. Python-koden med en utvidelses-py trenger ikke kompiles. Vi kan sende den direkte til Python-tolk og generere utdata.

2) Bruk

C++ har mange funksjoner og har også en relativt vanskelig syntaks. Det er ikke så enkelt å skrive C++ - koden.

Python er lett å skrive og har en tydelig syntaks. Derfor er det lettere å skrive Python-programmer sammenlignet med C++.

3) Språkets art

C++ er et statisk skrevet språk, det vil si at erklæringen til en variabel, datatypen på variabler osv. Blir bekreftet ved kompileringstidspunktet. Dette holder kildekoden feilfri ved kjøretid.

Python er derimot ikke statisk skrevet. Det er ingen typekontroll gjort på kompileringstidspunktet. Derfor er koden utsatt for feil.

4) Bærbarhet

C++ er ikke bærbar, det vil si at vi trenger å kompilere koden på hver annen plattform. C++ er hovedsakelig "Skriv en gang, kompilér hvor som helst".

Python er bærbar. Det er også plattformoverskridende, og vi kan utføre programmer på hvilken som helst plattform.

5) Søppelinnsamling / minnehåndtering

I C++ er minneadministrasjon manuell. C++ støtter ikke automatisk søppelinnsamling av ressurser.

Python, derimot, har en funksjon av automatisk søppelopsamling.

Hukommelsesadministrasjonen er systemstyrt.

6) Rask prototyping

Vi kan ikke gjøre rask prototyping ved bruk av C++.

Ved hjelp av Python kan vi gjøre rask prototyping av kode slik at den kan brukes senere til å bygge applikasjoner som bruker språk på høyere nivå.

7) Omfanget av variablene

C++ har kode avgrenset av blokker som bruker krøllete bukseseler ({}) og sløyfer. Omfanget av variablene er begrenset til disse blokkene og sløyfene som er avgrenset av {}.

Omfanget av variablene som brukes i Python er ikke begrenset til blokker eller sløyfer.

Variablene er tilgjengelige selv utenfor krøllete bukseseler.

8) Installasjon

C++ kan enkelt installeres på Windows. Python er imidlertid vanskelig å installere.

C++ vs Python

9) Typer

I C++ er datatyper bundet til navn og blir sjekket ved kompileringstidspunktet. Dette reduserer muligheten for feil ved kjøretid.

I Python er datatypene bundet til verdier og sjekkes ved kjøretid. Koden kan være mer utsatt for feil ved kjøretid, da vi ikke får tak i disse feilene på kompileringstidspunktet.

10) Funksjoner

Funksjoner er blokker av koder med en eller flere parametere og en returverdi. Hver av parametere og returverdien har en type.

I C++ må parametertypene og returtypen under funksjonsanropet stemme overens med det i definisjonen av funksjonen.

I Python er det ingen slik begrensning på parameter- og returtyper.

11) Effektivitet

C++ - kode er vanskelig å vedlikeholde, da den kan bli komplisert å lese når løsningene blir større.

Python, derimot, har ren kode og enkel syntaks. Kildekoden for Python er lettere å vedlikeholde.

12) Syntaks-kompleksitet

I C++ er det en tydelig avgrensning av koden ved å bruke blokker innelukket i {}, semikolon som indikerer slutten på utsagnet, osv. Dermed er syntaksen i C++ godt organisert.

I Python er det ingen blokker eller semikolon. I stedet bruker Python innrykk.

13) Utførelses hastighet

Når det gjelder kjøringshastighet, kjører C++ - programmer raskere. Faktisk er C++ kjent og brukt mye i applikasjoner som kreves for å kjøre raskere som spillplattformer.

Python, derimot, løper sakte. Videre kjører Python-programmer langsommere enn Java-programmer. Derfor bruker vi Python spesielt for applikasjoner som kan kompromittere hastigheten.

14) Ytelse

C++ er et statisk skrevet språk, og dermed har vi færre feil å ta vare på ved kjøretid. C++ skaper også en mer robust og raskere kjøretidskode. Dette gjør C++ til et språk med høy ytelse.

Python er dynamisk, har en mulighet for at noen feil eller uønskede situasjoner kan oppstå ved kjøretid. Så når det gjelder ytelse, henger Python etter C++.

Men når det gjelder maskinlæring, er Python den som har overtaket.

15) Popularitet

Python er lett å lære og lett å praktisere sammenlignet med C++, som blir vanskeligere når vi går videre gjennom funksjonene. En annen fordel med Python er dets biblioteker som lar oss skrive hvilken som helst funksjonalitet, spesielt dataanalyse og maskinlæring.

Så popularitetsmessig scorer Python over C++. Spesielt for utvikling av maskinlæringsapplikasjoner er det det beste valget for programmerere.

C++ vs Python

16) Enkelhet og brukervennlighet

Python med sin enkelhet og brukervennlige funksjoner lar oss skrive kortfattet, lett lesbar kode osv. Dette er nyttig når vi utvikler komplekse applikasjoner for maskinlæring, ettersom vi ikke trenger å slite med et programmeringsspråk.

For det andre er Python lett å lære og er et enkelt språk. Det samme kan ikke sies om C ++. C ++ er mer et lavere nivå som er lettere for datamaskiner enn mennesker.

Dermed scorer Python på disse parameterne, spesielt når vi må velge mellom C ++ og Python for å utvikle maskinlæringsapplikasjoner.

I blokkprogrammering brukes ferdig blokker med kode som legges etter hverandre linje for linje. Blokkene i hver linje kan kombineres med andre blokker. Enkelte blokker inneholder felt for tekst og tall. Fordelen med blokkprogrammering er at skrivefeil er umulig.

Sammenligningsparameter	C ++	Python
Samling	Kompilert	Tolket
Bruk	Ikke lett å skrive kode.	Lettere å skrive kode.
Språkets art	Statisk skrevet	Dynamisk skrevet
Bærbarhet	Ikke bærbar	Bærbar
Søppelsamling	Støtter ikke søppelinnsamling.	Støtter søppelinnsamling.
Installasjon	Ingen vanskeligheter	Vanskelig å installere
Typer	Dat typer bundet til navn som er sjekket på kompileringstidspunktet.	Bundet til verdier, sjekket ved kjøretid.
Omfanget av variablene	Begrenset innenfor løkkene eller blokkene.	Tilgjengelig utenfor løkkene eller blokkene.
Hurtig prototyping	Ikke mulig	Mulig
Funksjoner	Restriksjoner på typen parametere eller returverdi.	Ingen begrensninger på typen parametere eller returverdi.
Effektivitet	Vanskelig å vedlikeholde.	Enklere å vedlikeholde
Syntaks kompleksitet	Bruker blokker og semikolon.	Ingen blokker eller semikolon.
Utførelses hastighet	Raskere	Tregere
Opptreden	Høy ytelse	Lav ytelse
Popularitet	Mer populært for innebygde applikasjoner eller bedriftsapplikasjoner.	Mest populær for maskinlæring.
Enkelhet og brukervennlighet	Vanskelig å lære og brukes i et lavt nivå applikasjon.	Enkelt og brukes til maskinlæring eller webapplikasjoner.