Python

Är et interpreterat programmeringstad. Högnivåprogrammeringpråk stöder både funktinoel och objektorienterad programering. Har dynamisk typning. Kan köras i två lägen. Som skript eller interaktivt. Finns många olika bilbiotek för vetenskapliga tillämpingar mm.

Lågnivå programeringspråk ligger nära maskin, nollor och ettor. Högnivå är lätt att att förstå för oss och kräver mer av datorn att förstå.

Interpreterat programmeringsspråk. Är en uppdelning och skiljs från Kompilerade där man använder textredigerare eller IDE – man skriver källkod sen används kompilator som översätter till maskinkod (binär). Därefter länkare som skapar körbar exe fil från det.

Graphical user interface

Description automatically generated

Interpretearde språk som python och mathlab kan man också använda texteditor eller IDE för skriva Sedan används interpretator som utför programmet rad efter rad. Det skapar således inte binär kod. De kallas därför skriptspråk istället för programmeringsspråk.

A screenshot of a computer

Description automatically generated with medium confidence

Finns fördelar och nackdelar med dessa. Olika snabba att köra och översäatta, Läsa och förstå. Olika specifika till processorerna.

Table

Description automatically generated with low confidence

När man väl har kompilerat ett program måste man inte kompiler det igen, mwn swr äe bweownsw OS. Måste också ta hand om fel efter kompliering. Och även om det är effektivare (snabbt), ligger hela i datorminnet så tar mer RAM.

De rekomenderar att man använder Visal studio code för man kan köra som skript och som interaktivt.

Dynamisk typning innebär att man inte måste deklerar vad variablerna har för datatyp utan den fattar vad som ät int och vad string

**Print** har lite oliak formateraringar. =end” ger till exempel att det skrivs på samma rad och (f” ...”) fsträng som placeras med objekt i klamerparanteser. Ex Print(f”Multiplikation av {a} och {b} är {c}”). Där är bokstäverna objekt.

I python kan man också skriva saker på samma rad ex a, b = 3, 5

Graphical user interface, application

Description automatically generated

**= input( ” text ”)** Programmet stoppar in det användaren ger. Blir sträng så måste omvandla till numeriskt tal

**Int(arg)** Omvandlar arg till int.

Kan göra det direkt i sig också

A picture containing graphical user interface

Description automatically generated

f- strängar anges i utskrivningsformat i klammerparantesre efter obejktsnamn följt av kolon. Ex om man skriver {a:5) kommer fältet a vara fem tecken långt om man skriver c:05 kommer tomma tecken ersättas med nollor.

När man arbetar med långa tal kan man använda \_ ex 1\_000\_000 de ignoreras då. Vill man att de ska spereras med komma skriver man {a:, }

**Float(obj:X.Yf)** Gör om till float

X är länd på fält och Y är antal sifforer efter decimalpunkten och f flytt tal. Skriver man e skrivs det i exponentiell form. Skriver man ex {a:5.2f} kommer man få två siffror efter decimalpunkt och 5 tecken totalt

Operatorer!!

Table

Description automatically generated

Om man vill använda utökade tilldelningsoperatorer

A screenshot of a computer

Description automatically generated with medium confidence

Om man vill göra matimatik behöver man Biblioteket math. (trigonometrin är x vinkel i radianer)

**Import bibliteknamn** .

När man vill anävnda det för man sedan skriva biblioteknamn.funktionen exempelvis x = math.pi/6 är i radianer.

**#** Kommentar!!