

C

C on yleiskäyttöinen proseduraalinen ohjelmointikieli, joka tukee jäsentelyä. Kieli hyödyntää staattista lohkonäkyvyyttä sekä tyyppitystä.

Keskitason ohjelmointikieli

C kieli yhdistää sekä matalan että korkean tason kielen piirteet.

Kielen siirrettävyys

C kielellä tehty ohjelmisto voidaan suorittaa hyvin monessa erilaisessa ympäristössä tekemättä tai tekemällä vain pieniä muutoksia koodiin.

Nopeus

Nopeus johtuu kielen matalasta tasosta, pienistä yleiskustannuksista ja tehokkaista tietorakenteista.

Luotto kehittäjään

Esimerkiksi muistivuodon tarkistus, automaattinen roskienkeräys, ajoaikainen tyyppitarkastus ja poikkeuksien käsittely ovat jätetty kehittäjien toteutettavaksi.

C kielen piirteet tukevat varsinkin järjestelmäohjelmointia. Esimerkiksi käyttöjärjestelmät ja ajurit kehitetään usein kyseisellä kielellä. Kielen monikäyttöisyyden takia sitä käytetään myös työpöytäsovelluksien kehityksessä. Varsinkin sovelluksissa jossa optimointi, tehokkuus ja muistinhallinta ovat tärkeässä roolissa. C kielen laaja käyttö järjestelmissä ja kriittisissä applikaatioissa varmistavat sen, että C tulee olemaan vielä pitkään ajankohtainen kieli. Myös se, että monella uudella kielellä kuten Pythonilla ja JavaScriptillä on perustana C kieli, tukee väitettä.

C ohjelmaesimerkki

```
#define pINF (1.0f/0.0f)
#include <stdio.h>
int main() {
    char operator; double left, right;
    scanf("%lf %c %lf", &left, &operator, &right);

    double res = 0.0;
    switch (operator) { //tehdään valittu operaatio
        case '+': res = left + right; break;
        case '-': res = left - right; break;
        case '*': res = left * right; break;
        case '/': res = left / right; break;
        default: printf("%c is not supported", operator);
    }
    if (res == pINF || res == -pINF || res != res) {
        printf("Invalid operation"); return 1; //nollalla jako
    } else {
        printf("%.4lf %c %.4lf = %.4lf",
            left, operator, right, res); return 0;
    }
}
```

JavaScript

JavaScript on kevyt korkean tason ECMAScript-standardin toteuttava ohjelmointikieli, jota hyödynnetään erityisesti modernissa web-kehityksessä.

Mahdollista käyttää palvelimella ja selaimessa

Suosituin JavaScript-moottori tällä hetkellä on Googlen V8, jolla suoritetaan JavaScript-koodia verkkoselaimessa ja palvelimella. Moottoria hyödyntää esimerkiksi suosittu palvelinpuolen runtime-ympäristö Node.js. Koska JavaScript toimii myös selainpuolella, voidaan tehdä osa laskennasta ilman palvelinpuolen kutsuja.

Funktionaaliset ominaisuudet

JavaScriptillä voidaan ohjelmoida funktionaaliseen tapaan käyttämällä nimettömiä funktioita. Funktioita voidaan myös asettaa muuttujiin, ja niille voidaan antaa parametreina funktioita. Funktiot voivat myös palauttaa funktioita.

Yksisäikeinen suoritus

JavaScriptiä suoritettaessa ei voida hyödyntää enempää kuin yhtä säiettä. Jotta samanaikaista suorittamista voidaan toteuttaa hyödynnetään JavaScriptissa esimerkiksi asynkronista ohjelmointia.

JavaScriptin alustariippumattomuus ja kasvava määrä sitä käyttäviä kirjastoja on vahvistanut sen asemaa. Yhä suurempi osa applikaatioista toteutetaan selaimelle, mikä ennustaa JavaScriptille vahvaa asemaa tulevaisuudessakin.

JavaScript ohjelmaesimerkki

```
/* Yksinkertainen laskin. Syötä numero, operaattori ja
   numero */
var left = parseFloat(readLine());
var operator = readLine();
var right = parseFloat(readLine());

var result;
switch(operator) { //tehdään valittu operaatio
  case "+": result = left + right; break;
  case "-": result = left - right; break;
  case "*": result = left * right; break;
  case "/": result = left / right; break;
  default: throw operator + " is not supported";
}
if (Number.isNaN(result)) { // Jos jaetaan nolllalla
  throw "Invalid operation";
}
console.log(result);
```