

Andmed ja avaldised Scratchis

Andmed on informatsiooni esitus kujul, mis võimaldab seda salvestada, töödelda ja edastada programmjuhtimisega seadmete abil.

Andmete puhul käsitletakse nende esitusviisi ja organisatsiooni. Esitusviisi järgi jagunevad andmed liikidesse ja tüüpidesse, organisatsiooni poolest eristatakse skalaarandmeid, andmekogumeid ja objekte.

Sisu

Andmete liigid	3
Märkandmed	3
Arvud	3
Tekstid	3
Tõeväärtused	3
Graafikaandmed	4
Graafikaobjektid	4
Graafilised kujutised	4
Heliandmed	4
Heliobjektid	4
Hetkhelid	4
Avaldised	4
Arvavaldised	5
Tehted ja struktuur	5
Matemaatikafunktsioonid	5
Tekstavaldised	6
Loogikaavaldised	6

Andmete liigid

Siin vaadeldakse Scratchis kasutatavaid andmeliike ja tegevuste täitmist nendega. Tegevused (operatsioonid) esitatakse avaldiste abil. Kasutatavad tehted ja funktsioonid sõltuvad andmete liigist.

Scratchi projektides saab kasutada järgmisi andmeliike:

- märkandmed
- graafikaandmed
- heliandmed.

Märkandmed

Scratchis kasutatakse järgmisi märkandmeid:

- arvud
- tekstid
- tõeväärtused.

Arvud

Arvude jaoks ei ole Scratchis erinevaid vorminguid ja tüüpe, nagu on iseloomulik teistele programmeerimiskeeltele. Tegemist on nõ tavaliste täis- ja reaalarvudega, kusjuures formaalset vahet ei tehta. Murdosa eraldamiseks täisosast tuleks reaalarvudes kasutada punkti. Sisendväljas, kus peaks asuma arv (näiteks [liigu ... sammu], [oota ... sek]) võib reaalarvu sisestamisel kasutada ka koma (muudetakse automaatselt punktiks). Arve võib esitada käsuplokkides ja avaldistes konstantidena ning sisestada klaviatuurilt.



Arväärtusi saab omistada muutujatele ja loendi elementidele ja kasutada viimaseid käskudes.



Arvude jaoks on neli põhitehet: +, -, *, / (liitmine, lahutamine, korrutamine, jagamine) ning võrdlustehted, mis esitatakse vastavate plokkide abil. On olemas teatud valik matemaatilisi funktsioone. Vt „Arvavaldised“.

Tekstid

Tekstandmete väärtus on Scratchis, nagu ka teistes programmeerimissüsteemides, suvaliste märkide jada. Tekste saab kasutada ainult mõnedes plokkides: teadete väljastamiseks, dialoogide korraldamiseks jmt. Tekst kirjutakse konstandina otse plokki, sisestatakse klaviatuurilt, omistatakse muutujale või loendi elemendile. Tekstkonstante ei paigutata mingite piirajate (jutumärkide või ülakomade) vahele nagu seda peab tegema tekstipõhistes programmeerimiskeeltes. Tekstide jaoks on tehted: ühendamine, märkide eraldamine, võrdlemine jmt. Vt „Tekstavaldised“.



Tõeväärtused

Tõeväärtuste ehk loogikasuuste jaoks on ainult kaks väärtust ehk loogikakonstanti: **tõene (true)** ja **väär (false)**. Ilmutatud kujul neid praktilisel ei kasutata. Loogikaväärtused tekivad võrdlustes ja loogikaavaldistes, mida kasutatakse tingimuste esitamiseks. Vt „Loogikaavaldised“.



Graafikaandmed

Scratchis võib esineda kahte liiki graafikaandmeid:

- graafikaobjektid
- graafilised kujutised.

Graafikaobjektid

Graafikaobjektid (vt ka jaotist [Scratchi põhiobjektid](#)) imporditakse graafikafailidest või luuakse Scratchi joonistamisredaktoriga projekti loomise faasis. Formaalselt on tegemist kas lava taustadega või spraitide kostüümidega. Objektide loomiseks võib kasutada **PNG, GIF, JPG, BMP** faile. Objektidega saab määrata käsuplokkide abil erinevaid tegevusi: asukohta ja suuruse muutmine, pööramine, peitmine jm

Graafilised kujutised

Graafilised kujutised luuakse projekti täitmise ajal. Selleks kasutatakse käske gruppidest **Pliiats** ja **Liikumine** (vahel ka **Välimus**). Grupi **Pliiats** käsud võimaldavad määrata pliiatsi omadusi: värv, joone paksus, olek (üleval või all) jm. Grupi **Liikumine** käskude abil muudetakse joonistava spraidi asukohta vastavalt kujutise geomeetria.

Heliandmed

Sarnaselt graafikaandmetele võib Scratchis eristada ka kahte liiki heliandmeid:

- heliobjektid
- hetkhelid ehk helindid

Heliobjektid

Heliobjektid (vt ka jaotist [Scratchi põhiobjektid](#)) on heliklipid, mis imporditakse või lindistatakse projekti loomise ajal. Saab kasutada WAV ja MP3 vormingus faile. Heliklippe saab esitada grupi **Heli** käskudega **[mäangi heli nimi]** ja **[mäangi heli nimi kuni valmis]**.

Hetkhelid

Hetkhelid tekitatakse arvuti poolt vastavate käskude toimetel. Scratchis kasutatakse selleks käske **[mäangi nooti]** ja **[mäangi trummi]**. Mõlemal juhul saab valida erinevaid instrumente, heli tugevust ja tempot.

Avaldised

Avaldise abil antakse eeskiri vajaliku väärtuse leidmiseks: määratakse tehted ja operatsioonid ning nende täitmise järjekord. Üldjuhul koosneb avaldis **operandidest** ja **tehtemärkidest**. Operandideks võivad olla:

- konstandid
- muutujad
- loendite elemendid
- funktsioonid.

Konstandi väärtus esitatakse otse avaldises. Muutuja näidatakse nime abil. Loendi element esitatakse loendi nime ja indeksi abil ning funktsioon funktsiooniviida abil, mis koosneb funktsiooni nimest ja argumentidest, mis võib omakorda olla avaldis.



Lubatud tehtemärgid ja funktsioonid sõltuvad andmete liigist ning võib eristada arv-, tekst- ja loogikaavalisi. Üldjuhul võib kasutada ka segaavaldisi, kuid alati peab arvestama sobivust tehete, funktsioonide ja operandide liikide vahel.

Scratchi avaldiste eripäraks on see, et need moodustatakse graafilistest plokkidest ja neis **ei saa kasutada sulgusid**, mis on tüüpiline tekstipõhiste keeltele.

Arvavaldised

Tehed ja struktuur

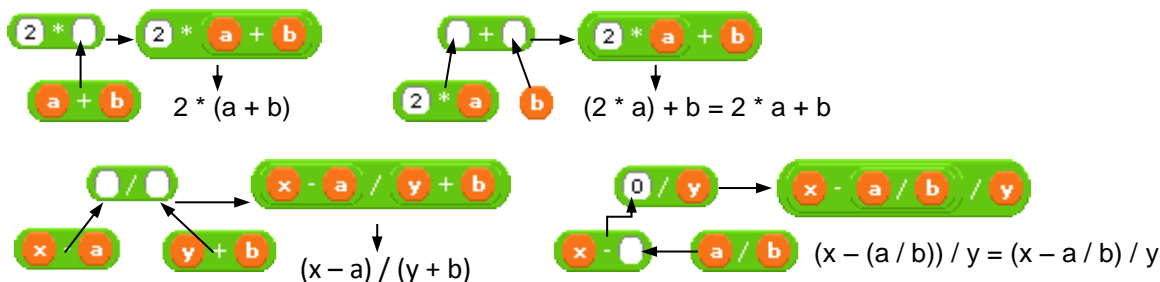
Arvude jaoks on neli põhitehet, mis esitatakse vastavate plokkide abil.



Kuna tehete täitmise järjekorra määramiseks ei saa kasutada sulgusid, vajab avaldiste koostamine mõningat harjumist. Ühes plokkis olev avaldis vastab sulgudes olevale avaldisele.

Näiteks plokk **a + b** vastab avaldisele **(a + b)**.

Järgnevalt on toodud mõned avaldiste koostamise näited.



Kui avaldis kipub minema pikaks ja keeruliseks, võib kasutada abimuutujaid ja lasta arvutada avaldis nõ ositi. Näiteks ülaloleva (mitte küll eriti keerulise) avaldise väärtuse leidmiseks võiks kasutada kahte abimuutujat: **V1 = x-a** ja **V2 = y+b**.

Matemaatikafunktsioonid

Arvandmete jaoks on teatud valik matemaatikafunktsioone. Need esitatakse avaldistes taolise ploki abil:



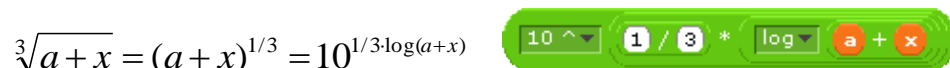
Vasakpoolses väljas saab valida funktsiooni nime:

abs, sqrt, sin, cos, tan, asin, acos, atan, ln, log, e^, 10^.

argument võib üldjuhul olla avaldis, sealhulgas ka muutuja või konstant:



Scratchis ei ole astendamistehet ega ka vastavat funktsiooni. Väikese täisarvulise astendaja korral võib astendamise asendada korrutamisega: **a³ = a*a*a**. Kui tegemist on suvalise astendajaga võib kasutada matemaatikast tuntud seost: **a^x = 10^{x·log₁₀a}**, kus **x** on suvaline reaalarv.

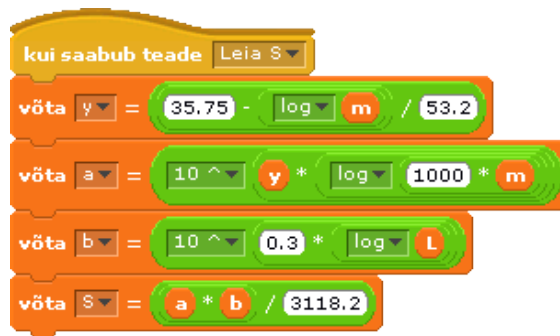


On pakutud järgmine empiiriline valem inimese pindala S (m²) leidmiseks.

$$S = \frac{(1000m)^y \cdot L^{0.3}}{3118,2}, y = (35,75 - \log m) / 53,2$$

kus m on mass (kg) ja L – pikkus (cm)

Kõrval on toodud skript, milles kasutatakse abi-muutujaid a ja b ning astendamist.



Tekstavaldised

Scratchis ei ole eriti palju vahendeid tegevuste täitmiseks tekstidega.

Tekstide ühendamiseks saab kasutada plokki **ühenda** tekst1 tekst2

Mõlemas väljas võivad olla omakorda ühendamisplokid.



Plokk **tekst pikkus** võimaldab teha kindlaks antud teksti pikkuse: märkide arvu tekstis. Tekst võib olla esitatud konstandi, muutuja või avaldise abil.

Käsk **võta m = ühenda eesnimi perenimi pikkus** omistab muutujale m eesnimes ja perenimes olevate tähtede arvu kokku.

Plokk **tekst nr -s täht** tagastab antud numbriga (nr) märgi.

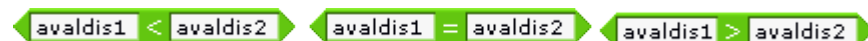


Skript küsib kõigepealt kasutaja käest nime. Kasutaja sisestatud tekst võetakse läbi ploki **vastus** muutuja **nimi** väärtuseks. Ploki **pikkus** abil tehakse kindlaks märkide arv nimes ja omistatakse see muutujale n .

Enne korduse algust võetakse muutuja k (näitab tähe järjenumbrit nimes) väärtuseks **1**. Korduse täitmisel eraldatakse nimest järjest järgmine täht (täht numbriga k) ja kuvatakse see. Korduse lõpus suurendatakse k väärtust ühe võrra, et võtta nimest järgmine täht.

Loogikaavaldised

Loogikaavaldisi kasutatakse tingimuste määramiseks mitmes juhtimisplokis: **kui**, **korda kuni**, **oota kuni**, **lõputult kui**. Loogikaavaldise väärtus on alati kas **tõene** või **väär**. Lihtsamad ja ka enim kasutatavad loogikaavaldised on **võrdlused**, mis esitatakse järgmiste plokkide abil.



Siin on **avaldis1** ja **avaldis2** üldjuhul kas arv- või tekstavaldised, erijuhul võivad olla muutujad või konstandid. Kui võrdlusemärgiga näidatud suhe väärtuste vahel kehtib, on avaldise väärtus tõene, vastupidisel juhul väär. Mõned näited.

```

kui a < b
  võta min = a
muidu
  võta min = b

```

Võrreldakse kahte suurust: **a** ja **b**. Kui **a** on väiksem **b**-st, võetakse muutuja **min** väärtuseks **a**, vastupidisel juhul **b**. Seega saab muutuja **min** väärtuseks väiksem kahest suurusest.

```

küsi Sisesta korrutis? ja oota
kui vastus = a * b
  muuda punkte 1 võrra
muidu
  ütle Vale 2 sekundit

```

Käsk **küsi** loeb kasutaja poolt sisestatava väärtuse ja salvestab selle plokis **vastus**.

Kui-plokis võrreldakse **vastus** väärtust korrutise **a * b** väärtusega. Kui need on võrdsed, suurendatakse muutujat **punkte** ühe võrra, vastupidisel juhul kuvatakse teade **Vale**.

```

kui kaugus hiir -st > 100
  osuta hiir -le
  liigu 5 sammu

```

Kui kaugus spraidist nimega **hiir** on suurem kui 100, siis osutab antud objekt hiirele ja teeb selles suunas 5 sammu.

```

taimer algseisu
korda kuni aeg > 60
  võta aeg = taimer
peata kõik

```

Stopperi töö modelleerimine.

Kõigepealt pannakse **taimer** nulli. Sellele järgneb kordus, kus muutujale **aeg** omistatakse pidevalt **taimer**i jooksev väärtus. Kordus toimib seni, kuni muutuja **aeg** saab suuremaks 60st.

Käsk **peata kõik** lõpetab programmi töö.

Plokk **mitte avaldis**, kus avaldis on loogikaavaldis (enamasti võrdlus), muudab avaldise väärtuse vastupidiseks. Scratchis puuduvad võrdlustehted **<>**, **<=**, **>=**. Neid saab realiseerida plokiga mitte:

```

mitte [ ] = [ ]
mitte [ ] > [ ]
mitte [ ] < [ ]

```

Näiteks ploki **mitte palk > 1000** väärtus on **tõene**, kui **palk <= 1000**.

Plokid **ja** ja **või** võimaldavad ühendada loogikaavaldisi (enamasti võrdlusi) ja moodustada liitningimusi. Näiteks ploki **x > a ja mitte x > b** väärtus on tõene, kui **a < x <= b**.

Ploki **nimi = Juku või nimi = Kalle ja vanus > 12** väärtus on tõene, kui **nimi** on **Juku** või **Kalle** ja **vanus** on suurem 12-st.

Mõned andurid tagastavad ka tõeväärtusi. Näiteks puuteandurid tagastavad väärtuse **tõene**, kui puude on ning **väär** vastupidisel juhul. Kui **pall** puudutab **väravat**, suurendatakse **punkte** arvu ühe võrra.

```

kui puudutab värav ?
  muuda punkte 1 võrra

```

