

## Scratch'is toetatavad programmeerimise kontseptsioonid ja oskused

Interaktiivsete lugude, mängude ja animatsoioonide loomise protsessis Scratch'iga, saab algaja teha endale selgeks mitmed rakenduste loomise ja programmeerimise tähtsad oskused ja kontseptsioonid.

### Probleemide lahendamise ja projektide disaini oskused



- loogiline ja algoritmiline mõtlemine
- süsteemne lähenemine probleemide lahendamisele
- ideede arendus alates lähtekontseptsioonist kuni projekti lõpplahenduseni
- kasutajaliideste loomise oskused ja kogemused
- silumise ja testimise vilumused
- keskendumisvõime ja visaduse arendamine

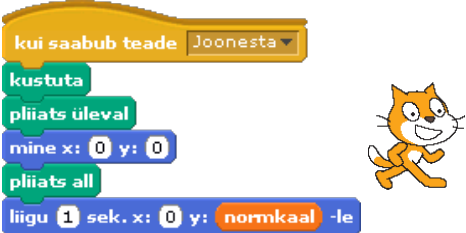

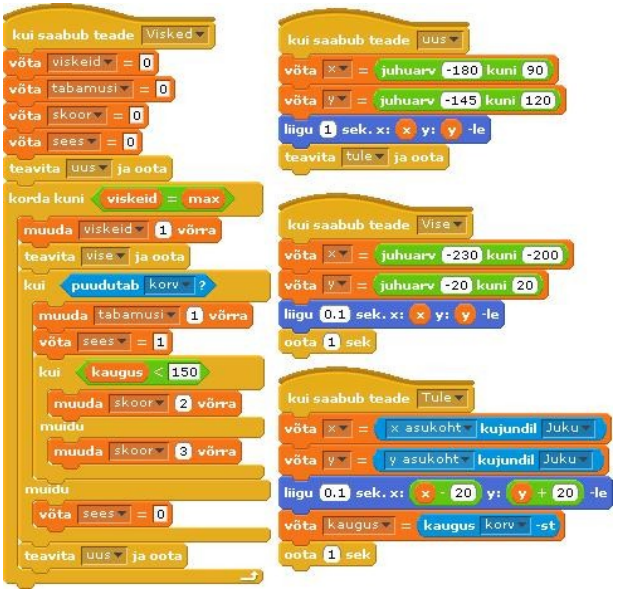
### Fundamentaalsed ideed arvutitest ja programmeerimisest




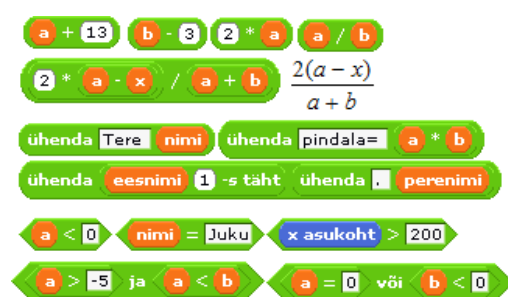
- programm ütleb arvutile täpselt, mida teha, samm-sammult
- programmide koostamine ei nõua spetsiaalseid teadmisi, vaid selget ja hoolikat mõtlemist

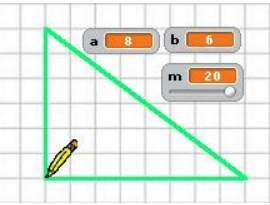
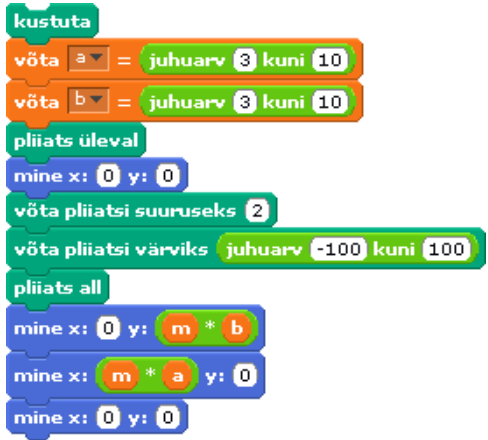
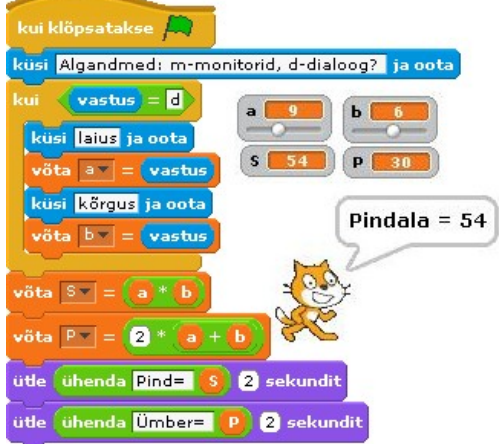

### Rakenduste loomise ja programmeerimise põhikontseptsioonid ja põhimõisted



Kontsept	Selgitus	Näide
<b>Kasutajaliides</b>	<p>Sõltumata programmeerimissüsteemist ja -keelest, kuulub rakenduse loomise protsessi ka kasutajaliidese disain ja realiseerimine.</p> <p>Kasutajaliides sisaldab vahendeid, mille abil saab kasutaja suhelda programmiga: anda vajalike korraldusi ja näha nende täitmise tulemusi, muuta algandmeid jm.</p> <p>Scratch'is luuakse liides <b>laval</b>. Selle elementideks võivad olla taustad, aktiivsed ja passiivsed spraidid, käsunupud, klahvid, muutujate monitorid jm.</p> <p>Teistes süsteemides kasutatakse selleks tavalisel vorme ja dialoogibokse.</p>	<p><a href="#">Näita</a></p>
<b>Programm. Lauseid ja plokkid. Programmi-üksused: protseduurid ja skriptid</b>	<p>Programm on <b>laused</b> (korralduste, käskude) kogum, mis määrab milliseid <b>tegevusi</b> peab arvuti täitma <b>andmete</b> ja/või <b>objektidega</b>, ning tagab ka <b>kasutajaliidese</b> töö.</p> <p>Igas keeles on piiratud valik lauseid, mille esitamiseks on kindlad reeglid.</p> <p>Scratch'is kujutavad <b>käsud</b> (lauseid) endast graafilisi <b>plukke</b>, mis on jagatud otstarve alusel gruppidesse: <b>Liikumine</b>, <b>Juhtimine</b> jne. Plokkiga on lause süntaks määratletud üheselt ja süntaksivigade tekkimine on peaaegu võimatu.</p>	<p><a href="#">Näita</a></p>


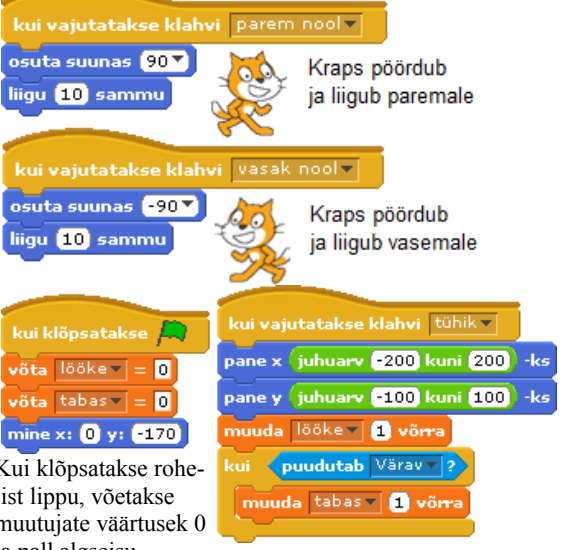

	<p>Programm võib koosneda mitmest üksusest. eid erinevalt: <b>protseduurid</b>, funktsioonid jm.</p> <p>Scratch'is nimetakse programmiüksusi <b>skriptideks</b>. Iga skript on seotud ühe kindla spraidiga ja määrab selle tegevusi. Ühel spraidil võib olla mitu skripti. Skript saab käivitada (öeldakse ka pöörduda, kutsuda välja) teisi skripte, mis kuuluvad samale või teistele spraitidele. Pöördumiseks kasutakse plokkide <b>teavita nimi</b> või <b>teavita nimi ja oota</b>.</p>	<p>Programm koosneb neljast skriptist. Kaks on spraidi Krapsu omad, Jukul ja plikal on üks skript. Kui klõpsatakse rohelist lippu, algab Krapsu esimese skripti täitmine. Peale "teretamist" käivitab antud skript käsuga <b>teavita start ja oota</b> korruga Juku ja plika skriptid, mis algavad plokkidega <b>kui saabub teade start</b>, ning ootab kuni nende täitmine jõuab lõpuni. Sejärel jätkub Krapsu skripti täitmine. Kraps liigub lava keskpunkti ja käivitatakse sama spraidi teine skript. Peale selle täitmist, jätkatakse esimese skripti täitmist, mis viib Krapsu etteantud punkti.</p>
<p><b>Objektid (spraidid). Objektide Omadused, meetodid ja sündmused</b></p>	<p>Scratch'is on kesksel kohal <b>graafilised objektid</b>, mida nimetakse <b>spraitideks</b> (sprite) ning nende kostüümid. Objektideks on sanuti ka <b>lava</b> ja selle taustad. Kuigi Scratch ei ole formaalselt objekt-orienteeritud süsteem, on sageli otstarbekas seda käsitleda, tuginedes objekt-orienteeritud lähenemisviisile.</p> <p>Iga objektiga on seotud teatud valik <b>omadusi</b>: nimi, asukoht laval (x-y), suurus, värvus jm, ning <b>meetodeid</b>, mille abil määratakse tegevusi antud tüüpi objektiga: asukohta muutmine, pööramine, suuruse ja värvuse muutmine jm. Käsuplokkid sisuliselt vastavad meetoditele.</p> <p>Objekt võib reageerida <b>sündmustele</b>: hiire klõps, vajutus etteantud klahvile, kokkupuude teise objektiga.</p>	<p>Skripti täitmine algab, kui klõpsatakse spraiti Kraps (reaktsioon sündmusele). <a href="#">Näita</a></p>  <p>Skripti plokkid muudavad objekti (spraidi) mitnesugusi omadusi: suund, asukoht, suurus, värvus, keere.</p>
<p><b>Andmete liigid</b></p>	<p>Scratch'is saab kasutada märk-, graafika- ja heliandmeid.</p> <p><b>Märkandmed</b>: arvud ja tekstid ehk stringid. Neid saab kasutada paljudes plokkides (käskudes) konstantidena, muutujatena ja loendite elementidena. Väärtusi saab leida (tuletada) avaldiste ja funktsioonide abil.</p> <p><b>Graafikaandmed</b> võivad esineda rakendustes kahes variandis:</p> <p><b>Spraidid</b> ja lava <b>taustad</b> – imporditakse või tehakse graafikaredaktori abil. Käskude abil saab määrata erinevaid tegevusi nendega.</p>	 <p>Skriptid demonstreerivad erinevat liiki andmete kasutamist. Märkandmed (arvud ja stringid) on kasutusel konstantidena ja muutujatena. <a href="#">Näita</a></p>

	<p><b>Pliiatsi käskudega tehtavad joonised</b></p> <p><b>Heliandmed.</b> Saab kasutada erinevaid vahendeid helide tekitamiseks (plokid <b>mängi nooti...</b>, <b>mängi trummi...</b> jm), importida ja lindistada heliklippe: kõne, muusika jm,</p>	 <p>Graafikaandmeid esindab sprait Kraps ja pliiatsiga tehtav joon spraidi liikumisel.</p> <p>Heliandmetga on tegemist plokkide <b>mängi heli ...</b> ja <b>mängi trummi...</b> kasutamisel.</p>
<p><b>Andmete organisatsioon</b></p>	<p>Organisatsiooni järgi eristatakse kõikides programmeerimiskeeltes:</p> <p><b>skalaarandmeid:</b> konstandid ja muutujad</p> <p><b>struktuurandmeid:</b> tabelid, massiivid, loendid jm</p> <p><b>Konstantide</b> väärtused esitatakse vahetult programmis. Ülaltoodud skripti näites on mitu arv- ja stringkonstanti. Need on plokkide väljades asuvad konkreetset arvud: 2, 100, 58, 0.2, 0 jne; ja tekstid: "Tere! Mina olen Kraps!", "Mis on Sinu nimi?" jne.</p> <p><b>Muutujad</b> esitatakse programmis nimede abil, nende väärtusi programmis ei näe. Väärtused tavaliselt tekkivad programmi täitmise ajal. Näiteks muutuja <i>pikkus</i> väärtus tekkitab sisestamisel käsuga <b>küsi</b>, muutuja <i>normkaal</i> väärtus tekkitab arvutuste tulemusena (<i>pikkus</i> – 100). Muutujate olemusest ja käsitlemisest vt allpool.</p> <p>Tabelid, massiivid, <b>loendid</b> kujutavad endast kindla struktuuriga väärtuste kogumeid. Scratch'is saab kasutada <b>ainult loendeid</b>, mis on on massiivide erijuht (vt. allpool)</p>	
<p><b>Muutujad ja omistamine</b></p>	<p>Muutuja on <b>mälupesa</b>, kuhu programm saab salvestada väärtusi: arve ja stringe ning kasutada (lugeda) neid hiljem näiteks uute väärtuste leidmiseks.</p> <p>Scratch'is saab muutujaid luua ja kasutada grupis <b>Muutujad</b> olevate plokkide abil. Loomisel saab anda muutujale <b>nime</b>, mida kasutatakse käskudes viitamiseks tema jooksvale väärtusele. Loomisel saab ka määrata, kas muutuja on kättesaadav kõikidele spraitidele (globaalne muutuja) või ainult ühele konkreetsele spraidile (lokaalne muutuja).</p> <p>Muutujatele väärtuste omistamiseks ja väärtuste muutmiseks kasutatakse plokkide</p> <p><b>võta</b> muutuja = avaldis ja</p> <p><b>muuda</b> muutuja avaldis võrra</p> <p>Muutuja väärtust saab kuvada laval nn <b>monitori</b> abil . Väärtust saab muuta ka "käsitsi" <b>liuguriga</b>.</p>	<p>Programm imiteerib korvpalli viskeid</p>  <p>Näited: <a href="#">Korvpall</a> ja <a href="#">Ideaal</a></p>

<p><b>Loendid (massiivid)</b></p>	<p>Scratch'i loend vastab enam vähem teistes keeltes kasutatavatele dünaamilistele ühemõõtmelistele massiividele. Taoline massiiv kujutab endast järjestatud mälupeade (elementide) kogumit. Elementidele saab viidata massiivi nime ja indeksite abil.</p> <p>Scratch'is saab luua ja kasutada loendeid grupi <b>Muutujad</b> plokkidega. Viimaste abil saab lisada elemente loendi lõppu ja vahele, asendada ja emaldada elemente, viidata elementidele jm.</p> 	 <p>kesk 9050 mini 7400 arvuti Marvin</p> <table border="1"> <thead> <tr> <th>Arvutid</th> <th>Hinnad</th> </tr> </thead> <tbody> <tr><td>1 Aragom</td><td>1 8400</td></tr> <tr><td>2 Frodo</td><td>2 7900</td></tr> <tr><td>3 Balrog</td><td>3 12300</td></tr> <tr><td>4 Eldar</td><td>4 9500</td></tr> <tr><td>5 Marvin</td><td>5 7400</td></tr> <tr><td>6 Sauron</td><td>6 8800</td></tr> </tbody> </table> <p>+ pikkus: 6 + pikkus: 6</p> <p>Skript keskmise hinna leidmiseks</p>  <p><a href="#">Näita</a></p>	Arvutid	Hinnad	1 Aragom	1 8400	2 Frodo	2 7900	3 Balrog	3 12300	4 Eldar	4 9500	5 Marvin	5 7400	6 Sauron	6 8800
Arvutid	Hinnad															
1 Aragom	1 8400															
2 Frodo	2 7900															
3 Balrog	3 12300															
4 Eldar	4 9500															
5 Marvin	5 7400															
6 Sauron	6 8800															
<p><b>Avaldised, tehted ja funktsioonid</b></p>	<p>Avaldise abil saab anda eeskirja vajaliku väärtuse leidmiseks. Avaldiste loomiseks kasutatakse Scratch'is grupi <b>Tehted</b> plokkide.</p> <p>Avaldis koosneb operandidest ja tehetest.</p> <p>Operandideks võivad olla: konstandid, muutujad, funktsioonid, loendite elemendid.</p> <p>Sõltuvalt tehetest võib avaldised jagada järgmistesse rühmadesse:</p> <ul style="list-style-type: none"> <li>• arvavaldised: +, -, *, /</li> <li>• stringavaldised: ühenda, eralda</li> <li>• võrdlused: &lt;, =, &gt;</li> <li>• loogikaavaldised: ja, või, mitte</li> </ul> <p>Avaldites saab kasutada mitmeid funktsioone: <b>abs</b> (absoluutväärtus), <b>sqrt</b> (ruutjuur), <b>sin</b>, <b>cos</b>, <b>asin</b>, <b>log</b>, <b>ln</b> jm.</p> <p>Sisuliselt on funktsioonideks ka eraldi esitaud plokkid; <b>mod</b> (jääk), <b>ümarda</b> ja <b>juhuarv</b>.</p> <p><b>NB!</b> Sulge tehete järjekorra määramiseks Scratch'is kasutada ei saa.</p>	 <p><math>a + 13</math>   <math>b - 3</math>   <math>2 * a</math>   <math>a / b</math></p> <p><math>2 * a - x / a + b</math>   <math>\frac{2(a-x)}{a+b}</math></p> <p>ühenda Tere nimi   ühenda pindala = <math>a * b</math></p> <p>ühenda esinimi 1 -s täht   ühenda .   perenimi</p> <p><math>a &lt; 0</math>   nimi = Juku   <math>x</math> asukoht <math>&gt; 200</math></p> <p><math>a &gt; -5</math> ja <math>a &lt; b</math>   <math>a = 0</math> või <math>b &lt; 0</math></p> <p>võta c = <math>a + 13 / \log   \text{abs} b</math></p> <p>võta c = ümarda <math>c * 100 / 100</math></p> <p><math>c = \frac{a+13}{\log  b }</math>   c ümardakse kahe kohani peale koma</p> <p>lõputult</p> <p>pane x juhuarv -240 kuni 240 -ks</p> <p>pane y juhuarv -180 kuni 180 -ks</p> <p>oota 1 sek</p> <p>palli asukohta laval muudetakse juhuarvude abil</p> <p>Näide <a href="#">Ideaal</a></p>														

<p><b>Jonistamine, joonistamine</b></p>	<p>Enamikus programmeerimiskeeltes on olemas vahendid, mis võimaldavad luua programmil täitmise ajal jooniseid.</p> <p>Scratch'is on peamised vahendid selleks grupides <b>Pliiats</b> ja <b>Liikumine</b>. Kõrvalolev skript joonistab täisnurkse kolmnurga, mille külgede pikkused <b>a</b> ja <b>b</b> tekitatakse juhuarvude abil.</p> <p>Täisnurk pannakse lava keskele: punkti (0,0). Muutuja <b>m</b> abil määratakse mastaap: punktide arv ühes pikkuse ühikus.</p> 	 <p>Demo <a href="#">Rist_Ring</a></p>
<p><b>Märkandmete sisestamine ja väljastamine</b></p>	<p>Ülesannete lahendamisel on sageli programmil vaja saada algandmeid ja peegeldada tulemusi. Esimesel juhul räägitakse andmete <b>sisestamisest</b> (lugemisest), teisel juhul – andmete <b>väljastamisest</b> (kirjutamine, kuvamine).</p> <p>Scratch'is saab andmete sisestamiseks kasutada muutujate liuguritega monitore (ainult arvud) ja plokki <b>küsi</b>, mis võimaldab sisestada arve ja tekste dialoogirežiimis.</p> <p>Tulemusi saab väljastada muutujate ja/või loendite <b>monitoridega</b> või plokkidega <b>ütle</b>.</p>	 <p>Näide <a href="#">Ideaal</a></p>
<p><b>Protsesside juhtimine</b></p>	<p>Väga tähtsal kohal on ülesannete lahendamisel tegevuste täitmise järjekorra määramine. Just selle seisneb programmeerimise kunst. Vajalikud tegevused ja nende järjekord ei sõltu üldiselt kasutatavast keelest, vaid tulenevad ülesande olemusest ehk selle lahendamise <b>algoritmist</b>.</p> <p>Võib eristada nelja liiki protsesse: <b>jada</b> ehk järjestikune protsess, <b>kordus</b> ehk tsükliline protsess, <b>valik</b> ehk hargnev protsess ning <b>paralleelne protsess</b>.</p> <p>Programmeerimiskeeltes on olemas vahendid erinevat liiki protsesside kirjeldamiseks.</p>	
<p><b>Jada</b> ehk järjestikune protsess</p>	<p>Programmi koostamisel peab arvestama, et plokkide poolt määratud tegevusi täidetakse kindlas järjekorras.</p> <p>Lihtsamal juhul on tegemist järjestikuse protsessiga, kus plokkide täidetakse järjest ülevalt alla.</p>	

<p><b>Kordus</b> ehk tsükliline protsess</p>	<p>Korduste kirjeldamiseks on Scratch'is, nagu ka teistes keeltes, mitu lauset (plokki): <b>lõputult</b>, <b>korda n</b>, <b>korda kuni tingimus</b>, <b>lõputult kui tingimus</b>. Ploki <b>lõputult</b> sees olevaid plokkide täidetakse põhimõtteliselt lõpmatult. Skripti töö saab katkestada näiteks punase nupuga. Korduse sees võib olla üks või mitu plokki <b>kui tingimus</b>, mis tingimuse täitumisel katkestab vastavate plokkide abil skripti (plokk <b>peata skript</b>) või terve programmi töö (<b>peata kõik</b>).</p> <p>Käsu <b>korda n</b> täitmisel korratakse ploki sees olevaid plokkide <b>n</b> korda. Kordamiste arv (<b>n</b>) võib olla antud konstandi, muutuja või avaldise abil.</p> <p>Käsu <b>korda kuni tingimus</b> täitmisel korratakse ploki sees olevaid plokkide seni, kuni tingimus saab tõseks.</p> <p>Käsu <b>korda kui tingimus</b> täitmisel korratakse ploki sees olevaid plokkide seni, kui tingimus on tõene</p>	 <p>Esimese ploki toimetel "jalutab" sprait lõpmatult laval edasi-tagasi. Katkestada saab punase nupu abil. Teise ploki täitmisel liigub sprait vasakult paremale. Liikumine (kordamine) lõpeb, kui spraidi x-koordinaat saab suuremaks 200st,</p> <p>Objekt (näiteks kiisu) "põristab" trummi ja teeb ringi (360°) laval</p> <p><a href="#">Näita</a></p>
<p><b>Valik</b> ehk hargnev protsess</p>	<p>Plokkidega <b>kui tingimus</b> ja <b>kui tingimus muidu</b> saab kirjeldada valikuid ehk hargnevaid protsesse.</p> <p>Lause <b>kui ...</b> korral (valik ühest), kui <b>tingimus</b> on tõene täidetakse ploki sees olevad plokkid, vastupidisel juhul jäetakse need vahele.</p> <p>Lause <b>kui ... muidu</b> korral (kahendvalik), kui <b>tingimus</b> on tõene täidetakse esimeses harus olevad plokkid, vastupidisel juhul teises harus olevad plokkid.</p>	 <p>Esimeses ploki kontrollitakse tingimust, kas x-koordinaat on suurem 200-st, kui jah, täidetakse sisemised plokkid. Teises skripti fragmendis kuvatakse teade, sõltuvalt sisestatud vanusest. Näide <a href="#">Ideaal</a></p>

<p><b>Harud</b> ehk paralleelsed protsessid</p>	<p>Kahte või enamat skripti (protsessi) saab täita samaaegselt ehk paralleelselt. Paralleelset täitmist saab määrata mitmel erineval viisil.</p> <p>Näiteks kõik skriptid, mille esimeseks plokiks on roheline lipuga plokk, käivitakse samaaegselt ja täidetakse paralleelselt, kui klõpsatakse rohelist lippu.</p> <p>Paralleelselt täidetakse ka skripte, mis algavad plokiga <b>kui saabub teade nimi</b>, milles on kasutusel sama <b>nimi</b>.</p>	 <p>Üks objekt liigub pidevalt mööda kolmnukset trajektori, teine – mööda rigikujulist.</p> <p>Kui saabub teade ”korras”, üks objekt teeb 3 hüpet, teine vahetab 8 korda kostüümi (näiteks tantsib)</p> <p>Projekt <a href="#">Tantsud</a></p>
<p><b>Sündmused</b></p>	<p>Objektid võivad reageerida teatud sündmustele: vajutus mingile klahvile, objekti klõpsamine hiirega, kokkupuude teise objektiga või lava servaga jm. Skriptides võib ettenäha reaktsioone kindlatele sündmustele. Peamisteks vahenditeks sündmuste haldamisele on nn päiseplokkid:</p> <p><b>kui vajutakse klahvi</b> ja <b>kui klõpsatakse spraiti</b></p> <p>Skriptide sees leiab sageli kasutust plokk <b>kui puudutab</b> {sprait   serv   kursor}</p> <p>Kui antud sprait puudutab teist spraiti, lava serva või hiire kursorit, tagastab plokk väärtuse <b>tõene</b>.</p> <p>Põhimõtteliselt on sündmusega tegemist ka roheline lipu kasutamisel ja skriptide käivitamisel plokkide <b>teavita</b> ja <b>kui saabub teade</b> abil (vt allpool).</p>	 <p>Kui klõpsatakse rohelist lippu, võetakse muutujate väärtuseks 0 ja pall algseisu. Kui vaiutakse tühiku klahvi, muudetakse juhuarvudega palli asukohta ja suurendatakse löökide arvu ühe võrra. Kui pall puudutab väravat, lisatakse 1 muutujale tabas. Proovige!</p>
<p><b>Dünaamiline intreaktsioon</b></p>	<p>Objekte saab mõjutada nõ reaalajas hiire kursoriga (hiire liigutamisega), heliga jm., kasutades grupi <b>Andurid</b> plokkide <b>hiire x</b>, <b>hiire y</b>, <b>helitugevus</b> jm</p>	 <p>Kui klõpsatakse lippu, liigub objekt horisontaalsuunas koos hiire kursoriga ja muudab kerme suurust, vastavalt y-i väärtusele. Proovige!</p>

<p><b>Koostöö spraitide ja skriptide vahel.</b></p>	<p>Kui programm koosneb mitmest skriptist tekib sageli vajadus nende töö koordineerimiseks ja sünkroniseerimiseks. Üks sprait võib pöörduda teiste poole, need omakorda järgimiste poole jne. Skriptide koostöö korraldamisel kasutatakse plokkide <b>teavita teade</b> ja <b>teavita teade ja oota</b> ning <b>kui saabub teade</b></p> <p>Siin vaadeldav programm koosneb neljast skriptist. Rohelise lipuga skrip on <b>peaskript</b>. Sellest algab programmi töö ning ta käivitab teisi. Kui kasutaja sisestab tähe "d", käivitatakse skript <b>Loe</b> ning peale selle töö lõppu pannakse tööle skript <b>Arvuta</b>. Kui sisestati d-st erinev väärtus, käivitatakse kohe <b>Arvuta</b>.</p>	<p>Ootamata <b>Arvuta</b> töö lõppu pannakse tööle ka skript <b>Joonesta</b> (ei ole siin näidatud, on analoogiline skriptiga punktis <b>Joonestamine</b>). <a href="#">Näide</a>.</p>
<p><b>Algoritmimine</b></p>	<p>Algoritmi all mõistetakse eeskirja, mis määrab milliseid tegevusi ja mis järjekorras peab täitma antud ülesande lahendamiseks või töö täitmiseks. Vajalikud tegevused ja nende täitmise järjekord üldiselt ei sõltu konkreetsest keelest vaid ülesandest või mudelist. Tegevused algoritmis esitatakse üldisemal kujul. Viimasel ajal kasutatakse algoritmide esitamiseks sageli modelleerimiskeelt UML.</p> <p>Scratch'i skripte võib vaadelda kui ühte algoritmide esitusviisi</p>	<p>Demod: <a href="#">Jalka</a>, <a href="#">Ruutvõrand</a>, <a href="#">Maks_Vek</a></p>
<p><b>Modelleerimine</b></p>	<p>Scratch'i vahendeid saab kasutada objekt-orienteeritud modelleerimise põhimõtete selgistamiseks ja illustreerimiseks. Scratch'i spraidid kuuluvad ühte universaalse <b>klassi - Sprait</b>. Spraitidel kindel valik <b>omadusi</b>: nimi, asukoht, suurus, ... ja <b>meetodeid</b>: liigu(), pööra(), muudaX(), ... Viimased esitatakse käskude ehk plokkide abil</p> <p>Käskudest (baasmeetoditest) moodustatud skripte võib käsitleda kasutaja <b>alamklassidena</b>. Alamklassidena on käsitletavat ka spraidiga seotud kostüümid, pliats, muutujad, loendid.</p> <p><b>Lava</b> on vaadeldav nõ peaklassina: klass <b>Sprait</b> on selle alamklass: kõik spraidid asuvad laval.</p> <p>Klassi <b>Lava</b> võib vaadelda kui süsteemi või projekti nõ esindajana.</p>	<p>Näide. Projekt <a href="#">Tantsud</a>.</p>



**Programmeerimise kontseptid, mida Scratch praegu ei toeta:**

funktsioonid; parameetrite ja argumentide kasutamine; rekursioon; oma objektiklasside defineerimine; pärimine; vigade ja erandite haldamine; failide sisend/väljund.