

УНИВЕРЗИТЕТ ПРИВРЕДНА АКАДЕМИЈА У НОВОМ САДУ  
Факултет за примењени менаџмент, економију и финансије, БЕОГРАД



ФАКУЛТЕТ ЗА 21. ВЕК

Мастер рад:

**Основни програмски елементи Blockly пакета**

Студент:

Емина Мустафић  
I01-12/2018

Ментор:

проф. Др Миодраг Брзаковић

Београд, 2021.

## Садржај

Увод.....	4
Хипотеза.....	4
1. Увод у програмирање у визуелном графичком окружењу .....	5
1.1 Креирање апликације.....	7
1.2 Креирање апликације -Coding with Chrome (Blockly Games) .....	11
2. Програмирање вођено догађајима и руковање догађајима .....	14
2.1 Упознавање основних стандардних компоненти и догађаја.....	16
Неки примери у Blockly .....	20
3. Основни елементи програмских језика JavaScript и PHP .....	23
Основни типови података .....	23
Текстуални тип.....	25
Логичке вредности.....	25
Вредности undefined .....	26
3.1 Основни типови података у Php- у су: .....	26
3.2 Преглед оператора .....	30
4. Конверзија типова података .....	34
4.1 Експлицитна конверзија типова података (eng. Type Casting) .....	34
4.2 Аутоматска конверзија типова података (eng. Type Juggling).....	35
4.3 Уношење и приказ података .....	36
4.4 Алгоритми линијске структуре.....	36
5. Цртање разних облика.....	38
5.1 Цртање углова .....	41
5.2 Померање лоптице (Blockly).....	43
6. Blockly Games .....	44
6.1 Puzzle - Слагалице.....	45
6.2 Maze – Лавиринт .....	45
6.3 Bird – Птица.....	46
6.4 Turtle- Корњача .....	47
6.5 Movie – Премести .....	47
6.6 Pond Tutor – Рибњак тутор .....	48

Pond – Рибњак .....	49
Задатак 1. ....	49
Задатак 2. ....	51
ЗАКЉУЧАК .....	54
ЛИТЕРАТУРА .....	55

## Увод

У последњих четрдесетак година сведоци смо великих друштвених промена и интензивног развоја информационих технологија - информационо доба. Напредак у информационим технологијама и комуникацијама мења и наш начин живота: утиче на то како радимо и послујемо, како се образујемо, како самостално учимо, како проводимо слободно време. Информације се крећу брже него што се одвија физичко кретање. Данас у развијеним земљама живи једна нова генерација - "Миленијумска генерација" (рођени након 1981.год.), коју карактерише познавање и повећана употреба комуникација, медија и дигиталних технологија у свакодневним активностима. То је генерација која је одрасла уз рачунаре и интернет. Поставља се питање: Да ли данашња генерација учи другачије од претходних генерација? Одговор на постављено питање је потврдан и води нас до неких специфичности које одликују начин учења савремених генерација.

### Алгоритамски начин размишљања

Живимо окружени рачунарима. Сакривени су у нашим мобилним уређајима, сатовима, телевизорима, кухињским апаратима, аутомобилима, лифтовима. Олакшавају нам послове, а учење чине занимљивијим. Захваљујући рачунарима, слободно размењујемо информације и имамо брз приступ знању.

Често се чини да су рачунари паметни. Али рачунари, још увек, не могу самостално да мисле! Рачунари су машине које следе упутства (наредбе) и тако обављају различите задатке. Наредбе рачунарима пишу људи, мали и велики програмери. Скупови наредби које рачунар извршава називају се програм.

Да би рачунар могао да ти помогне да се брже и боље обавља неки посао, мора да буде за то „обучен“ - програмиран.

Рачунар, сам по себи, не рађа креативност, али помаже да се она чује далеко и одјекне громогласно.

## DA ODVOJIM U NABRAJANJE

### Хипотеза

Настава предмета Рачунарство и информатика у основним и средњим школама обухвата изучавање барем једног програмског језика у циљу савладавања алгоритамског начина размишљања. Међутим, апстрактност линија кода и примери који су честу уско везани за наставу математике код ученика изазивају ефекат одбојности и дистанцирања од неразумљивог градива.

У овом раду показати како се коришћењем графичког корисничког окружења и Блокку библиотеке алгоритамско размишљање и решавање проблема може савладати на много интересантнији и ученицима пријемчивији начин.

blockly pruza uvod u javascript i php na nacin koji ce deci priliziti programiranje na laksi nacin.

blockly omogućuje na principu slagalice da savladaju osnove programiranja.

kroz blockly program deca upotpunjuju poznavanje matematike kroz programiranje i iste savladavaju

blockly program pomaze deci da nauce nove tipove podataka koji ce im kasnije koristiti

---

## Преглед Blockly пакета

### 1. Увод у програмирање у визуелном графичком окружењу

**Графичко корисничко окружење (GUI)** је тип корисничког окружења који омогућава људима да рукују рачунаром или уређајима контролисаним рачунаром. То је начин интеракције човека са рачунаром кроз манипулацију графичким елементима И додацима уз помоћ текстуалних порука и обавештења. GUI користи иконице, визуелне индикаторе или специјалне графичке елементе уместо обичних текстуалних менија или ручног уношења команди. Основни елементи овог модела су: **Window** (прозор)- ограничене правоугаоне области које садрже радну површину, меније, дугмиће, траке за вертикално померање скривања садржаја прозора и друго.

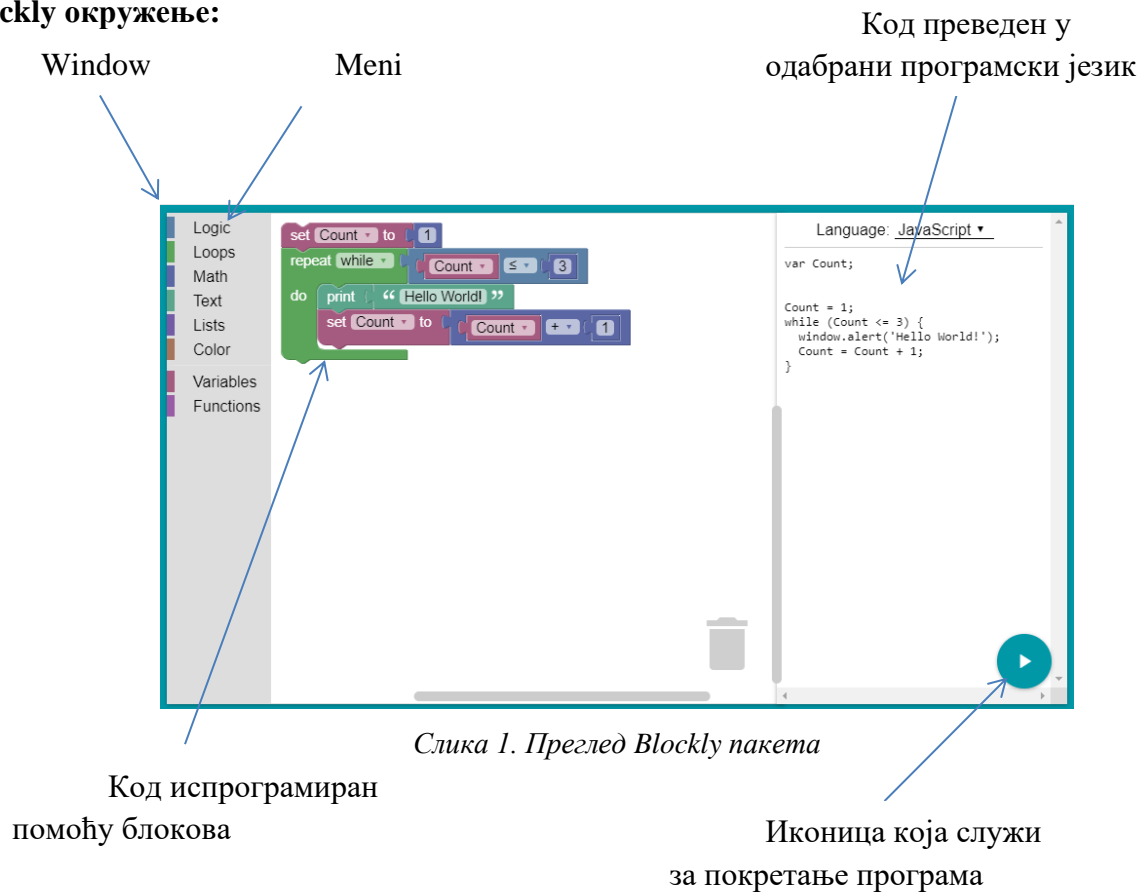
- **Icon** (икона или сличица)-визуелни репрезент објеката у саставу рачунара
- **Menu** (мени) основни механизам интеракције корисника са програмом
- **Pointer** (показивач)-графички симбол којим управља миш, или неки други улазни уређај, а користи се за лоцирање, позиционирање и избор визуелног репрезента
- објекта на екрану монитора

**Blockly** представља библиотеку за изградњу апликација за програмирање блокова. Блок програмирање омогућава корисницима да креирају скрипте и програме користећи визуелне блокове, чак и ако не познају ниједан програмски језик. Blockly укључује све што вам је потребно за дефинисање и приказивање блокова у drag-and-drop едитору. Сваки блок представља део кода, који се може слагати и превести у код. Blockly омогућава

корисницима да прилагођавају компоненте и да одреде понашање одређених делова апликација.

Официјални сајт је: <https://developers.google.com/blockly>

### Blockly окружење:



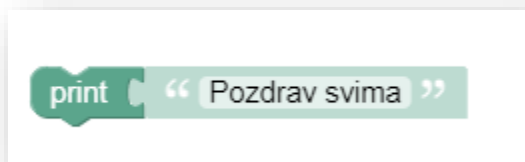
Слика 1. Преглед Blockly пакета

**Blockly** долази са великим бројем унапред дефинисаних блокова. Ипак да би искористили Blockly за спољну апликацију, некада је потребно да креирамо нове блокове помоћу Blockly developers tools. На пример да би креирали програм за цртање, претходно морамо креирати блок “цртање круга полупречника r”.

Оно што испрограмирамо помоћу блокова, можемо превести у неки од понуђених програмских језика као што су JavaScript, Python, PHP, Lua и Dart.

**Пример . Направите свој први програм који на излазу исписује “Поздрав свима” помоћу блокова у Blockly пакету.**

**Решење:**



## 1.1 Креирање апликације

Да би били у могућности да покренемо код који укључује Blockly морамо пратити следеће кораке.

1. Са официјалне странице <https://developers.google.com/blockly/guides/get-started/web> преузмемо zip file
2. Додавање Blockly библиотека

Прво морамо додати скрипте и скуп основних блокова. Имајте на уму да путања може варирати у зависности од тога где се налази ваша страница у односу на Blockly.

```
<script src="blockly_compressed.js"></script>
<script src="blocks_compressed.js"></script>
```

Затим укључимо поруке за кориснички језик.

Додамо празан div негде у body и подесимо његове димензије.

```
<div id="blocklyDiv" style="height: 480px;
width: 600px;"></div>
```

Додамо структуру toolbox-a:

```
<xml id="toolbox" style="display: none">
  <block type="controls_if"></block>
  <block type="controls_repeat_ext"></block>
  <block type="logic_compare"></block>
  <block type="math_number"></block>
  <block type="math_arithmetic"></block>
  <block type="text"></block>
  <block type="text_print"></block>
</xml>
```

На крају, додамо следећу скрипту, обично на крају странице, како би

```
<script src="msg/js/en.js"></script>
```

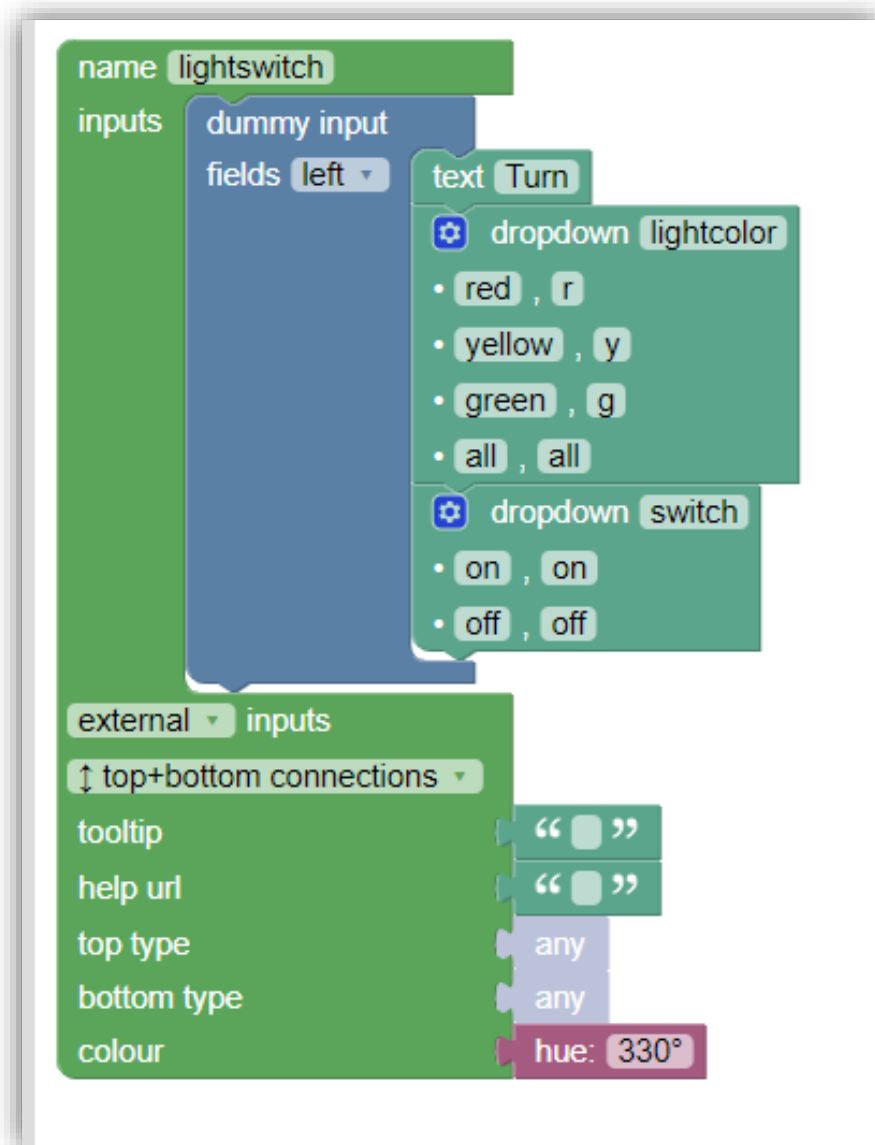
укључили  
Blockly у  
празан div.

```
<script>
var workspace = Blockly.inject('blocklyDiv',
  {toolbox: document.getElementById('toolbox')});
</script>
```



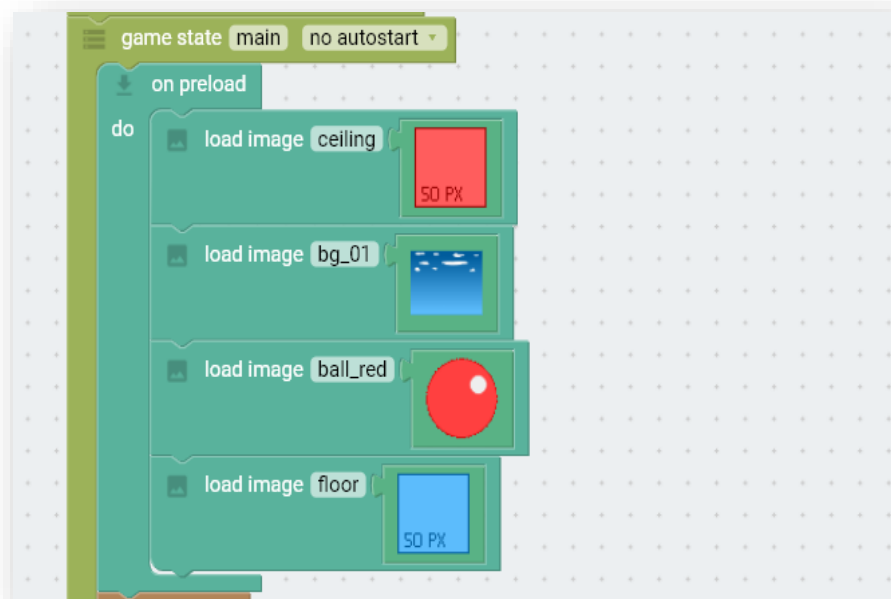
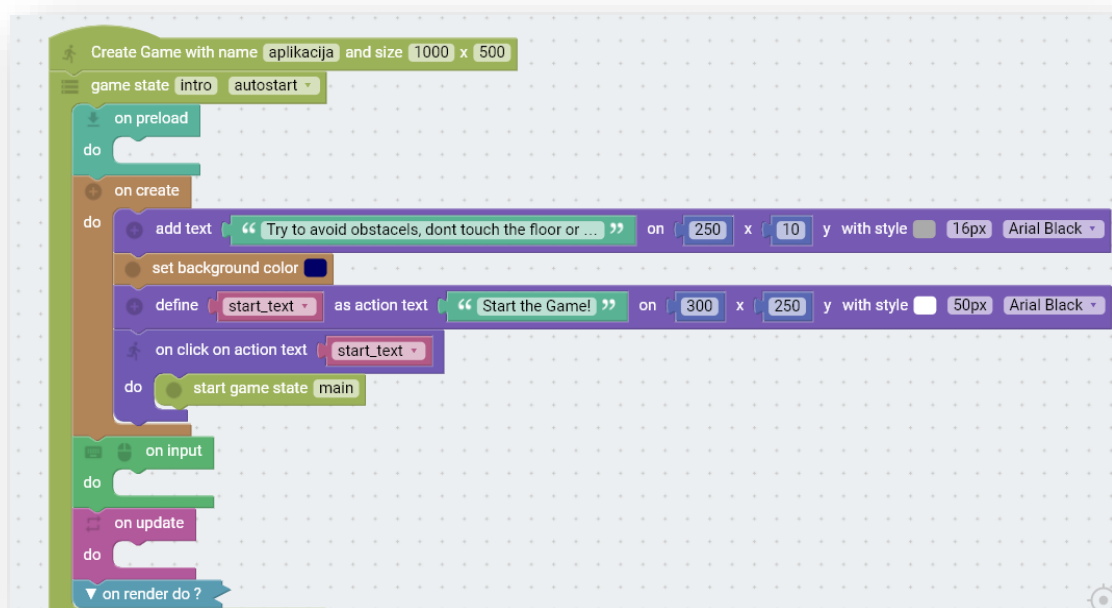
У [Blockly developers tools](#) можемо испрограмирати нове блокове који нам требају за нашу апликацију.

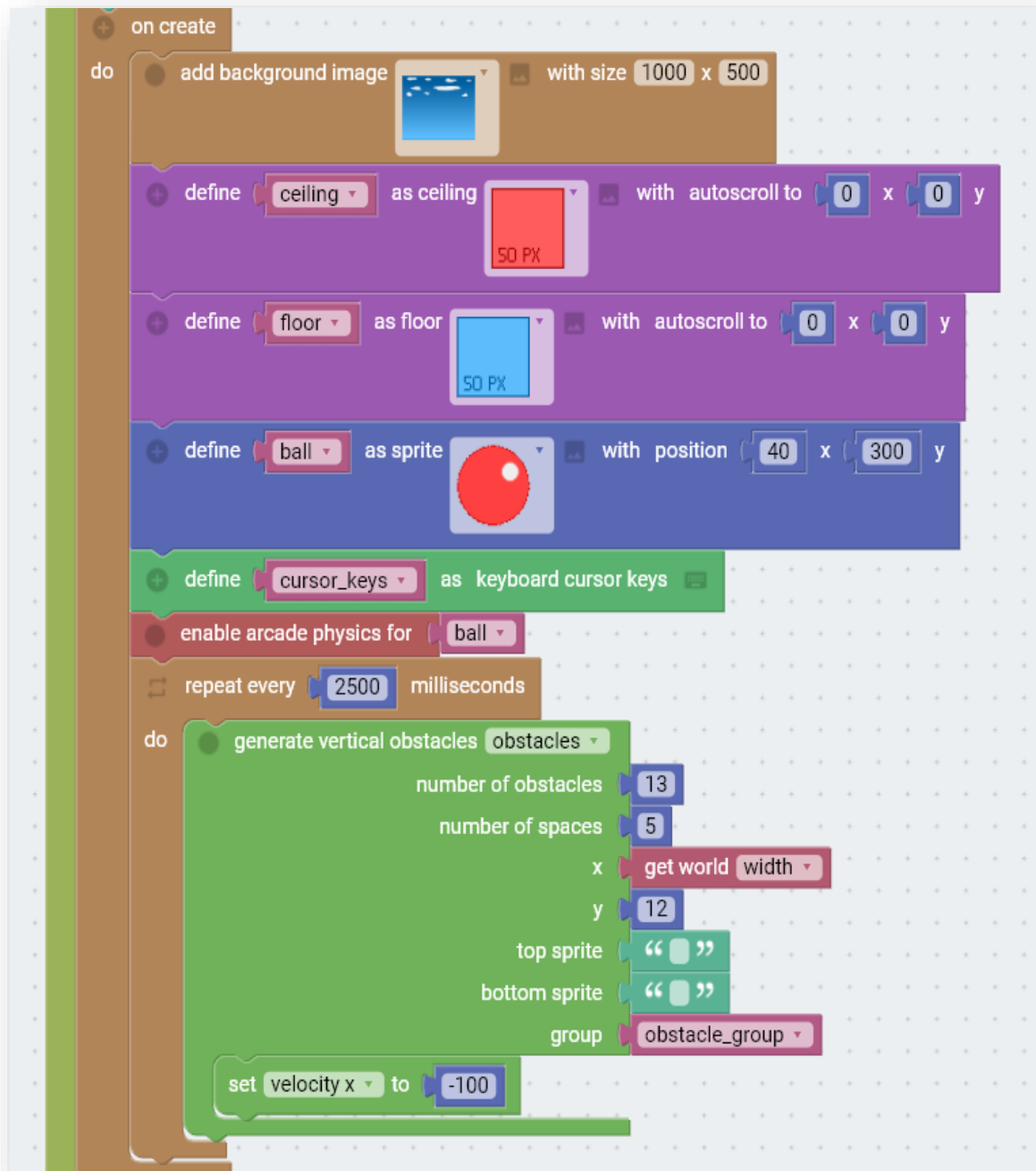
Пример:

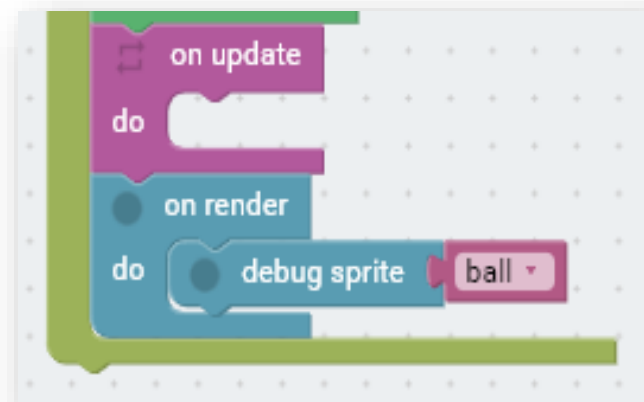
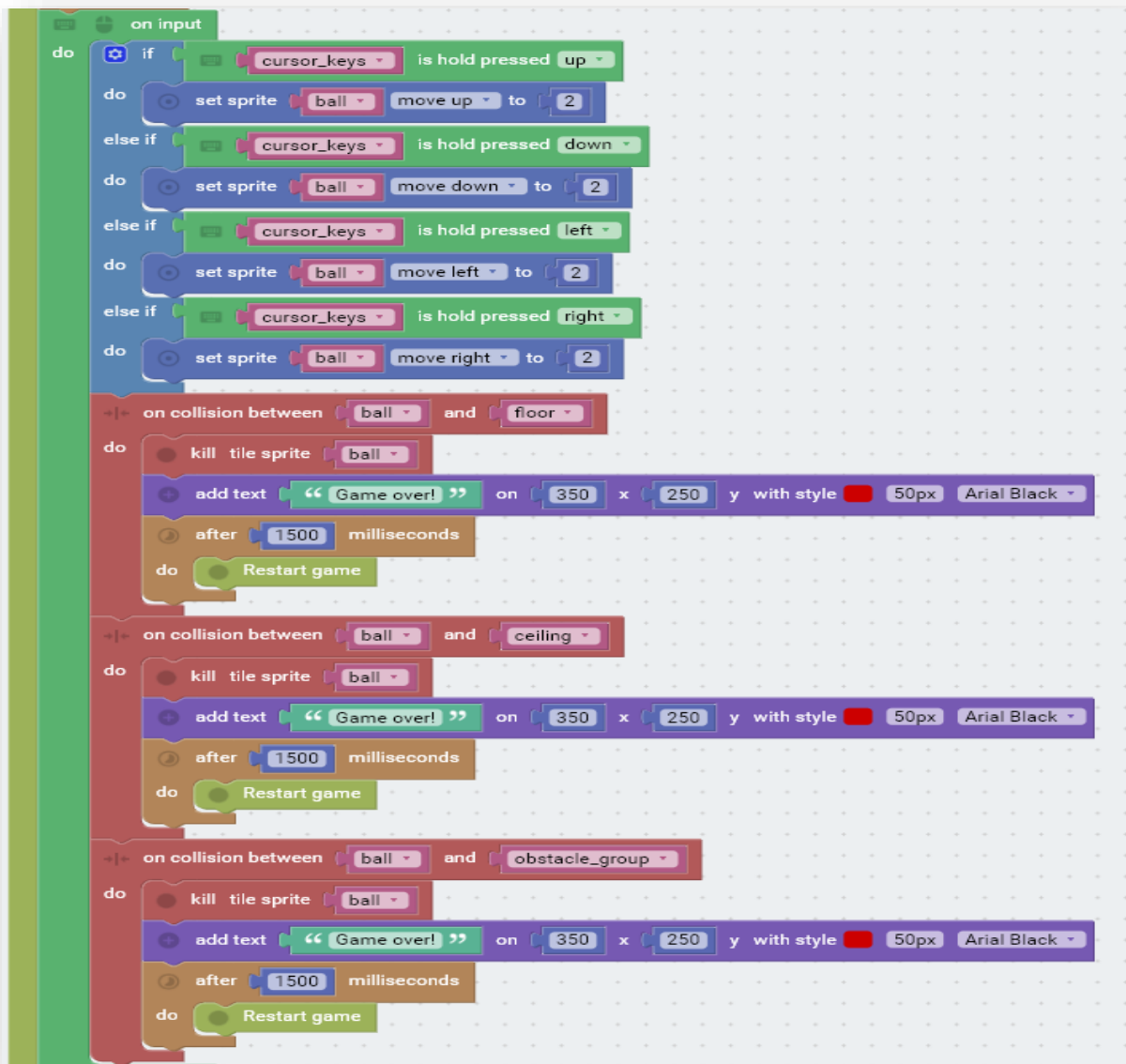


Дугме које смо креирали служи да одаберемо боју, и то ћемо искористити како би направили семафор.

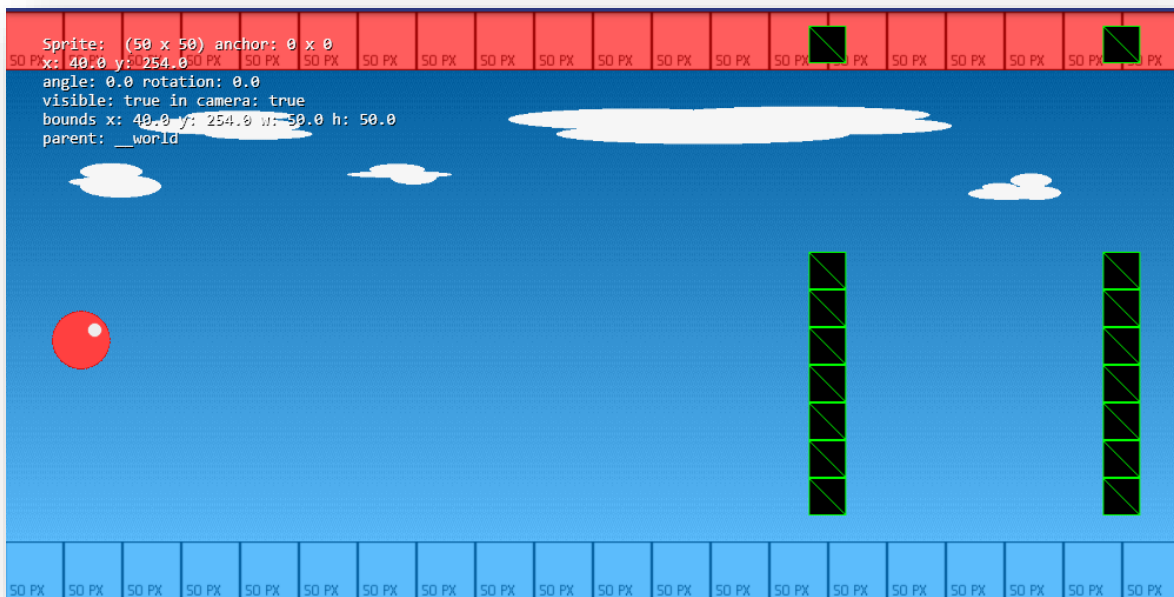
## 1.2 Креирање апликације -Coding with Chrome (Blockly Games)







## Резултат:



## 2. Програмирање вођено догађајима и руковање догађајима

**Програмирање вођено догађајима** (енгл. Event-driven programming) је парадигма у програмирању у којој је ток програма одређен догађајима као што су акције корисника (клик мишем, притискање тастера), сензор излаза, или порукама из других програма / нити. Програмирање вођено догађајима је доминантна парадигма која се користи код графичких корисничких интерфејса и у другим апликацијама (нпр. JavaScript веб апликације) које су усмерене да обављају поједине радње у одговору на кориснички унос.<sup>1</sup>

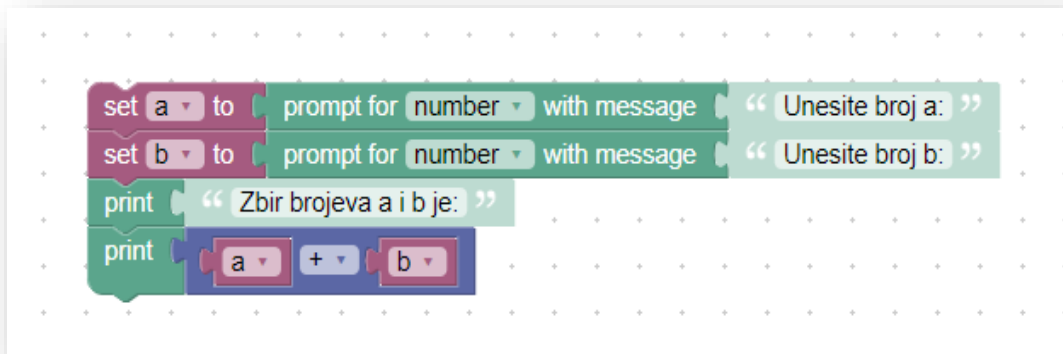
У апликацији вођеној догађајима, генерално постоји главна петља која слуша догађаје, а затим активира повратни позив функције када се детектује догађај. У уграђеним системима исти се може постићи коришћењем хардверских прекида уместо главне петље која се стално извршава. Програми вођени догађајима могу бити написани у било ком програмском језику, мада их је лакше писати у језицима који обезбеђују апстракције на високом нивоу, као што су затворена.

Углавном сви програми написани у Blockly-у су вођени догађајима тј. свака акција коју изврши програм проузрокована је дешавањем некаквог догађаја. А догађаји су у суштини, акције које предузима корисник тј. посетилац ваше веб странице.

<sup>1</sup> <https://sr.glosbe.com/en/sr/event-driven%20programming>, datum pristupa sajtu 15.03.2021. godine.

**Пример. Направити програм који ће наћи збир два броја која уноси корисник.**

**Решење:**



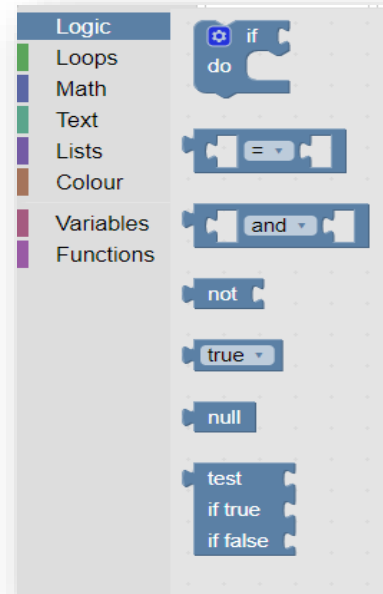
Када покренемо програм (кликом мишем на иконицу за покретање програма) кориснику се тражи да унесе број а, затим када унесемо а корисник ће кликнути на потврди и затим унети б, на крају добијамо збир та два броја. Ово је пример програмирања вођеног догађајима и руковања догађајима.

## 2.1 Упознавање основних стандардних компоненти и догађаја

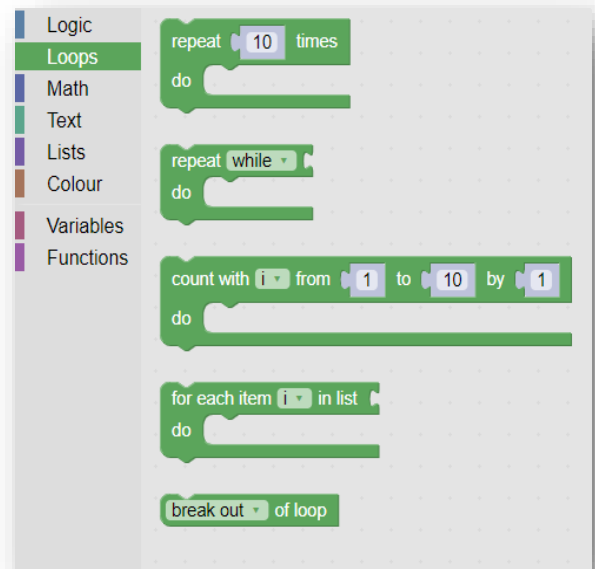
Покретањем програма отвара се неколико прозора које ћемо укратко описати.

Са леве стране мени са иконицама.

**Logic**-део за логичке операције

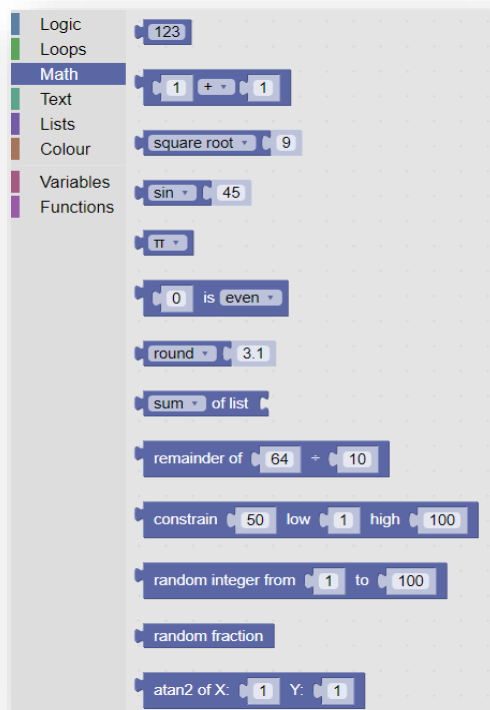


**Loops**-петље (repeat,while,do while,for...)

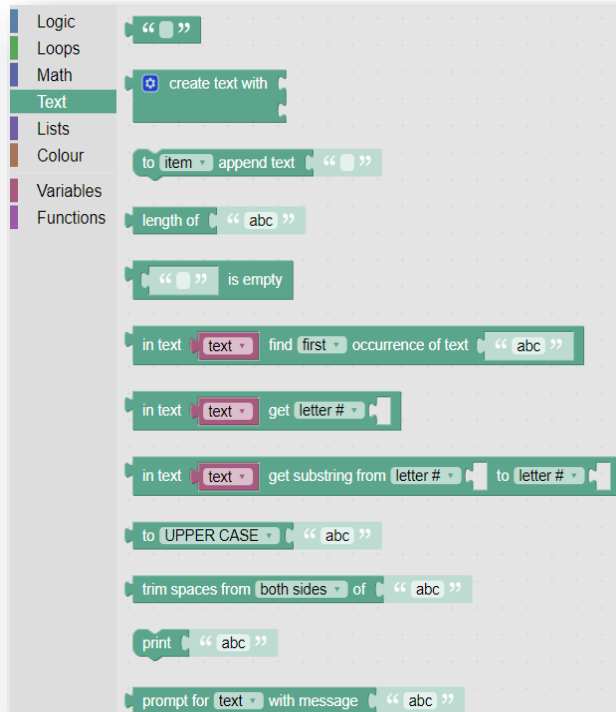




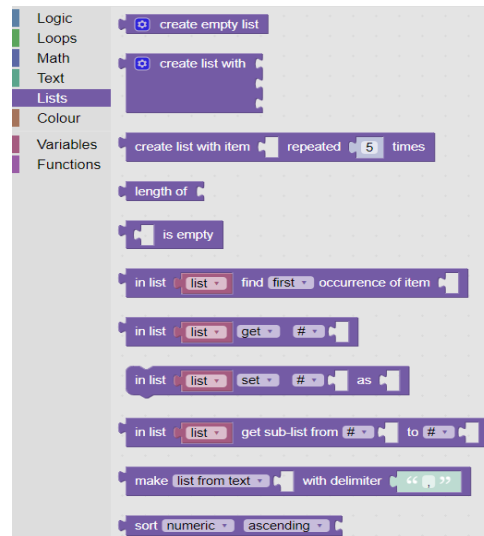
**Math-** математичке операције (сабирање, одузимање, множење, дељење, кореновање...)



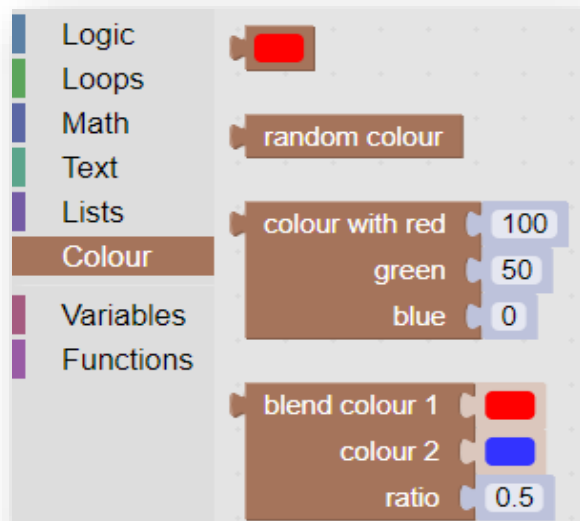
**Text-** Испис текста, промпт наредба за унос текста или бројева од стране корисника...



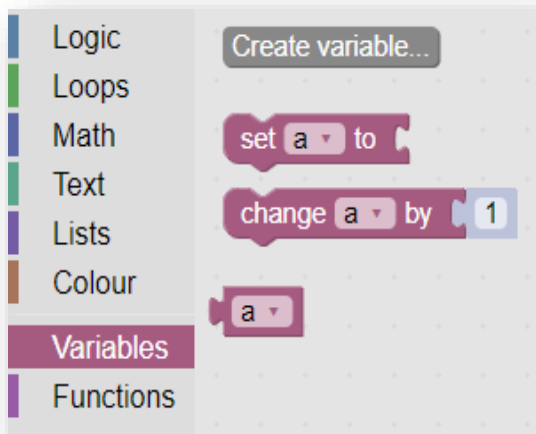
## Lists- Креирање листи



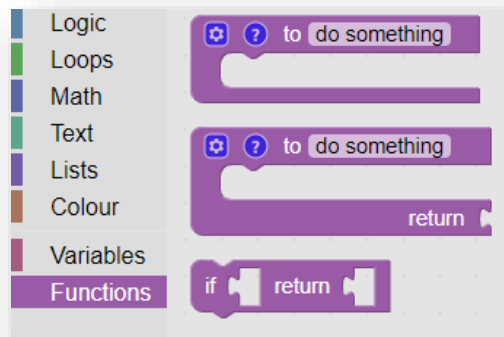
## Colour- Боје



**Variables-** Део у коме можемо креирати варијабле како би их могли искористити за наш програм

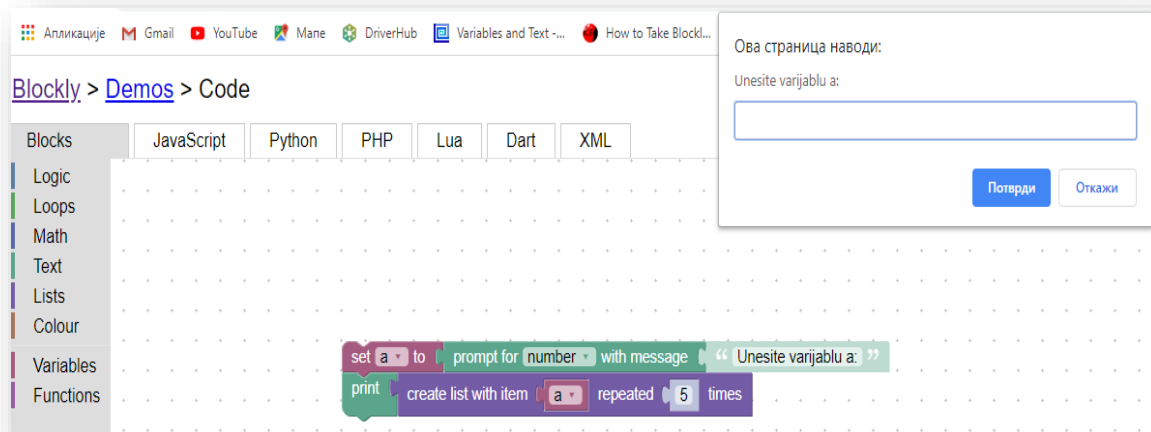


**Functions-** Функције које можемо креирати, дати им назив и касније их позвати у програму



У средини је радно окружење, где програмирамо користећи блокове, а затим можемо изабрати један од програмских језика: JavaScript, Python, Php, Lua, Dart, XML, и превести наш блок програм у код који касније можемо имплементирати у нашу апликацију.

**Пример.** Програм који испишује број неку варијаблу коју корисник уноси 5 пута.



### Неки примери у Blockly

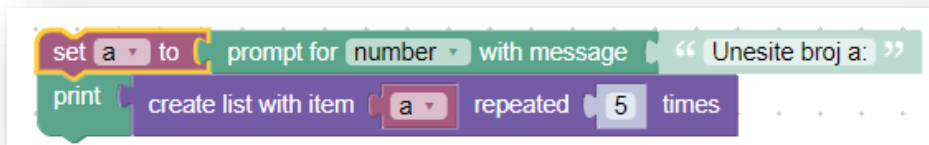
1. Израчунај следећи израз:  $2 + (3 * 5)$

#### Решење:



2. Креирај листу са 5 бројева које корисник уноси са тастатуре.

#### Решење:



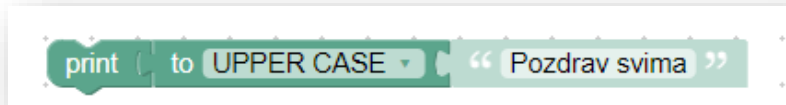
3. Написати програм који рачуна корен унетог броја.

#### Решење:



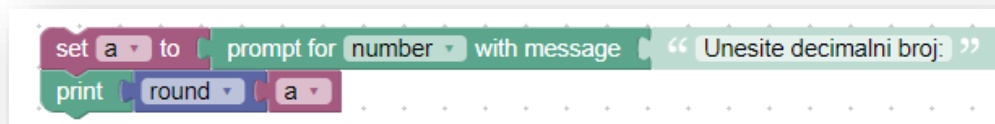
4. Пребацити текст “Pozdrav svima” у велика слова помоћу функције у Blockly.

**Решење:**



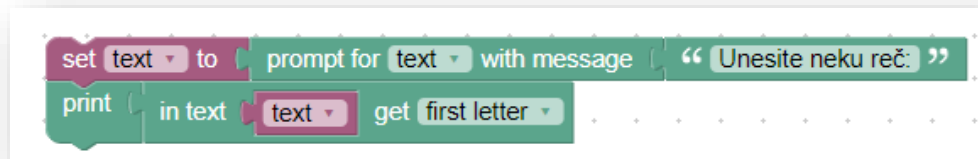
5. Заокружити број који корисник унесе помоћу **round**.

**Решење:**



6. Издвојити прво слово из неке речи коју унесе корисник.

**Решење:**



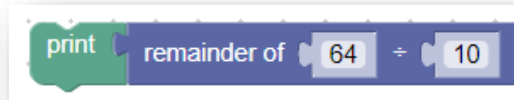
7. Израчунати cos од  $\pi$ .

Решење:



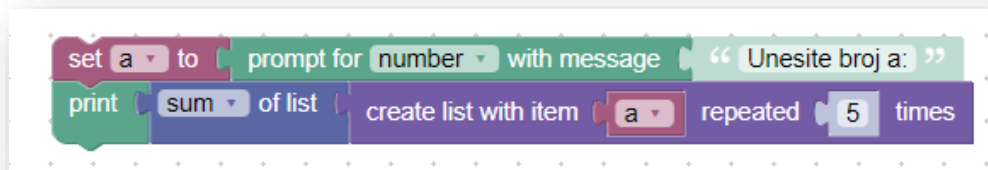
8. Израчунати колики је остатак при дељењу броја 64 са 10?

Решење:



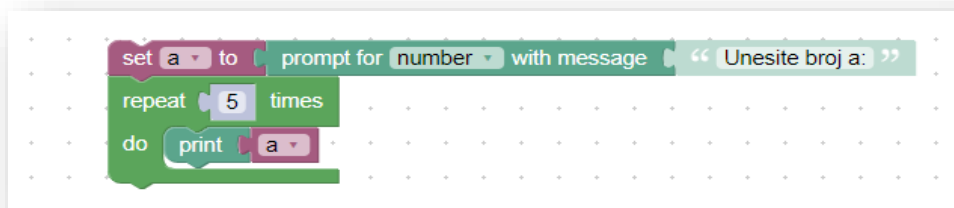
9. Направи програм помоћу којег се креира листа а затим се сумирају елементи те листе.

Решење:



10. Исписати 5 пута варијаблу коју корисник уноси (помоћу репеат петље).

Решење:



### 3. Основни елементи програмских језика JavaScript и PHP

JavaScript је најпопуларнији језик на интернету, и функционише на свим познатијим претраживачима, као што су Интернет Explorer, Mozilla-Firefox, Netscape, Opera.

Шта је JavaScript?

· JavaScript је дизајниран да би се додала интерактивност HTML странама · JavaScript је скрипт језик (једноставан програмски језик)

- JavaScript се састоји од линија кода који може да се извршава
- JavaScript се обично директно уграђује у HTML стране
- JavaScript је интерпретерски језик (што значи да се извршава без претходног компајлирања) · JavaScript није лиценциран језик

Да ли је JavaScript исто што и Java?

НЕ! Java и JavaScript су по концепту и начину програмирања два потпуно различита језика. Java (коју је развио Sun Microsystems) је комплекснији програмски језик, у истој категорији као C и C++.

#### Основни типови података

Основни типови података у JavaScript -у су:

- Нумерички (Number)
- Текстуални (String)
- Логички (Boolean)
- Симболички (Symbol)

Ту су затим и две "недефинисане" вредности, које су толико специфичне да представљају засебне типове:

- Недефинисана вредност (undefined)
- Нулта вредност (null)

Све остало што може да се нађе у JavaScript programу, спада у:

- Објекат (object)

Нумерички литерали се бележе цифрама 0 до 9, са водећим минусом за негативне бројеве и децималном тачком ако имамо децимале.

## Примери:

42, -231, 3.1415

Такође, могуће је задати и бројеве у бинарном, окталном и хексадекадном систему:

- Бинарни - бројеви се састоје само из цифара 0 и 1, и почињу са 0б или 0Б ("нула-Б")
- Октални - бројеви се састоје из цифара од 0 до 7, и почињу са 0 ("нула" - по старом стандарду) или 0о ("нула-О" - по стандарду ES2015)
- Хексадекадни - бројеви се састоје из цифара од 0 до Ф, и почињу са 0х или 0Х ("нула-Х")

```
0b01101011 // број 107 (binarno)
053         // број 43 (oktalno)
0o53        // број 43 (oktalno - po novom standardu)
081         // број 81 - ako su cifre iznad 7, gleda se kao dekadni
0x1f        // број 31 (heksadekadno)
```

+0 и -0 - да, и ово је "ствар" у JavaScriptu. "Позитивна нула" је исто што и обична вредност 0, а у већини случајева нећемо приметити неку разлику ако користимо "негативну нулу", осим што при дељењу са нулом можемо добити "плус бесконачно" или "минус бесконачно".

- Infinity и -Infinity - "плус бесконачно" и "минус бесконачно". Ове вредности се у математичким изразима понашају онако како смо навикли у математици. Нпр. дељење неког броја са Инфиниту као резултат враћа нулу.
- NaN - посебан случај броја је и *Not a Number*, односно вредност коју добијамо као резултат неке немогуће математичке операције (корен од -1, множење текста и броја, покушај да претворимо текст у број и сл.).

Још неке специјалне нумеричке вредности су дефинисане у објекту Number.

Овде ћемо их само поменути:

- Number.POSITIVE\_INFINITY - исто као Infinity
- Number.NEGATIVE\_INFINITY - исто као -Infinity
- Number.MAX\_VALUE - највећи могући број у JavaScriptu
- Number.MIN\_VALUE - најмањи могући број у JavaScriptu
- Number.MAX\_SAFE\_INTEGER - највећи могући цели број за који су очуване све цифре (сећате се кад смо рекли да нумерик у JavaScriptu има око 16 прецизних цифара? Е, то је то. Вредност је око 9 квадриљона.)
- Number.MIN\_SAFE\_INTEGER - најмањи "сигурни" цели број у JavaScriptu - очигледно, -9 квадриљона.
- Number.EPSILON - најмања могућа разлика између два броја у JavaScriptu, можемо рећи - најмањи позитиван број:  $2^{-52}$



## Текстуални тип

Осим нумеричког, постоји и текстуални, односно String тип.

Овај тип представља било који текстуални податак у нашем програму.

JavaScript

```
var s = "WEBnSTUDY"; // променљива s, типа string, добија вредност 'WEBnSTUDY'
var z1 = s.charAt[3]; // има вредност 'n' - четврти знак stringa
var z2 = s[3];        // има вредност 'n' - четврти знак stringa (ES5)
s[3] = "X";          // ово НЕЋЕ РАДИТИ
```

String литерали се задају унутар наводника или апострофа. Потпуно је свеједно да ли задајемо string на један или други начин:

**"Ovo je string."**

**'I ovo je string.'**

**`Šablonski string sa vrednošću \${x+1}`** // у string убације вредност x увећану за 1

**""** // празан string - два **navodnika** један до другог

**"** // празан string - два **apostrofa** један до другог

Дужина празног стринга је нула знакова. У логичким изразима празан стринг се евалуира као false.

## Логичке вредности

Трећи и наизглед најједноставнији тип је **логички тип**. Има само две могуће вредности:

- true - тачно
- false – нетачно

Међутим, када се врши логичка евалуација, JavaScript такође врши имплицитну конверзију типова. То суштински значи да када се проверава неки услов, ако вредност није логичког типа, она се своди на логичку вредност. Ево шта се "рачуна" као **false**:

- 0 - нумеричка нула се рачуна као false, а све остале вредности (и позитивни и негативни бројеви) су true
- NaN - Not a Number је исто false
- "" i " " - празан string се рачуна као false, сви остали stringovi су true
- null - "nulti" објекат је false
- undefined - недефинисана вредност је такође false

Све остале вредности се рачунају као **true**. Дакле, било какав string, било који број (позитиван или негативан, чак и Infinity), било какав објекат или датум. Пазите, чак и "празни" објекти су и даље true. Ево неких вредности за које се погрешно мисли да ће бити false:

- "0" - string који се састоји од цифре "nula" је true
- " " - string који се састоји од једног размака је true
- {} - празан објекат је true
- [] - празан низ се ипак рачуна као true

### Вредности undefined

Ова вредност описује стање недефинисаности. Вредност **undefined** имају:

- променљиве које покушавамо да користимо у изразима, а нигде раније нису декларисане
- променљиве које су само декларисане са var или let, а никад им није додељена вредност
- параметри функције који нису добили вредност приликом позива функције (формални параметри који нису "покривени" стварним вредностима)

## 3.1 Основни типови података у Php- у су:

1. **Integer**- целобројни тип
2. **Double** или **float**- покретни зарез двоструке прецизности, користи се за реалне бројеве
3. **String**- знаковни тип
4. **Boolean**- логички тип
5. **Array**- низ, користи се за чување више података истог типа
6. **Object**- објекат, користи се за чување примерка (instance) једне класе

Два специјална типа:

1. **Null**- променљиве којима није додељена вредност, које су недефинисане или којима је изричито додељена вредност null
2. **Resource**- неке уграђене функције (нпр. За рад са базама података)

Тип података код Php- а се не наводи експлицитно код променљивих, већ се одређује на основу вредности која јој је додељена.

### Примери:

```
$x = 8980;    // позитиван број
$x = -129;    // негативан
$x = 02266;   // октални број
$x = 0x3F;    // хексадекадни број
$x = 0b110010111; // бинарни број
```

### Бројеви у покретном зарезу

Ово су бројеви са децималама. У меморији се бележе као бројеви у покретном зарезу. Број прецизних цифара варира од платформе до платформе, мада бисмо увек требали да можемо да рачунамо на бар 14 прецизних цифара. Децимални знак је увек тачка.

```
$a = 3.14;    // број у фиксној нотацији
$a = 2.3e5;    // број у покретном зарезу
$a = 4.3E-2;   // број у покретном зарезу
```

### Логичке вредности

Постоје само две логичке вредности: true и false. Небитно је да ли их пишемо великим или малим словима.

```
$b = false;   // број у фиксној нотацији
$b = 0;       // тумачи се као false
```

```
$b = 0.0;    // тумачи се као false
$b = "";     // тумачи се као false
$b = "0";    // тумачи се као false
```

## Стрингови

```
$t = Ово је обичан текст с \' апострофон.';
// ТЕКСТ: Ово је обичан текст с ' апострофом.
```

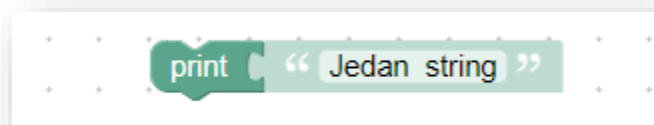
У **Blockly** програмском језику имамо блокове података, као на пример логички блок података (**Logic**) у коме се налазе наредбе if затим **Loops** где се налазе петље: while, do while, repeat, repeat while... **Math** је део за математичке операције. Blockly има типове података као и php то су : integer, float, string (Text), такође и null. Затим имамо блок података **Colour** где можемо бирати боју.

Код Blockly програмског језика као и код Php- а не морамо експлицитно дефинисати тип податка променљиве, већ се он одређује на основу вредности која јој је додељена.

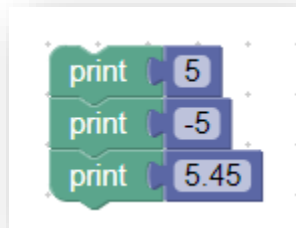
**Пример.** Једна уграђена random функција, која враћа неки број од 1 до 100 типа integer.



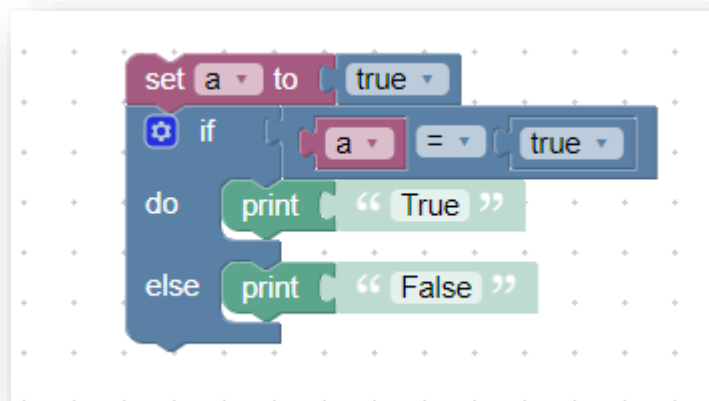
## String



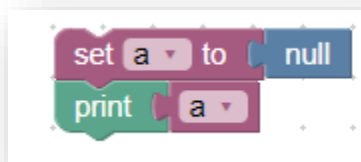
## Бројеви (позитивни, негативни, децимални)



## Boolean



## Null вредност



## 3.2 Преглед оператора

У изразима учествују два типа елемената: операнди и оператори. Операнди (константе, променљиве, позиви функција) су вредности са којима се ради, док су оператори ознаке операција које се изводе (сабирање, множење, упоређивање...).

Да би израз као целина имао вредност, морају је имати и сви операнди у тренутку евалуације (израчунавања) изрази.

Blockly програмски језик омогућава нам да програмирамо помоћу блокова, затим да те блокове преведемо у код бирајући један од понуђених програмских језика. У следећим лекцијама бавићемо се Blockly програмским језиком и превођењем блокова у JavaScript програмски језик.

1. Направити програм у Blockly програмском језику који израчунава следеће изразе, и штампа резултате. Након тога превести у JavaScript kod.

- $2+3$
- $7-5$
- $2*6$
- $9/3$

### Решење

#### Blockly



## JavaScript

```
window.alert(7 - 5);
```

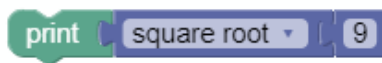
## Php

```
print(7 - 5);
```

2. Направити програм који рачуна корен броја 9, и исписује резултат:

## Решење

## Blockly



## JavaScript

```
window.alert(Math.sqrt(9));
```

## Php

```
print(sqrt(9));
```

3. Израчунати вредност израза:

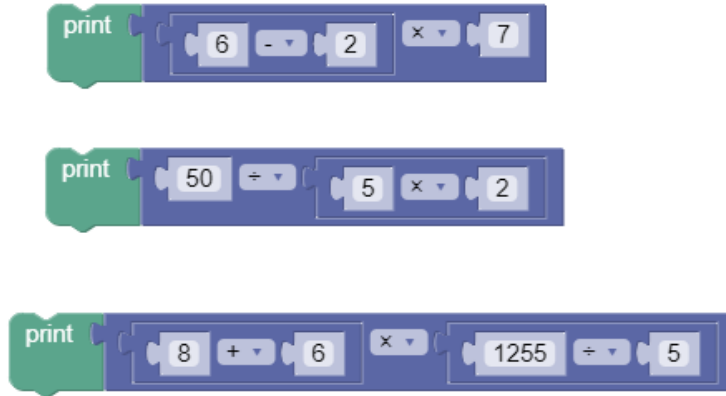
$(6 - 2) * 7$

$50 / (5 * 2)$

$(8 + 6) * (1255 / 5)$

## Решење

### Blockly



### JavaScript

```
window.alert((6 + 2) * 7);  
  
window.alert(50 * (5 / 2));  
  
window.alert((8 + 6) * (1255 / 5));
```

### Php

```
print((6 - 2) * 7);  
  
print(50 / (5 * 2));  
  
print((8 + 6) * (1255 / 5));
```

4. Направити програм који израчунава вредност израза и тај резултат чува у променљивој резултат.

$$3*5+1$$



## Решење

### Blockly



### Php

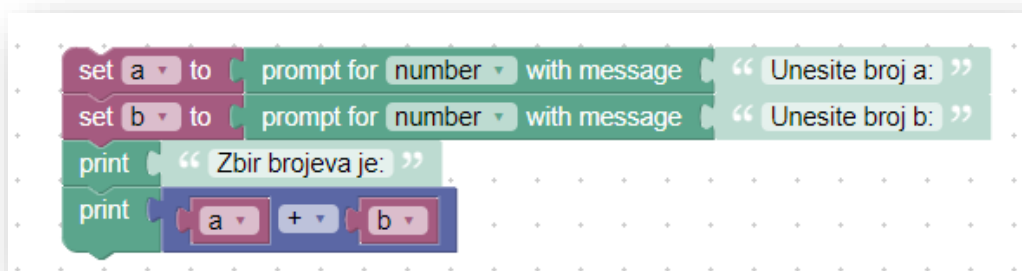
```
$rezultat;
```

```
$rezultat = 3 * 5 + 1
```

### 3.2.1 Izrazi, konverzije podataka

Израз је у програмирању било која исправна комбинација операнда повезаних операторима.

### Пример:



У php- у:

```
$a;  
$b;  
  
$a = floatval(readline('Unesite broj a:'));  
$b = floatval(readline('Unesite broj b:'));  
print('Zbir brojeva je:');  
print($a + $b);
```

## 4. Конверзија типова података

Конверзија типова података омогућује претварање варијабле из једног типа у други тип варијабле. Конверзија типова података може бити експлицитна или аутоматска.

### 4.1 Експлицитна конверзија типова података (eng. Type Casting)

Експлицитна конверзија експлицитно наводи у који тип података желимо извршити конверзију преко оператора конверзије.

Оператори

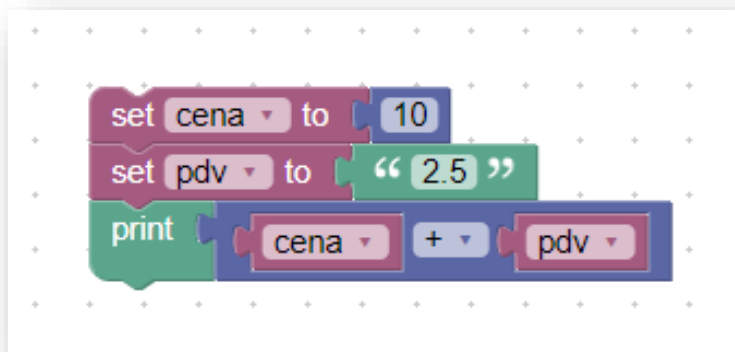
- (array)
- (bool) или (boolean)
- (int) или (integer)
- (object)
- (real) или (double) или (float)
- (string)

## 4.2 Аутоматска конверзија типова података (eng. Type Juggling)

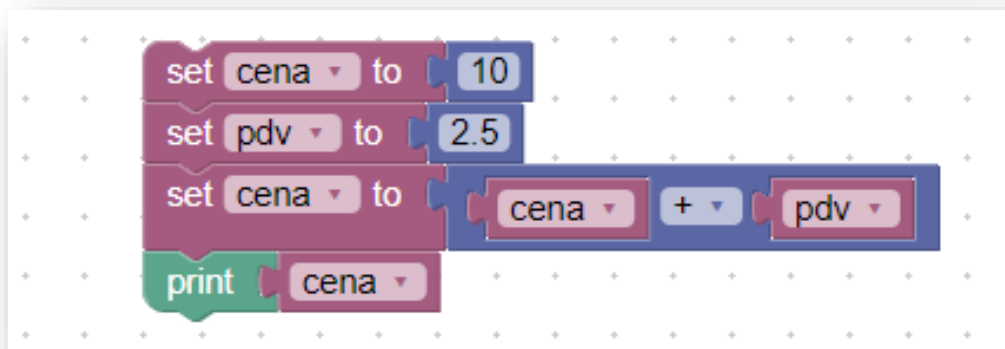
JavaScript И PHP не захтевају ни не подржавају експлицитну дефиницију типа података приликом декларисања варијабли. Тип варијабле се одређује из контекста у којем се варијабла користи. Уколико се варијабли додели вредност типа стринг она ће постати варијабла истог типа, уколико јој се додели интегер вредност и варијабла се мења у тај тип података.

### Blockly

Blockly не може да аутоматски стринг вредност пребаци у интегер уколико је то потребно. Следећи пример у Blockly-ју не ради, јер Blockly pдv посматра као string и не конвертује дату вредност у integer као што је случај за JavaScript.



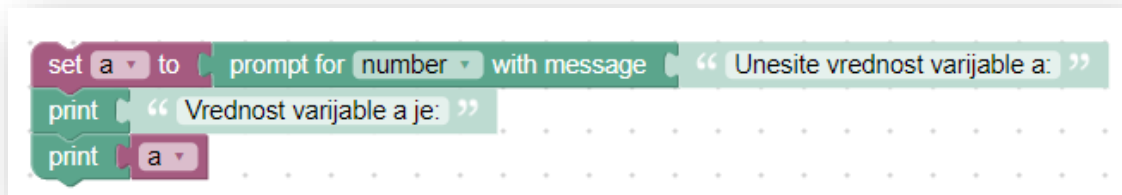
Када варијаблу дефинишемо као децимални број, програм ће радити и конвертовати варијаблу цена у децимални број, иако је претходно била дефинисана као integer вредност.



### 4.3 Уношење и приказ података

У Blockly програмском језику корисник може да уноси податке помоћу наредбе **prompt**. Ова наредба има две опције: унос текста И унос бројева. На пример креирамо неку варијаблу **a** И од корисника тражимо да помоћу промпт команде унесе вредност те варијабле. Приказ или исписивање података вршимо помоћу **print** команде.

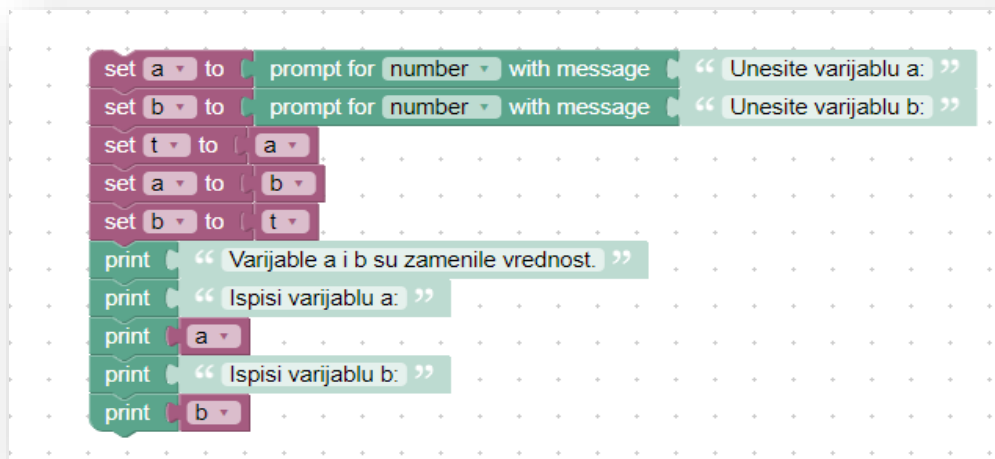
Пример:



### 4.4 Алгоритми линијске структуре

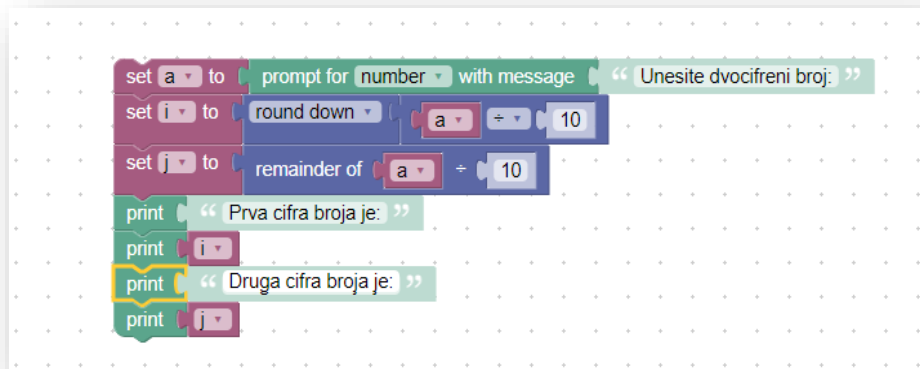
Пример 1. Алгоритам замене две вредности:

Blockly

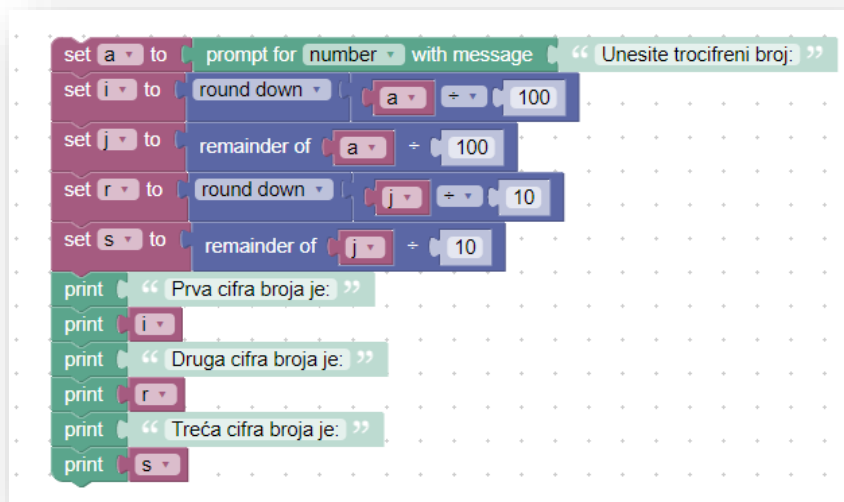


Пример 2: Алгоритам- издвајање цифара у запису двоцифреног/троцифреног броја:

### Blockly (двоцифрени број)

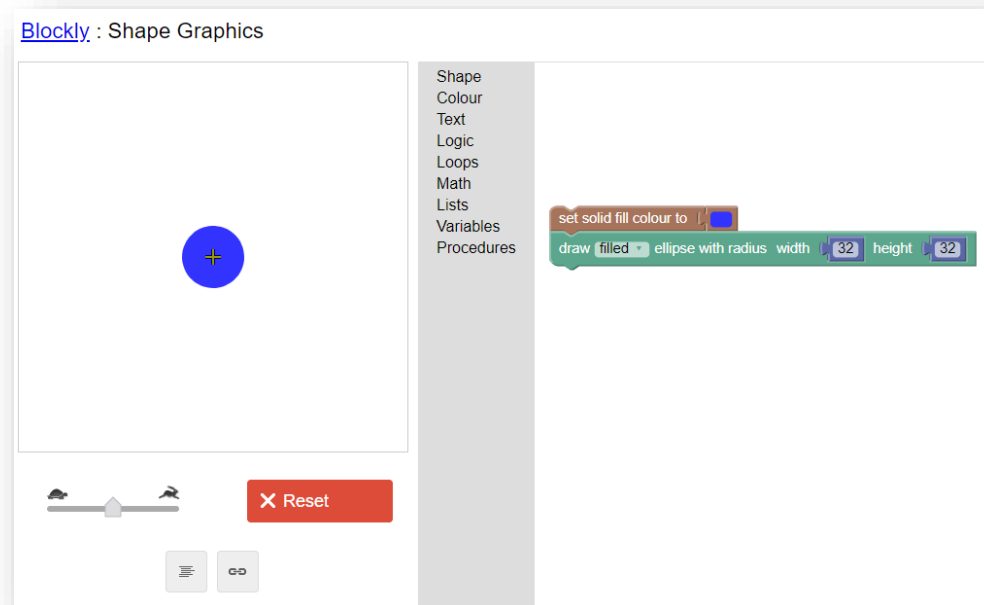


### Blockly (троцифрени број)



## 5. Цртање разних облика

### Blockly (круг)



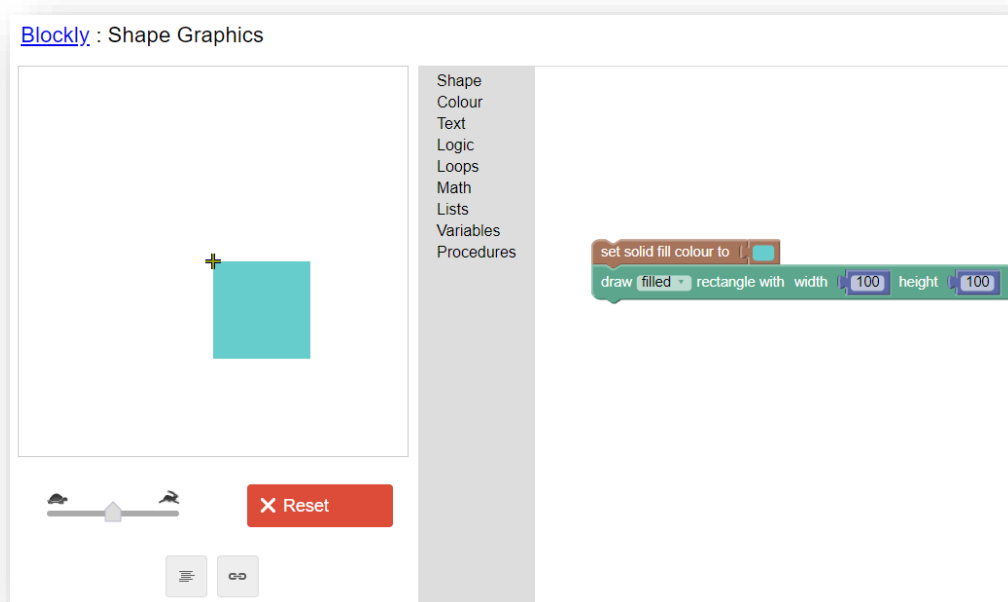
Слика 1. Круг

### JavaScript (круг)

```
var y, x;

draw.circle(32, 32, 100, '#ff0000', '#990000', 1);
for (y = 0; y <= 800; y += 10) {
  for (x = 0; x <= 800; x += 15) {
    draw.line(0, 0, x, y, '#ffcc33', 0.1);
  }
}
```

## Blockly(четвороугоао)

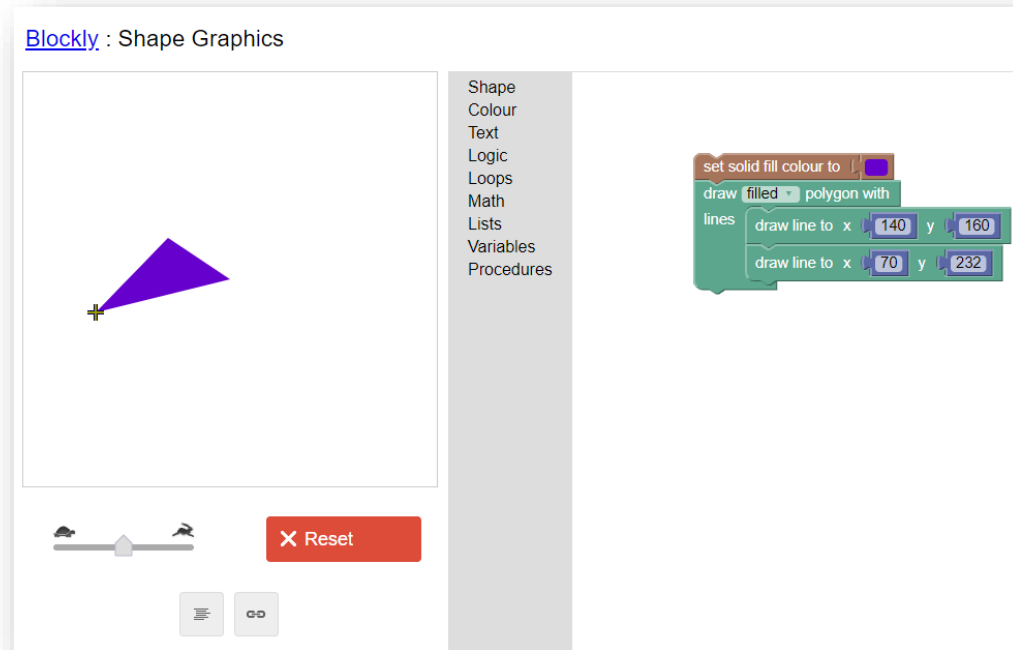


Слика 2. Четвороугоао

## JavaScript (четвороугоао)

```
draw.rectangle(250, 200, 350, 350, '#339999', '#000000', 1);
```

## Blockly (τρουγαο)



Слика 3. Τρουγαο

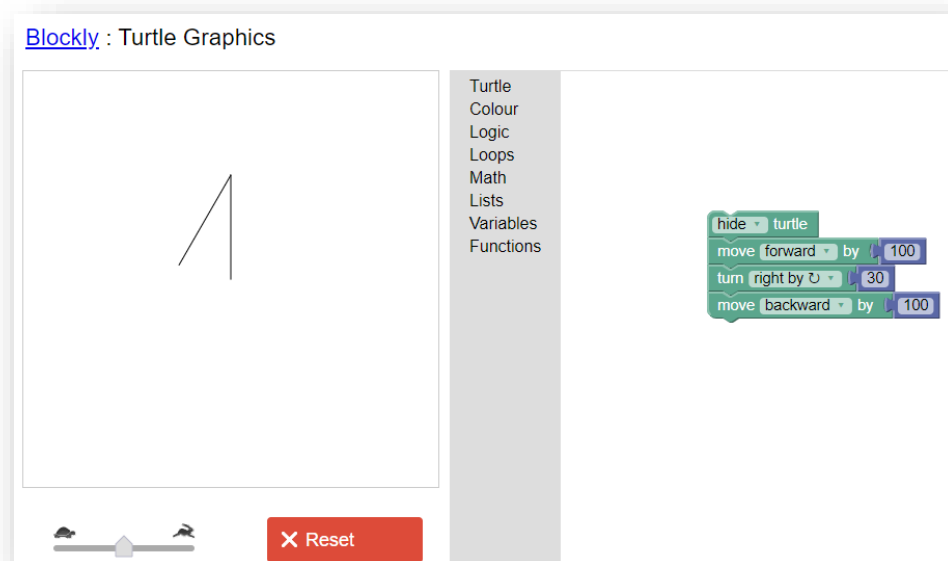
## JavaScript (τρουγαο)

```
draw.triangle(50, 50, 150, 150, 300, 100, '#6600cc', '#6600cc', 1);
```

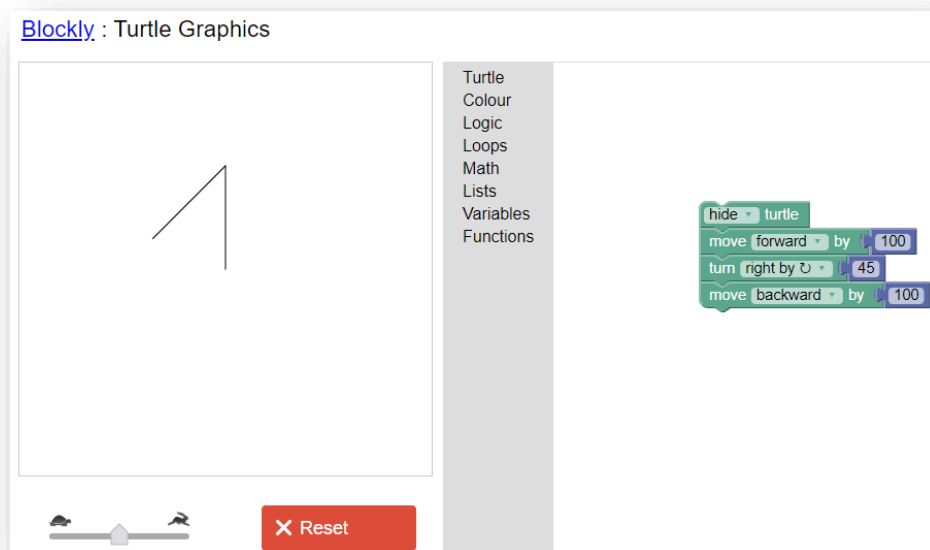


## 5.1 Цртање углава

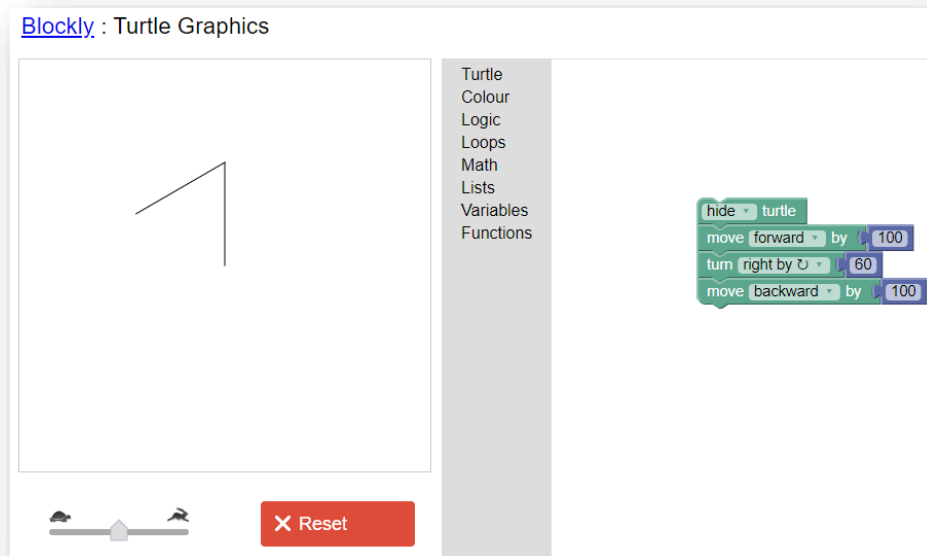
Угао од  $30^\circ$



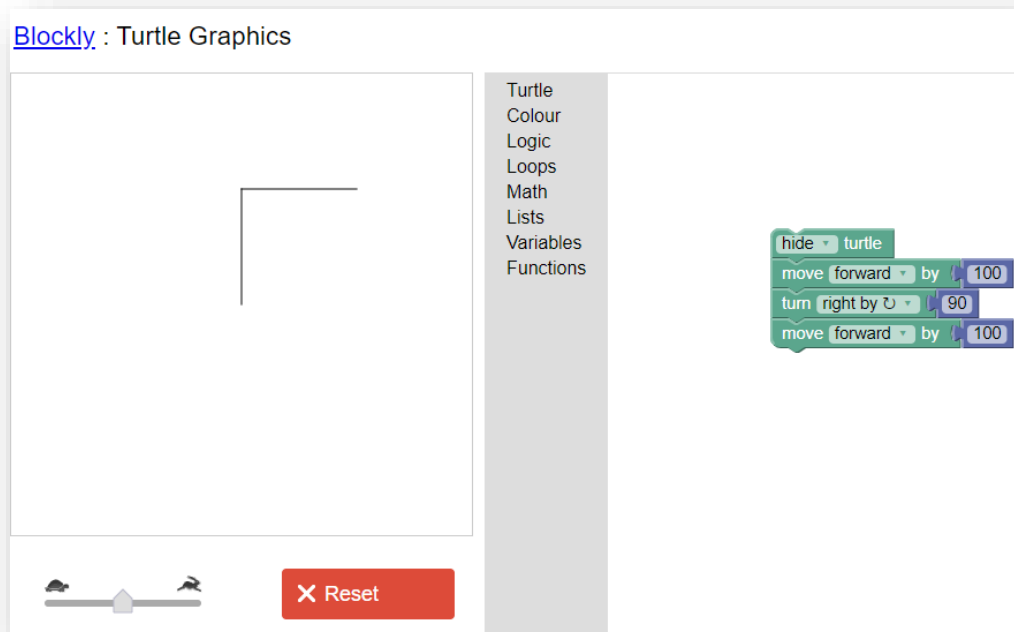
Угао од  $45^\circ$



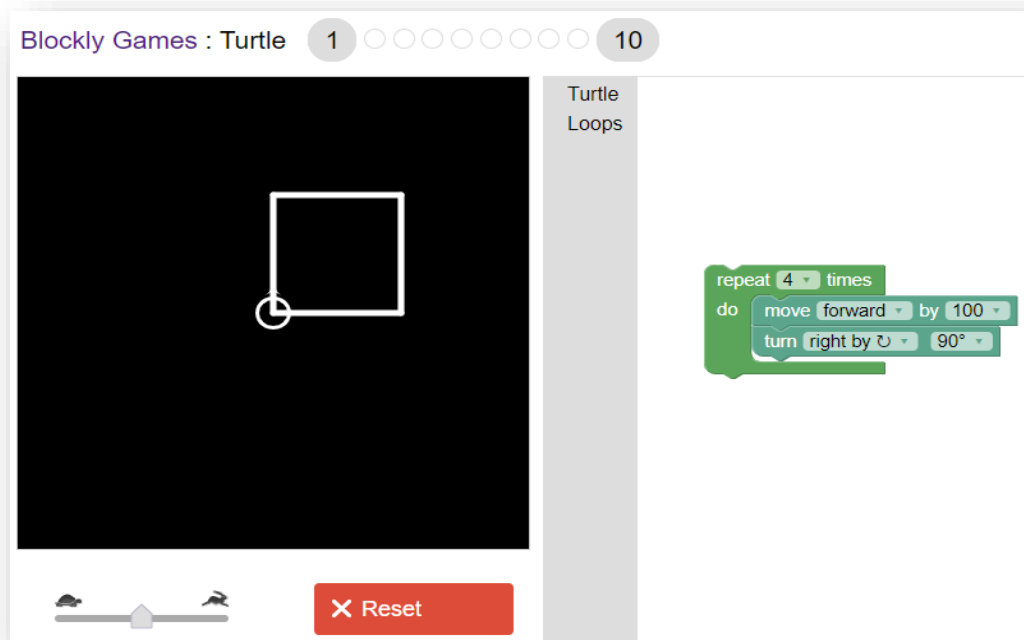
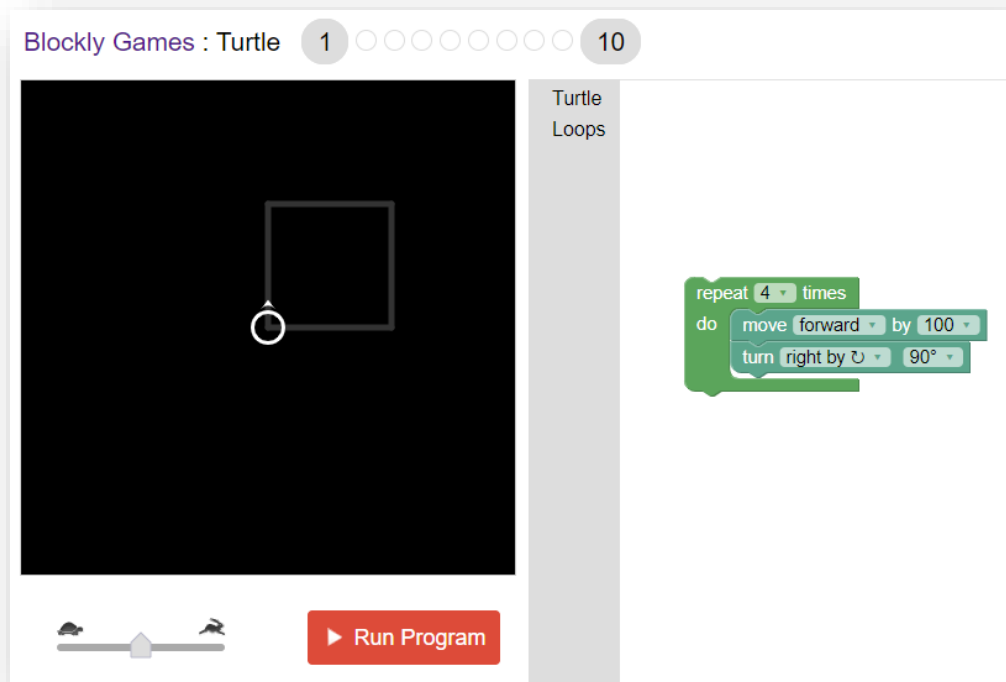
Угао од  $60^\circ$



Угао од  $90^\circ$

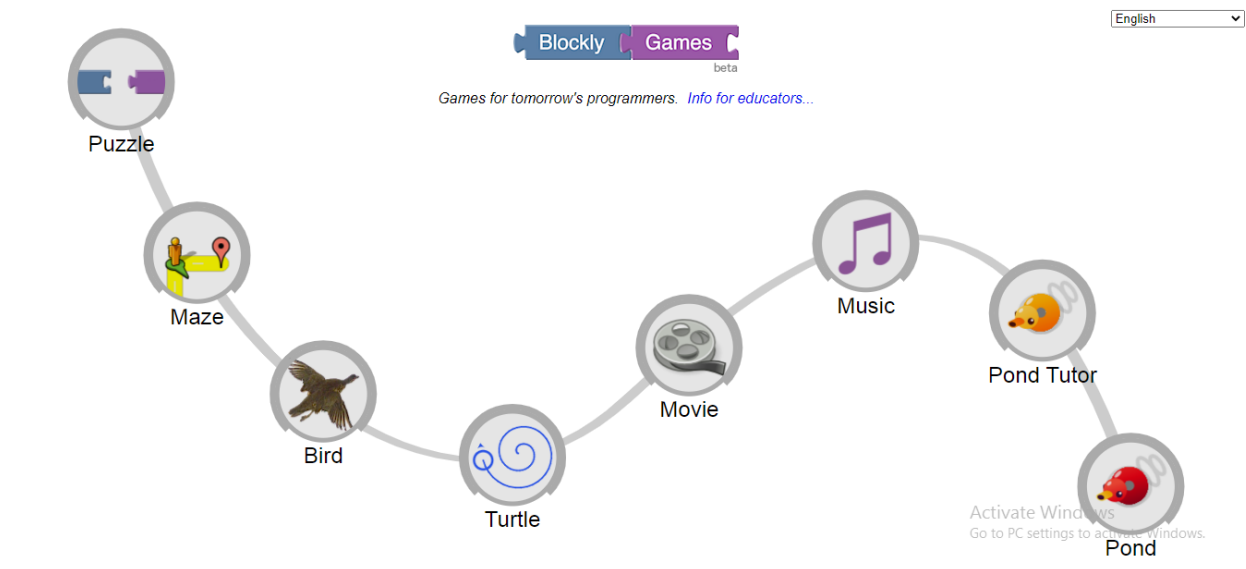


## 5.2 Померање лоптице (Blockly)



Обични Blockly едитор не можемо користити за прављење апликација и померање лоптице, као ни цртање углова па самим тим немамо код за JavaScript. Апликације можемо правити помоћу Blockly Games: Turtle.

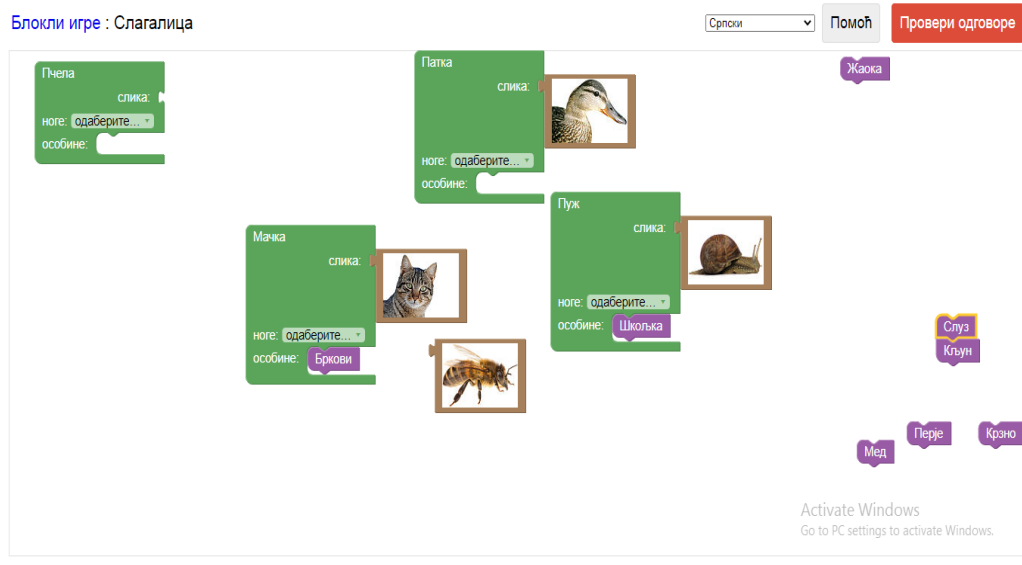
## 6. Blockly Games



Слика 1.

Blockly Games су серија едукативних игрица које уче основним корацима програмирања . Дизајнирана је за децу која нису имала претходно искуство у програмирању. На крају игре играч би могао бити спреман да користи конвенционалне текст базиране програмске језике ( то је тачно али за енглеско говорно подручје, нашој деци треба мало дуже).

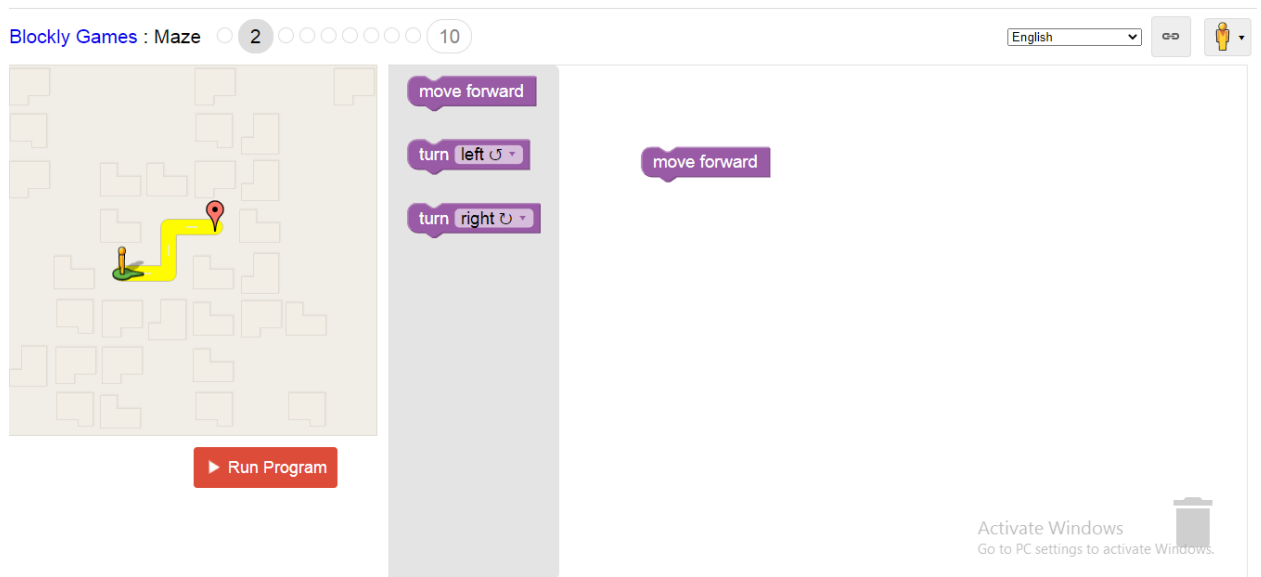
## 6.1 Puzzle - Слагалице



Слика 2.

Puzzle – Слагалице су брзо упознавање са Блокли облицима и могућности њиховог уклапања. Потребно је да се са сликом животиње упаре неке њене особине тако што ће се делоби- блокови превлачити.

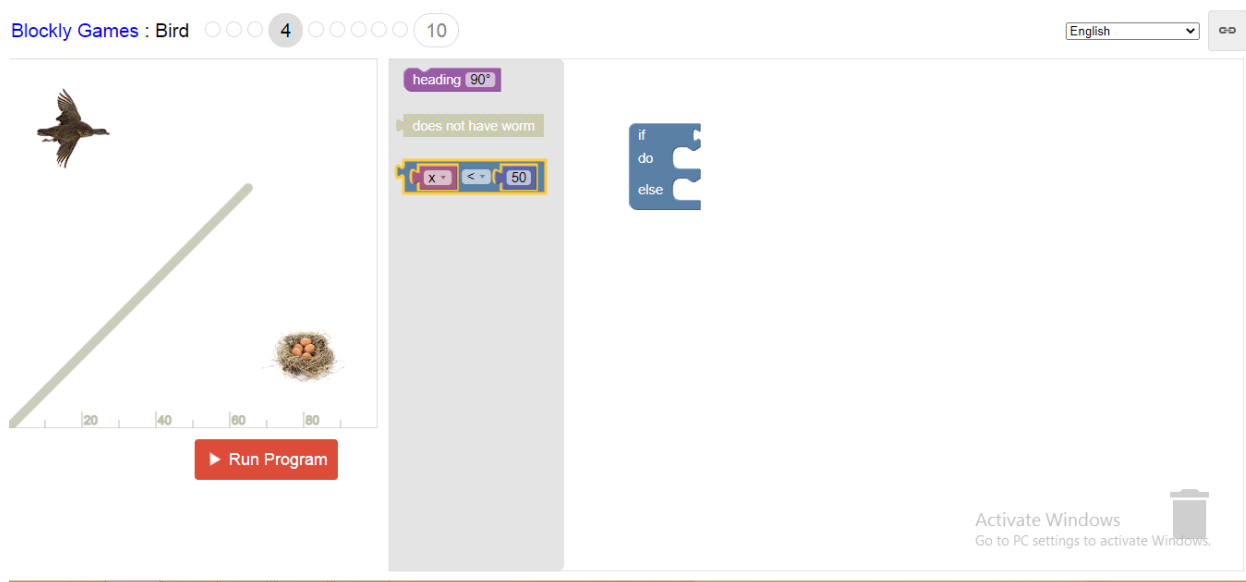
## 6.2 Maze – Лавиринт



Слика 3.

Maze – Лавиринт је увод у петље и условна кретања. Почиње веома једноставно али сваки нови ниво је сложенији. Захтева да ученик увиди правило по коме треба да се креће до циља и примени слажући команде као блокове. Сваки пут када реши ниво програм му нуди могућност да види како решење изгледа написано у програмском језику JavaScript.

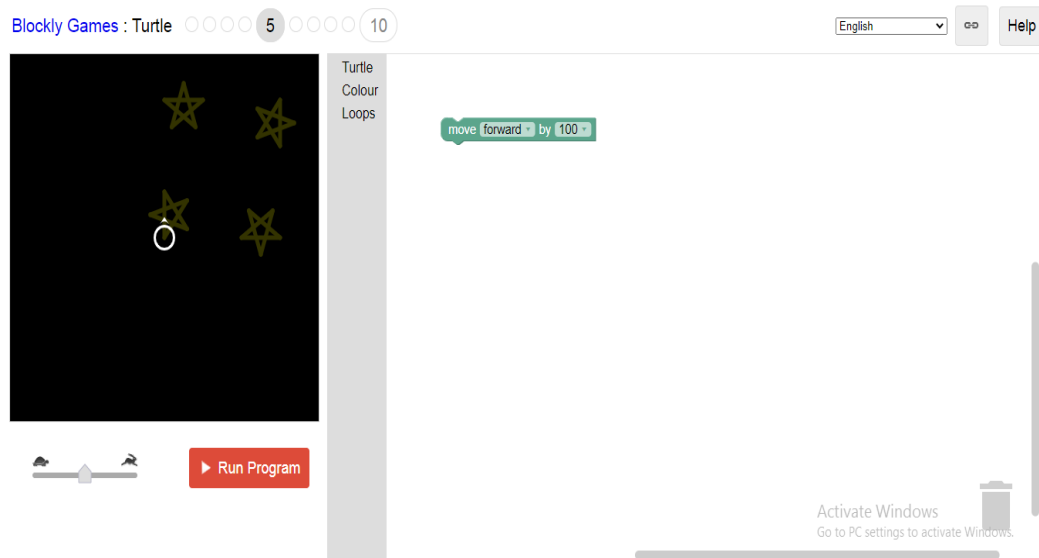
### 6.3 Bird – Птица



Слика 4.

Bird – Птице су већ озбиљније бављење питањима „ако- онда“. Деца морају уочавати услове по коме треба поступити да би решили проблем. У овом случају птици треба мењати угао летења како би извршила задатак а успут заобишла одређене препреке. И ова активност се постиже слагањем блокова али и променом параметара унутар њих.

## 6.4 Turtle- Корњача



Слика 5.

Turtle – Корњача је савладавање петљи. Потребно је користити уметнуте петље како би нацртали задату слику геометријског облика.

## 6.5 Movie – Премести



Слика 6.

Movie– Премести је увод у математичке једначине. Потребно је употребити математичку једначину да би направио жељену промену. Занимљиво је док мењате параметре на слици са стране видите како сте нешто подесили и да ли је то урађено како треба.

## 6.6 Pond Tutor – Рибњак тутор

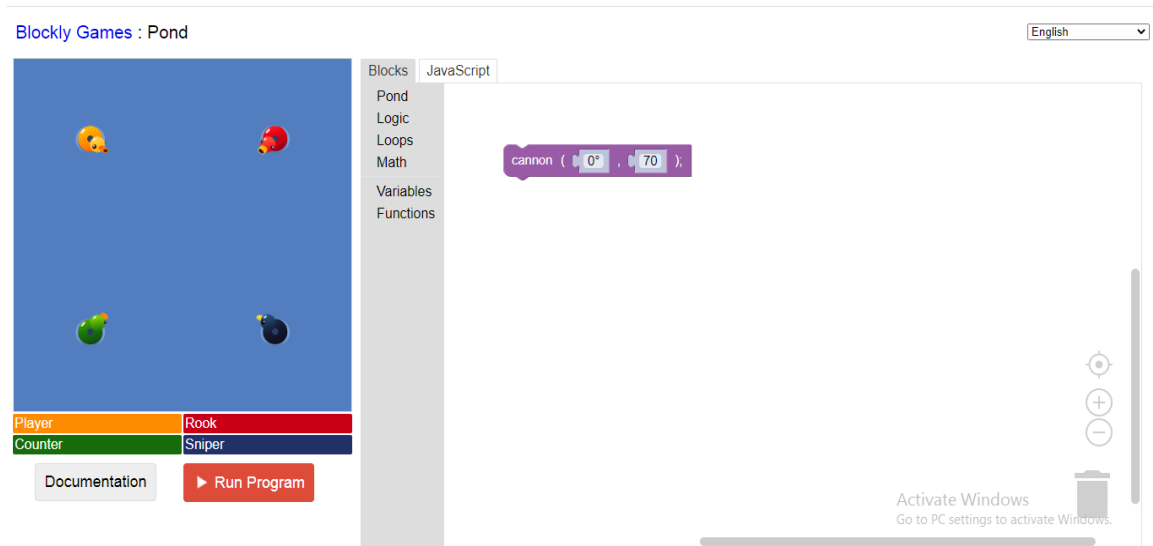


Слика 7.

Pond Tutor– Рибњак тутор је прави увод у текстуално базирано програмирање користећи програмски језик JavaScript. Током нивоа пребацују се између блокова и текст editora. Навођењем помоћу блокова или текст едитора треба из малог топа погодити паткицу у рибњаку.



## Pond – Рибњак



Слика 8.

Pond– Рибњак је такмичење отвореног типа да програмирањем победите најпаметнију патку. Користите или блокове или JavaScript.

### Задатак 1.

Састави програм који одређује да ли је уписани број дељив са 13.

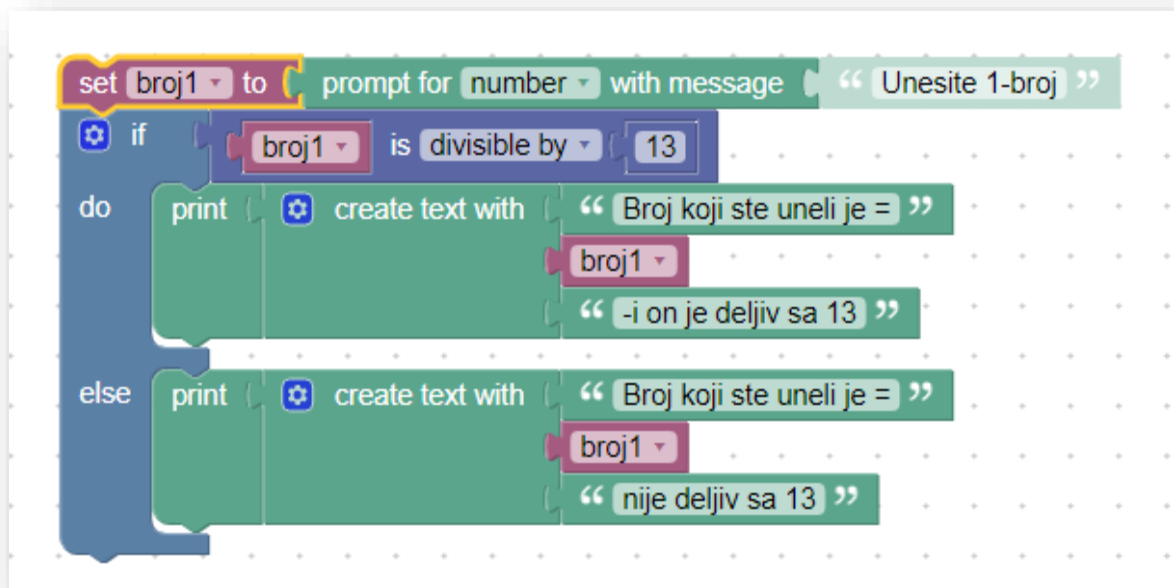
#### JavaScript

Листинзи који се добијају Blockly - интерпретером

```
var broj1;

broj1 = Number ( window.prompt ( 'Unesite 1-broj' ));
if ( broj1 % 13 == 0 ) {
  window.alert ( [ 'Broj koji ste uneli je =',broj1,'-i on je deljiv sa 13' ].join (""));
} else {
  window.alert ( [ 'Broj koji ste uneli je =',broj1,'nije deljiv sa 13'].join (""));
}
```

## Blockly код



```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2> Brojevi </h2>
```

```
<p id="demo"></p>
```

```
<script>
```

```
    var broj1;
```

```
    broj1 = Number(window.prompt('Unesite 1-broj'));
```

```
    if (broj1 % 13 == 0) {
```

```
        window.alert(['Broj koji ste uneli je =',broj1,'-i on je deljiv sa  
13'].join(''));}
```

```
    else {
```

```
        window.alert(['Broj koji ste uneli je =',broj1,'nije deljiv sa 13'].join(''));}
```

```
    }
```

```
</script>
```

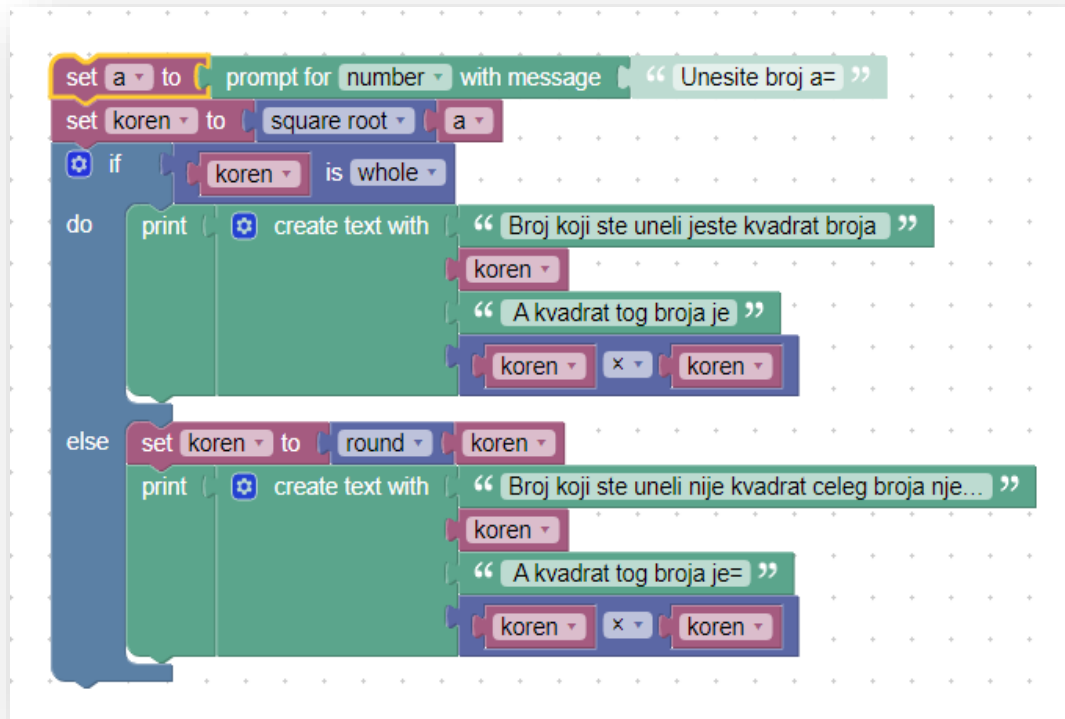
```
</body>
```

```
</html>
```

## Задатак 2.

Саставити програм који одређује да ли је уписани цео број квадрат неког целог броја ако јесте исписати тај број, а ако није исписати њему најближи квадрат целог броја.

## Blockly код



## JavaScript

Листинзи који се добијају Blockly - интерпретером

```
var a, koren;

a = Number(window.prompt('Unesite broj a='));
koren = Math.sqrt(a);

if (koren % 1 == 0) {
  window.alert(['Broj koji ste uneli jeste kvadrat broja ',koren,' A kvadrat tog broja je',koren * koren].join(''));
} else {
  koren = Math.round(koren);
  window.alert(['Broj koji ste uneli nije kvadrat celeg broja njemu najblizi kvadrat je kvadrat broja =',koren,' A kvadrat tog broja je=',koren * koren].join(''));
}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2> Kvadrat broja </h2>
```

```
<p id="demo"> </p>
```

```
<script>
```

```
var a, koren;
```

```
a = Number(window.prompt('Unesite broj a='));
```

```
koren = Math.sqrt(a);
```

```
if (koren % 1 == 0) {
```

```
    window.alert(['Broj koji ste uneli jeste kvadrat broja ',koren,' A kvadrat  
tog broja je',koren * koren].join(""));
```

```
    } else {
```

```
        koren = Math.round(koren);
```

```
        window.alert(['Broj koji ste uneli nije kvadrat celeg broja njemu  
najblizi kvadrat je kvadrat broja =',koren,' A kvadrat tog broja je=',koren *  
koren].join(""));
```

```
    }
```

```
</script>
```

```
</body>
```

```
</html>
```

## ЗАКЉУЧАК

Стварање атмосфере погодне за рад и размену мишљења, одржавање проналазачког духа и праве помоћи у право време, у многome би помогло ученицима у савладавању градива. Стварајући атмосферу погодну за примену свих знања и вештина уз савремена средства за рад у многome подстиче продуктивност и крајње резултате рада. За ученике је битно да теоријско знање које усвајају путем наставе примене и кроз практичан рад. Треба променiti положај ученика у школи од улоге пријемника знања ка улози ствараоца свог знања. Етатизам који влада у образовним круговима је појава истрошености кадрова запослених у образовним установама. Увођење Блоклија је битан процес у настави информатике. Уз помоћ овог софтвера монотон час се претвара у креативан и занимљив час.

Мислим да би коришћење Блоклија изазвало интересовање код ученика и подстакло на самосталан рад, а уједно и стицање нових знања. Инспирација и домишљатост су непресушан извор могућности креативне индустрије. Увођењем модерних облика извођења наставе и тестирања усвојених знања у виду постављања практичних задатака ученицима, који су могући у реалном свету и од чијих знања и вештина се може живети, ученици бивају оспособљени за рад, а такође и много конкурентнији на тржишту рада. Задавањем вежбе кроз коју би самостално радили пројекте повећао би се број заинтересованих ученика. Најављивањем предстојећег градива јавило би се интересовање за самосталан рад и показивање добијеног. Пустимо ученике да сами откривају математичке и остале креативне ствари и односе помоћу Блоклија, како би ученици добили више простора да траже своје путеве за самостално решавање постављеног задатка.

Учење је индивидуални процес тако да то можемо захтевати од ученика. Овај софтвер нам помаже да тај процес изгледа видљивији.

hipoteza

predmet istrazivanja i problem

predmet ovog istrazivanja je da se prikaze blockly aplikacije koja omogucuje da se pomocu grafickog programiranja približi koncept programiranja deci.

problem istrazivanja

kako uz pomoc blocklya poducavaju osnovnog obrazovanja u programiranju?

koji su osnovni elementi javascript i php?

koji su osnovni tipovi podataka?

kako se vrsi konverzija tipova podataka?

kako se pomocu blocklya crtaju geometrijski oblici?

cilj istrazivanja

istrazivanje ovog master rada je da se prikaze blockly i njegov potencijal u obucavanju dece o programiranju.

u ovom master radu cemo pokazati osnovne tipove podataka i njihovu konverziju,, takodje bice i objasnjeni osnovni elementi javascript i php-a

kroz prakticne promere u blockliyu bice priblizeno znanje o crtanju osnovnih geometrijski oblika pomocu grafickog prikaza u aplikaciji.

## ЛИТЕРАТУРА

- [1.] Мирослав Илић: Програмски језик ДЕЛФИ први кораци (Programiranje od A do Š), Београд, 2014.
- [2.] <https://developers.google.com/blockly>
- [3.] <https://blockly-demo.appspot.com/static/demos/code/index.html>
- [4.] <https://www.znanje.org/knjige/computer/JavaScript/12iv0311/tipoviPodataka.htm>
- [5.] <https://www.w3schools.com/js/>