Квиз на два начина

Разлика између обичних променљивих и листи најбоље се може уочити ако се исти проблем решава употребом једне, а затим и употребом друге врсте променљивих.

Задатак је да направиш један квиз, али на два начина.

Другим речима, пред тобом је креирање два програма у Scratch-у која дају исти резултат.

У првом програму, направићеш квиз помоћу обичних променљивих (које памте само један податак, вредност),

У другом програму, направићеш исти квиз помоћу листи (које представљају колекцију података, вредности).

Пре израде програма, потребно је да осмислиш до пет питања из једног или више наставних предмета које, у V разреду, учиш. За свако од питања треба да имаш записан тачан одговор.

Чим размотримо логику квиза, моћи ћеш да пређеш на програмирање.

Основни алгоритам за креирање квиза подразумева:

Постављање питања и чекање на одговор корисника

Упоређивање одговора корисника са тачним одговором

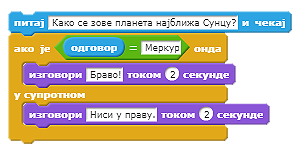
Извршавање једног низа блокова ако је одговор тачан, односно другог низа блокова ако је одговор нетачан.

Понављање прва три корака онолико пута колико има питања.

**Квиз помоћу *обичних* променљивих**

У првом програму користићемо *обичну* променљиву L10_PromOdgovor.

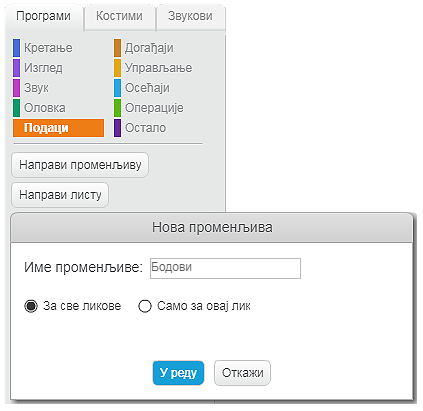
Нека питање буде **Која је планета најближа Сунцу?**. Ти знаш да је то **Меркур**.

[](https://petlja.org/biblioteka/r/lekcije/scratch-support/_images/L10_KvizObicanPitanje.png)

Низањем оваквих блокова, у којима се једино мењају питања и одговори, лако се креира квиз. Имаш пет осмишљених питања и одговора. Направи Sctarch програм који проверава знање корисника.

Згодно би било да твој квиз броји тачне одговоре и приказује кориснику колики број поена има.

Право је време да уведеш променљиву L10_PromBodovi.

[](https://petlja.org/biblioteka/r/lekcije/scratch-support/_images/L10_KreiranjePromBodovi.png)

Колика треба да буде вредност променљиве L10_PromBodovi на почетку програма (упиши број): L10_Z1 

ПРОВЕРИ

Јасно је да вредност променљиве L10_PromBodovi треба да се увећа за **1** када корисник тачно одговори на питање.

[](https://petlja.org/biblioteka/r/lekcije/scratch-support/_images/L10_BodoviPlusJedan.png)

**Квиз помоћу листи**

Креирање квиза коришћењем листи далеко је елегантније решење од претходног.

За почетак, треба да креираш две листе: L10_ListaPitanja и L10_ListaOdgovori. Познат ти је поступак додавања елемената у листу.

Приликом уноса, важно је да повежеш питања са одговорима, тј. да редни број елемента у листи L10_ListaPitanja одговара редном броју елемента у листи L10_ListaOdgovori. Погледај листе приказане на слици.

[](https://petlja.org/biblioteka/r/lekcije/scratch-support/_images/L10_DveListe.png)

Уочаваш да питање на позицији **1** листе L10_ListaPitanja има одговор на позицији **1** листе L10_ListaOdgovori. Овај принцип односи се и на остале елементе листи.

Оваква структура неопходна нам је због спаривања питања са тачним одговором и врло згодна за унапређивање динамике нашег квиза. Уместо да се квиз извршава сваки пут на исти начин, листе ће нам омогућити разноврсност!

Хајде да размишљамо о насумичном одабиру питања из листе L10_ListaPitanja.

Ако рачунару кажемо да насумично одабере питање, нећемо знати које питање бира, а самим тим нећемо знати који је тачан одговор на њега.

У овој ситуацији, згодно нам је да користимо обичну променљиву у којој ћемо чувати број насумично одабраног питања.

Q-9: Зашто је потребно да чувамо редни број (положај у листи) насумично одабраног питања?

Top of Form

A. Овај број нам је неопходан да бисмо могли да упоредимо одговор корисника са тачним одговором (елементом листе Одговори).  
B. Овај број нам је неопходан да бисмо знали да су постављена сва питања.  
ПРОВЕРИ

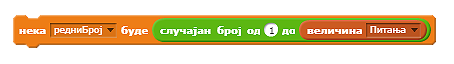
Bottom of Form

Дакле, креираћемо променљиву L10_PromRBrListe.

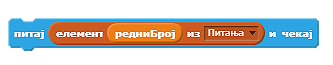
Наредни корак јесте обезбеђивање да променљива L10_PromRBrListe узме насумичну вредност од 1 до 3. Наравно, број 3 узимамо зато што је дужина наших листи 3 (оне садрже по три елемента).

Да ли је овакво размишљање заиста оправдано? Шта ако будемо желели да додамо елементе у листе?

Далеко елегантније решење је да наша променљива L10_PromRBrListe узме насумичну вредност од 1 до **величине листе** (броја елемената у листи).

[](https://petlja.org/biblioteka/r/lekcije/scratch-support/_images/L10_DoDuzineListe.png)

Читаву ову заврзламу радили смо да би стигли до могућности да кориснику поставимо питање следећег облика:

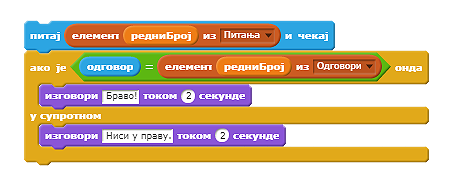
[](https://petlja.org/biblioteka/r/lekcije/scratch-support/_images/L10_NasumicnoPitanje.png)

Наш програм поставља кориснику под редним бројем сачуваним у променљивој L10_PromRBrListe и чека на одговор корисника.

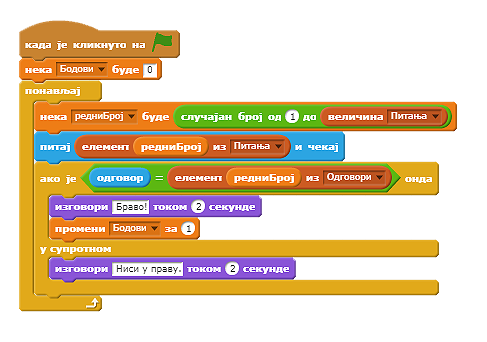
Следи поређење корисниковог одговора са тачним одговором.

[_images/L10_PoredjenjeOdgovora.png](https://petlja.org/biblioteka/r/lekcije/scratch-support/_images/L10_PoredjenjeOdgovora.png)

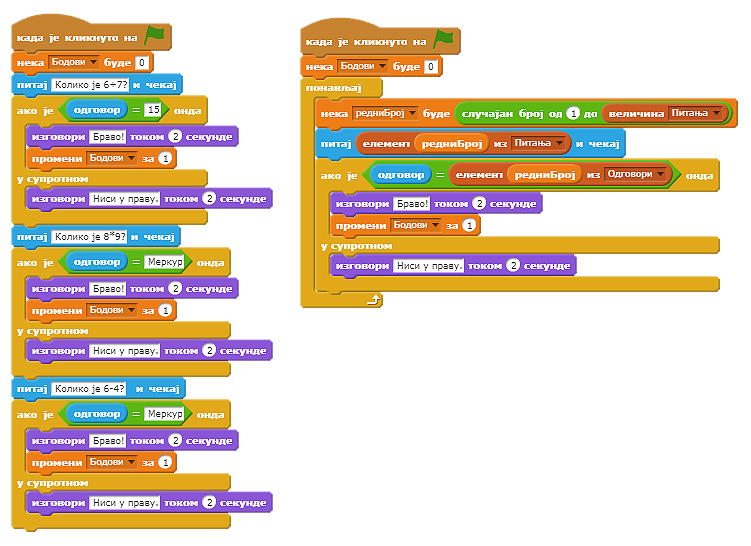
Ако све ово ставимо у контекст, скрипта у којој постављамо питање кориснику, упоређујемо његов одговор са тачним и извршавамо гранање изгледа као на слици.

[](https://petlja.org/biblioteka/r/lekcije/scratch-support/_images/L10_PitanjeOdgovor.png)

Остало је да бројимо бодове и понављамо насумични избор питања из листе.

[](https://petlja.org/biblioteka/r/lekcije/scratch-support/_images/L10_KvizListe.png)

Скрипте квиза програмираног уз помоћ обичних променљивих и уз помоћ листи.

[](https://petlja.org/biblioteka/r/lekcije/scratch-support/_images/L10_2Kviza.png)