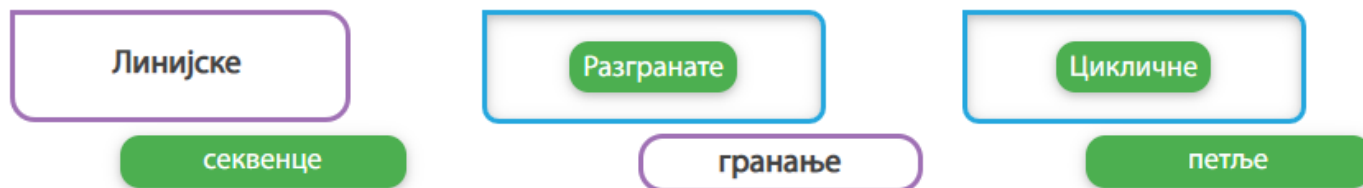


27. Час Информатика 6. разред _Гранање програма (разграната програмска структура)

Програмска структура представља редослед појединих наредби приликом извршавања програма. Редослед извршавања наредби у програмирању не мора увек бити исти. Некада је условљен претходним резултатима.



Прво погледај видео материјал са овог [линка](#):

Разграната програмска структура

Често се сусрећеш са потребом доношења одлуке о нечему, што ће бити услов даљег тока догађаја. На пример код питања: Да ли си урадио/урадила домаћи задатак? Услов код овог питања је да одговор буде тачан (енг. *true*) да би могао/могла да се играш.

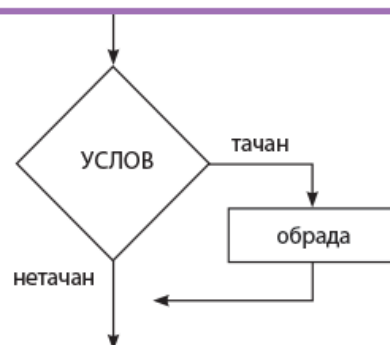
Кликни на поље да видиш пример, а затим проучи како се проверава тачност услова.

```
>>> uradio_sam_domaci = True
>>> if uradio_sam_domaci:
print("Možeš da se igraš")
```

Важно

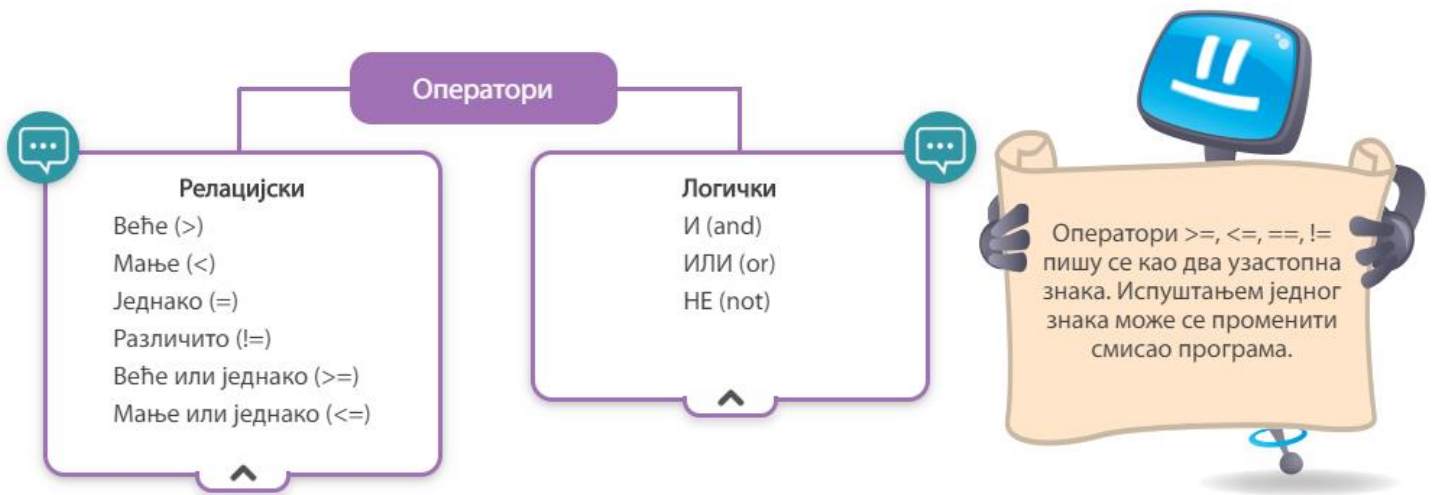
Након услова обавезно се мора навести двотачка (:). Наредбе које се условно извршавају морају се мало увући (можеш испред сваке наредбе откуцати три размака).

У програмирању услов је израз чија се истинитост проверава. Ако је услов тачан, извршиће се наведена наредба, а затим и остатак програма. Ако услов није тачан, одмах се прелази на остатак програма.



Оператори поређења

Услов се може испитивати уз помоћ оператора поређења, који могу бити релацијски и логички.



Логички оператори

Логички оператори омогућавају да се усмерава ток извршавања програма. Логички оператори су and (срп. и), or (срп. или) и not (срп. не).

И (AND)

Логички оператор **and** (срп. и) захтева да оба израза буду тачна да би цео сложени израз био тачан. На пример: $(5 < 7) \text{ and } (7 < 9)$ је тачно (True).

ИЛИ (OR)

Код логичког оператора **or** довољно је да је само један од два релацијска операнта или израза тачан да би цео сложени израз био тачан. На пример: $(5 < 7) \text{ or } (7 > 9)$ је тачно (True) јер је задовољан услов да бар један израз, у овом случају $(5 < 7)$, буде тачан.

НЕ (NOT)

Логички оператор **not** мења стање истинитости у супротно. На пример $\text{not}(5 > 7)$ је тачно (True).

If-else naredba

У програмима поједине naredbe ће се извршити само ако је задовољен неки услов. Замисли ситуацију из свакодневног живота да би боље разумео/разумела naredбу if-else.

Naredбом `if` омогућава се извршавање одређених услова у програмима написаним у Пајтону. If-else naredba значи да постоји још једно решење, и у зависности од испуњености услова омогућава да се изврши једна или друга naredba.

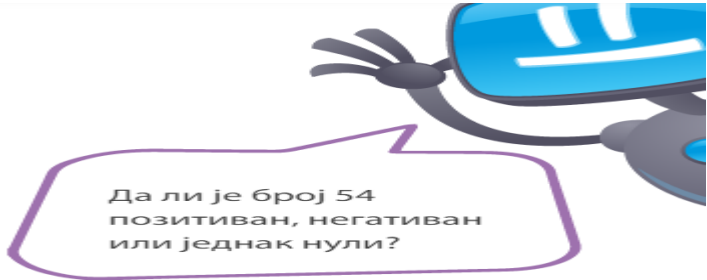
```
>>> if uslov:
naredba_1
...
naredba_m
>>> else:
naredba_1
...
naredba_n
```

На пример, ако је зелено светло на семафору, прећи ћеш улицу, ако није зелено светло, нећеш прећи улицу.



Конструкција `elif`

Понекад је потребно комбиновати више услова и потпуно их проверавати. То нам омогућује конструкција `elif`. На пример, желимо да испитамо да ли је неки број позитиван, нула или негативан.



Да ли је број 54 позитиван, негативан или једнак нули?

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

>>> broj = int(input("Unesi broj: "))
Unesi broj: 54
>>> if broj >0:
print("Broj je pozitivan")
>>> elif broj==0:
print("Broj je nula")
>>> else:
print("Broj je negativan")
Broj je pozitivan

Ln: 11 Col: 4
```

Први услов: да ли је број већи од нуле. Ако јесте, исписаће се да је број позитиван.

Други услов: да ли је број једнак нули и ако јесте исписаће се да је број 0.

Последњи услов: да ли је број мањи од нуле и ако јесте исписаће се да је број негативан.

Илуструјмо сада примену гранања и поређења кроз неколико једноставних програма:

1. Млади програмери

Напишимо програм који корисницима млађим од 15 година шаље посебну похвалу јер су кренули да програмирају веома рано.

```
1 godine = int(input("Koliko imate godina: "))
2 if godine < 15:
3     print("Tako ste mladi, a već programirate! Svaka čast!")
4
```

ПОКРЕНИ ПРОГРАМ

Koliko imate godina:

Потврди

Откажи

Tako ste mladi, a već programirate! Svaka čast!

2. Тобоган

Напиши програм који на основу висине детета одређује да ли дете сме да иде на тобоган.

```
1 visina = int(input("Unesi svoju visinu: "))
2 if visina >= 140:
3     print("Možeš na tobogan")
4
```

ПОКРЕНИ ПРОГРАМ

Unesi svoju visinu:

Потврди

Откажи

Možeš na tobogan

Можемо мало и да модификујемо овај програм:

```
1 visina = int(input("Unesi svoju visinu: "))
2 if visina >= 140:
3     print("Možeš na tobogan")
4 else:
5     print("Još ne možeš na tobogan")
6     print("Moraš još malo da porasteš")
7
```

ПОКРЕНИ ПРОГРАМ

Unesi svoju visinu:

Потврди

Откажи

Još ne možeš na tobogan
Moraš još malo da porasteš

3. Парност броја

Напиши програм који испитује да ли је унети број паран или непаран.

Проверу да ли је један број дељив другим можемо извршити тако што израчунамо остатак при дељењу и упоредимо га са нулом. Подсетимо се, о целобројном дељењу је већ било речи у глави о дељењу и тада смо рекли да се остатак може израчунати коришћењем оператора %.

```
1 broj = int(input("Unesite broj za proveru parnosti - neparnosti"))
2 if broj % 2 == 0:
3     print("Broj je paran")
4 else:
5     print("Broj je neparan")
6
```

ПОКРЕНИ ПРОГРАМ

Unesite broj za proveru parnosti - neparnosti

Потврди

Откажи

Broj je paran

ПОКРЕНИ ПРОГРАМ

Unesite broj za proveru parnosti - neparnosti

Потврди

Откажи

Broj je neparan

Наком ових претходних програма спремни смо сада за још мало теорије:

Ред речи у речнику

Ниске се могу поредити и по абецедном или азбучном реду, како су ваша имена сложена у школском дневнику или како су речи сложене у речнику. Такво поређење се назива *лексикографско*. За то се користе исти оператори који се користе и за бројеве. На пример, важи да је "Ana" < "Biljana" јер је слово **A** испред слова **B** у абецеди. Слично, важи да је "Ana" > "Aleksandar", јер су им прва слова једнака, па се онда прелази на гледање другог слова, а на том месту се десило да је слово **n** у абецеди иза слова **l**. На крају, важи и да је "Ana" < "Anastasija", јер када се не наиђе на различито слово, тада се краћа реч ставља пре друге. Напоменимо да овакво поређење савршено ради за слова енглеског алфабета, али да су за слова наше латинице или ћирилице понекад потребна нека додатна подешавања. Обрати пажњу и на то да се приликом поређења прави разлика између великих и малих слова (сва велика слова иду пре свих малих слова).