

# Interpret jazyka Logo

## Shrnutí

Cílem je naimplementovat zjednodušený interpret jazyka Logo. Zdrojový soubor obsahuje příkazy pro kreslení obrázku. Výstupem je soubor `out.html` obsahující nakreslený svg obrázek nebo chybová hláška vypsána na std. výstup, pokud došlo k chybě.

## Příkazy

Každý příkaz je na samostatném řádku. K odsazení příkazů jsou použity mezery nebo tabulátory. Pokud je ve zdrojovém `.logo` souboru řádek, který neodpovídá syntaxi dané níže (např. prázdný řádek nebo příkaz *repeat* bez koncové uzavírací závorky), považujte takový vstup za chybný a ohlašte chybu parsování.

`penup`

Zvedne kurzor, následná volání *forward* kurzor pouze přesouvají (tj. nedochází ke kreslení na plátno).

`pendown`

Spustí kurzor, následné volání *forward* přesouvá kurzor a zároveň kreslí úsečku na plátno.

`right <int degrees>`

Natočí kurzor o úhel daný parametrem *degrees*. Otočení proběhne po směru hodinových ručiček. Hodnota může být i záporná, v tom případě odpovídá otočení proti směru hodinových ručiček.

`forward <int distance>`

Posune kurzor o vzdálenost danou celočíselným parametrem *distance*. Pokud je kurzor spuštěný, nakreslí úsečku, jinak jej pouze přesouvá. Kurzor kreslí (příp. se jen posouvá) ve směru aktuálního natočení kurzoru.

```
repeat <int repetitions> [  
  <command 1>  
  <command 2>  
  ...  
  <command n>  
]
```

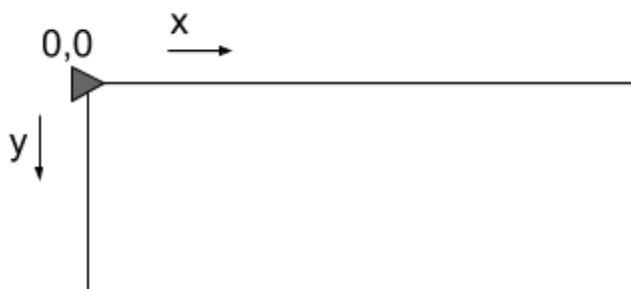
Vykoná *repetitions* opakování sekvence příkazů *command 1* až *command n*.

# Plátno, startovní pozice a hodnota kurzoru

Plátno je orientováno tak, že souřadnice  $[0,0]$  se nachází v jeho levém horním rohu. Osa x je vodorovná, osa y svislá. Hodnoty se na osách zvyšují zhora dolů a zleva doprava.

Kurzor je před prvním příkazem spuštěný *dolů*.

Plátno má vždy rozměry 500 x 500



## Výstup

Výstupní html soubor má následující strukturu:

### Hlavička

```
<!DOCTYPE html>
<html><body>
<svg width='500' height='500'>
<style type='text/css'>
<![CDATA[
line { stroke: rgb(0, 0, 0); stroke-width:2}
]]>
</style>
```

### Tělo

Pro každou úsečku z bodu  $[x_1, y_1]$  do bodu  $[x_2, y_2]$  obsahuje výstupní soubor řádku v následujícím formátu:

```
<line x1='..' y1='..' x2='..' y2='..' />
```

## Patička

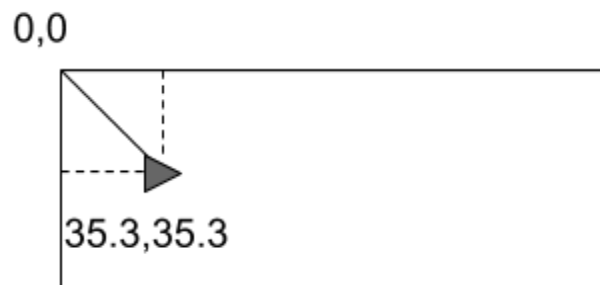
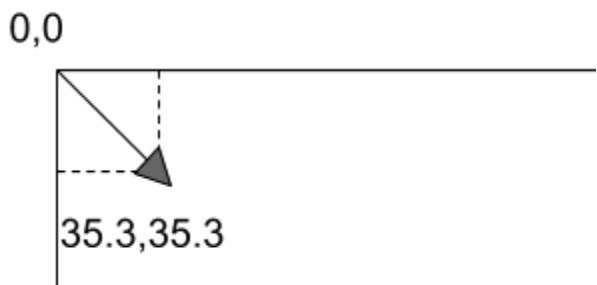
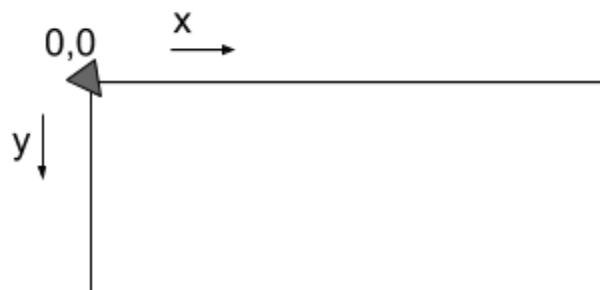
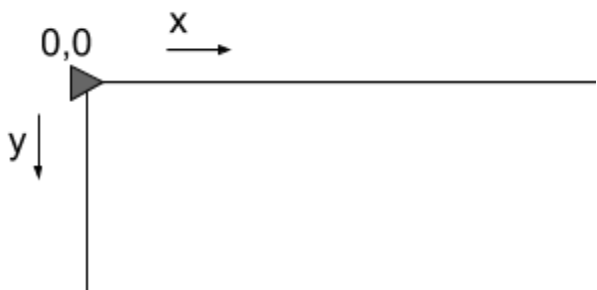
```
</svg></body></html>
```

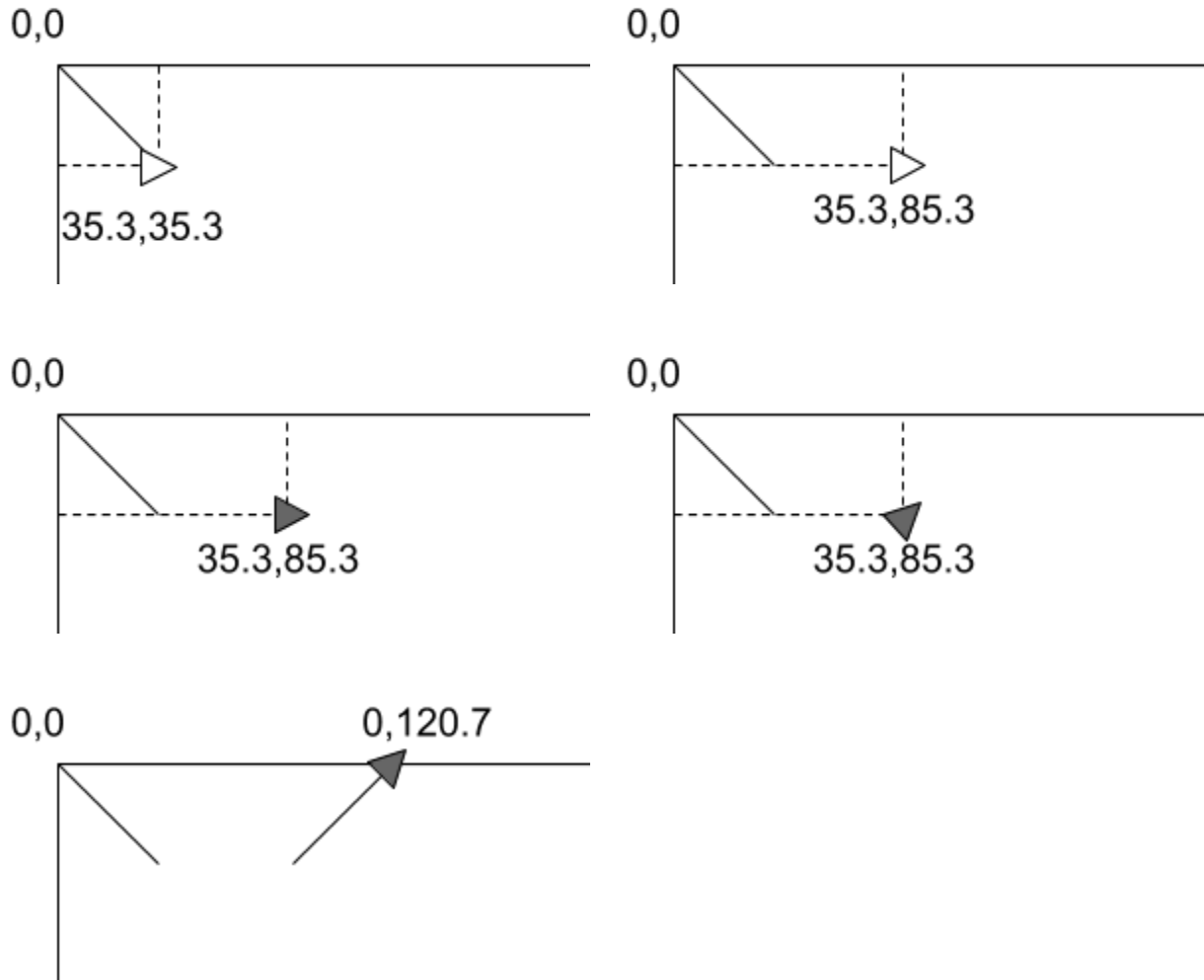
## Příklad

### Vstupní soubor

```
right 45  
forward 50  
right -45  
penup  
forward 50  
pendown  
right -45  
forward 50
```

### Postup výpočtu





## Výstupní soubor

```
<!DOCTYPE html>
<html><body>
<svg width='500' height='500'>
<style type='text/css'>
<![CDATA[
line { stroke: rgb(0, 0, 0); stroke-width:2}
]]>
</style>
<line x1='0' y1='0' x2='35.3553390593274' y2='35.3553390593274' />
<line x1='85.3553390593274' y1='35.3553390593274'
x2='120.710678118655' y2='0' />
</svg></body></html>
```

## Chybové stavy, okrajové podmínky

Program při správném spuštění dostane právě jeden argument se jménem vstupního souboru.

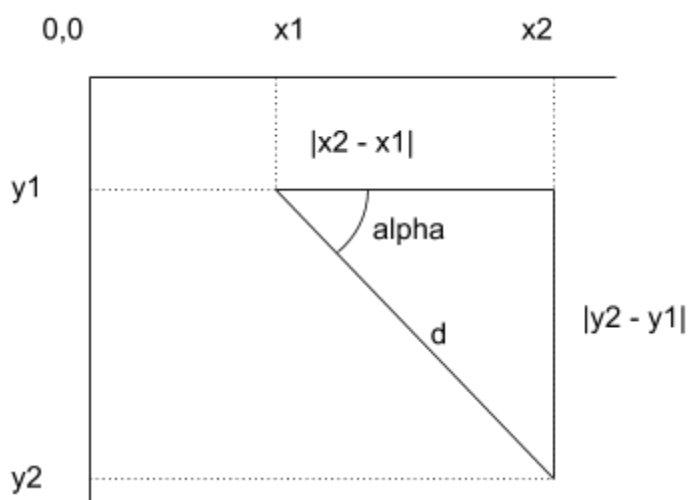
Při chybě program vypíše:

- Špatný počet argumentů - `Argument error`
- Chyba při čtení ze vstup. souboru či zápisu do výstup. souboru - `File error`
- Chyba při parsování vstup. souboru - `Parse error`

Pohyb kurzoru po plátně je omezen rozměry plátna. Poloha kurzoru tedy nemůže klesnout pod hodnotu 0 a nemůže překročit hodnotu 500 pro obě osy.

## Goniometrické funkce

- `Math.Cos`, `Math.Sin` - pozor, funkce počítají v radiánech, pro přepočítání z úhlů na radiány se může hodit konstanta `Math.PI`:



$$\sin(\alpha * \text{PI} / 180) = (y2 - y1) / d$$

$$\cos(\alpha * \text{PI} / 180) = (x2 - x1) / d$$

## Upozornění

- Parametry u všech příkazů jazyka Logo jsou celočíselné, ale výsledné souřadnice úseček na plátně jsou desetinná čísla.
- Rozsah otočení není omezen, tj. otočení o více než 360 stupňů je v korektní. Stejně tak je korektní otočení o záporný počet stupňů.
- Podobně posunutí je možné provést o lib. hodnotu, tj. i zápornou.
- Vícenásobné zvednutí kurzoru po sobě (či jeho spuštění) je korektní, kurzor se v takovém případě chová stejně, jako kdyby se zvednutí (spuštění) zavolalo jen jednou.
- Aby se vypisovala desetinná čísla s tečkou místo čárky, nastavte `CultureInfo` pomocí `Thread.CurrentThread.CurrentCulture = new CultureInfo("en-US");`

- Pokud při vykreslování narazíte na okraj, tak se vykreslování může buď zarazit nebo může pokračovat jen v jedné ose (vyberte si, která varianta je pro vás jednodušší).