

Matematické funkce

Numerické funkce

Do této skupiny lze zařadit funkce pro zaokrouhlování, hledání zbytku po dělení, hledání nejvyššího společného dělitele a funkce pro převod čísel mezi různými formáty

■ **Floor[x]** ... zaokrouhlení čísla x na nejbližší nižší celé číslo

```
In[1]:= Floor[2.6]
```

```
Out[1]= 2
```

```
In[2]:= Floor[2.2]
```

```
Out[2]= 2
```

■ **Ceiling[x]** ... zaokrouhlení čísla x na nejbližší vyšší celé číslo

```
In[3]:= Ceiling[2.6]
```

```
Out[3]= 3
```

```
In[4]:= Ceiling[2.2]
```

```
Out[4]= 3
```

■ **Round[x]** ... zaokrouhlení čísla x dle matematických pravidel

```
In[5]:= Round[2.6]
```

```
Out[5]= 3
```

```
In[6]:= Round[2.2]
```

```
Out[6]= 2
```

■ **Mod[m, n]** ... zbytek po dělení m/n

```
In[7]:= Mod[5, 2]
```

```
Out[7]= 1
```

■ **Quotient[m, n]** ... celočíselné dělení m/n

```
In[8]:= Quotient[5, 2]
```

```
Out[8]= 2
```

- **GCD[x1, x2, ..., xn] ... největší společný dělitel čísel x1, x2, ..., xn**

In[19]:= **GCD**[150, 25, 100]

Out[19]= 25

- **LCM[x1, x2, ..., xn] ... nejmenší společný násobek čísel x1, x2, ..., xn**

In[20]:= **LCM**[6, 4, 8]

Out[20]= 24

- **N[x, n] ... převod čísla x na n platných míst**

In[12]:= **N**[π , 4]

Out[12]= 3.142

- **Rationalize[x] ... převod desetinného čísla x na zlomek**

In[13]:= **Rationalize**[1.8654]

Out[13]= $\frac{9327}{5000}$

- **Rationalize[x, dx] ... převod desetinného čísla x na zlomek s přesností dx**

In[14]:= **Rationalize**[1.8654, 0.01]

Out[14]= $\frac{13}{7}$

In[16]:= **Rationalize**[π , 0.01]

Out[16]= $\frac{22}{7}$

Funkce pro práci s prvočíslly

- **FactorInteger[x] ... rozklad čísla x na součin prvočísel**

In[21]:= **FactorInteger**[52]

Out[21]= {{2, 2}, {13, 1}}

Pozn. číslo 52 rozloženo na $2^2 + 13^1$

- **PrimeQ[x] ... zjistí, zda x je prvočíslo**

In[24]:= **PrimeQ**[1253]

Out[24]= False

PrimePi[x] ... zjistí počet prvo čísel v intervalu (1, x)

In[25]:= PrimePi[10]

Out[25]= 4

Prime[x] ... vypíše x - té prvo číslo

In[26]:= Prime[4]

Out[26]= 7

Úkol : Vypíšte p ehled prvních 100 prvo čísel

Teorie čísel a číselné soustavy**BaseForm[x, b] ... převede číslo x do číselné soustavy o základu b**

In[27]:= BaseForm[29, 2]

Out[27]//BaseForm=
11101₂

In[28]:= BaseForm[29, 16]

Out[28]//BaseForm=
1d₁₆**b^x ... převede číslo x ze soustavy o základu b do desítkové soustavy**In[29]:= 16^{1d}

Out[29]= 29

In[30]:= 2¹¹¹⁰¹

Out[30]= 29

IntegerDigits[x, b] ... zobrazí jednotlivé číslice celého čísla x v číselné soustavě o základu b. Výsledek bude zobrazen ve formátu vektoru

In[31]:= IntegerDigits[29, 2]

Out[31]= {1, 1, 1, 0, 1}

RealDigits[x, b] ... zobrazí jednotlivé číslice reálného čísla x v číselné soustavě o základu b. Výsledek bude zobrazen ve formátu vektoru. Navíc zobrazí i počet číslic před desetinnou tečkou

RealDigits[16.36, 10]

Out[34]= {{1, 6, 3, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, 2}

- **RealDigits[x, b, d]** ... zobrazí jednotlivé číslice reálného čísla x v číselné soustavě o základu b na d desetinných míst. Výsledek bude zobrazen ve formátu vektoru. Navíc zobrazí i počet číslic před desetinnou tečkou

```
In[35]:= RealDigits[ $\pi$ , 10, 3]
```

```
Out[35]= {{3, 1, 4}, 1}
```

- **FromDigits[list, b]** ... sestaví číslo v číselné soustavě o základu b z cifer uvedených v seznamu list

```
In[36]:= FromDigits[{1, 1, 1, 0, 1}, 2]
```

```
Out[36]= 29
```

- **DigitCount[n, b, d]** ... určí počet číslic d v zápisu čísla n v číselné soustavě b

```
In[37]:= DigitCount[29, 2, 1]
```

```
Out[37]= 4
```

```
In[38]:= DigitCount[29, 2, 0]
```

```
Out[38]= 1
```

Logické operace s čísly

- **BitAnd[n1, n2, ...]** ... logický součin jednotlivých cifer čísel $n1, n2, \dots$ ve dvojkové soustavě. Čísla jsou ve funkci zadávána v soustavě desítkové.

```
In[41]:= BitAnd[10, 12]
```

```
Out[41]= 8
```

```
In[44]:= BaseForm[10, 2]
```

```
Out[44]//BaseForm=
10102
```

```
In[45]:= BaseForm[12, 2]
```

```
Out[45]//BaseForm=
11002
```

```
In[46]:= BaseForm[8, 2]
```

```
Out[46]//BaseForm=
10002
```

- **BitOr[n1, n2, ...]** ... logický součet jednotlivých cifer čísel $n1, n2, \dots$ ve dvojkové soustavě. Čísla jsou ve funkci zadávána v soustavě desítkové.

```
In[42]:= BitOr[10, 12]
```

```
Out[42]= 14
```

- **BitXor[n1, n2, ...]** ... logický exkluzivní součet jednotlivých cifer čísel n1, n2, ... ve dvojkové soustavě. Čísla jsou ve funkci zadávána v soustavě desítkové.

In[43]:= `BitXor[10, 12]`

Out[43]= 6

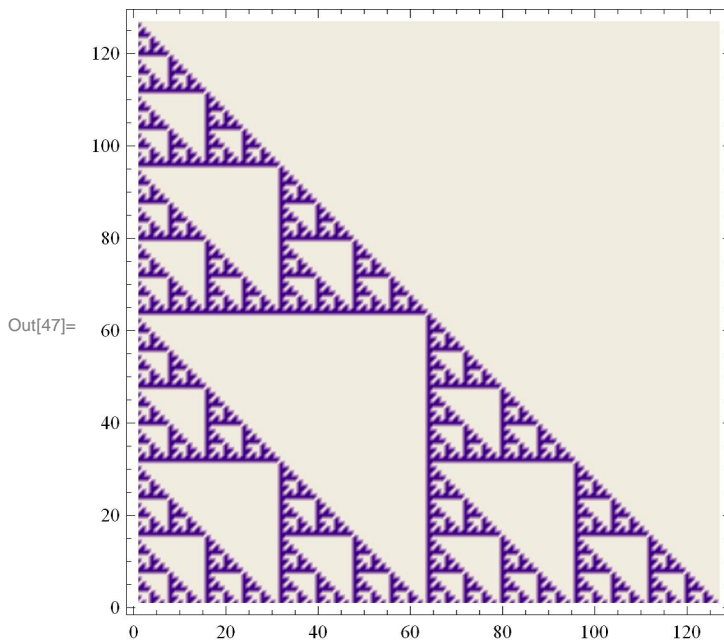
- **BitNot[n]** ... negace čísla n

In[39]:= `BitNot[12]`

Out[39]= -13

- **Využití logických operací pro konstrukci fraktálů**

In[47]:= `ListDensityPlot[Table[Sign[BitAnd[x, y]], {x, 127}, {y, 127}]]`



Generování náhodných čísel

- **Funkce Random[]** ... vygenerování náhodného čísla

- **Random[]** ... Základní volání bez parametrů - náhodné číslo 0 - 1

In[48]:= `Random[]`

Out[48]= 0.147288

- **Random[DatovýTyp, {od, do}]** ... generování náhodného čísla v zadaném intervalu. Datový typ : Integer, Real

In[49]:= `Random[Integer, {20, 30}]`

Out[49]= 27

```
In[50]:= Random[Real, {1, 5}]
```

```
Out[50]= 2.86172
```

- **Random[DatovýTyp, {od, do}, Po etMíst] ... generování náhodného ísla v zadaném intervalu. Datový typ : Real**

```
In[51]:= Random[Real, {1, 5}, 2]
```

```
Out[51]= 2.4
```

- **Precision[íslo] ... ov ení p esnosti zadaného ísla**

```
In[52]:= Precision[%]
```

```
Out[52]= 2.
```

- **Funkce SeedRandom[n] ... nastavení po átku generátoru náhodných ísel**

```
In[55]:= SeedRandom[5]
```

```
In[56]:= Table[Random[], {10}]
```

```
Out[56]= {0.786599, 0.848339, 0.612827, 0.398038,  
0.238012, 0.517232, 0.238887, 0.194399, 0.893263, 0.981672}
```

```
In[57]:= Table[Random[], {10}]
```

```
Out[57]= {0.674856, 0.354845, 0.953934, 0.846544,  
0.931578, 0.0765291, 0.140308, 0.488178, 0.343053, 0.682399}
```

```
In[58]:= SeedRandom[5]
```

```
In[59]:= Table[Random[], {10}]
```

```
Out[59]= {0.786599, 0.848339, 0.612827, 0.398038,  
0.238012, 0.517232, 0.238887, 0.194399, 0.893263, 0.981672}
```