

Matematika

Funkcia sčítania

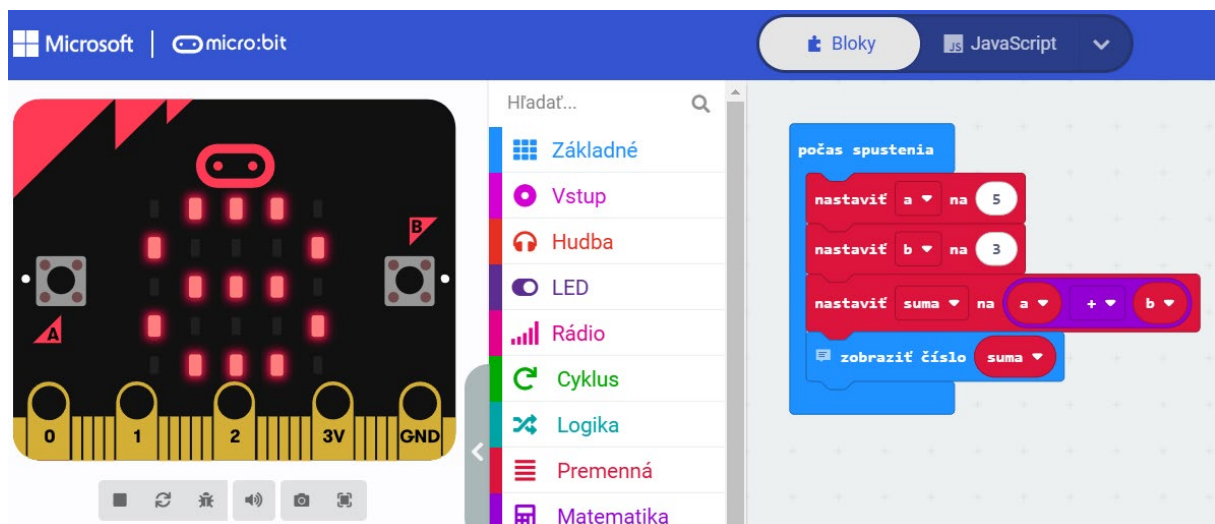
Funkcia sčítania v MakeCode slúži na vykonanie aritmetickej operácie sčítania medzi dvoma číslami. Táto operácia je veľmi užitočná na rôzne výpočty a je často využívaná v rôznych projektoch s BBC micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme sčítať.
2. Vykonanie sčítania: Potom použijeme príkaz na sčítanie týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť sčítanie v MakeCode:



Obrázok 1 Funkcia sčítania

V tomto príklade:

1. Definujeme premennú a s hodnotou 5.
2. Definujeme premennú b s hodnotou 3.
3. Sčítame tieto dve premenné a výsledok uložíme do premennej suma.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej suma, čo je 8.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí výsledok sčítania.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia odčítania

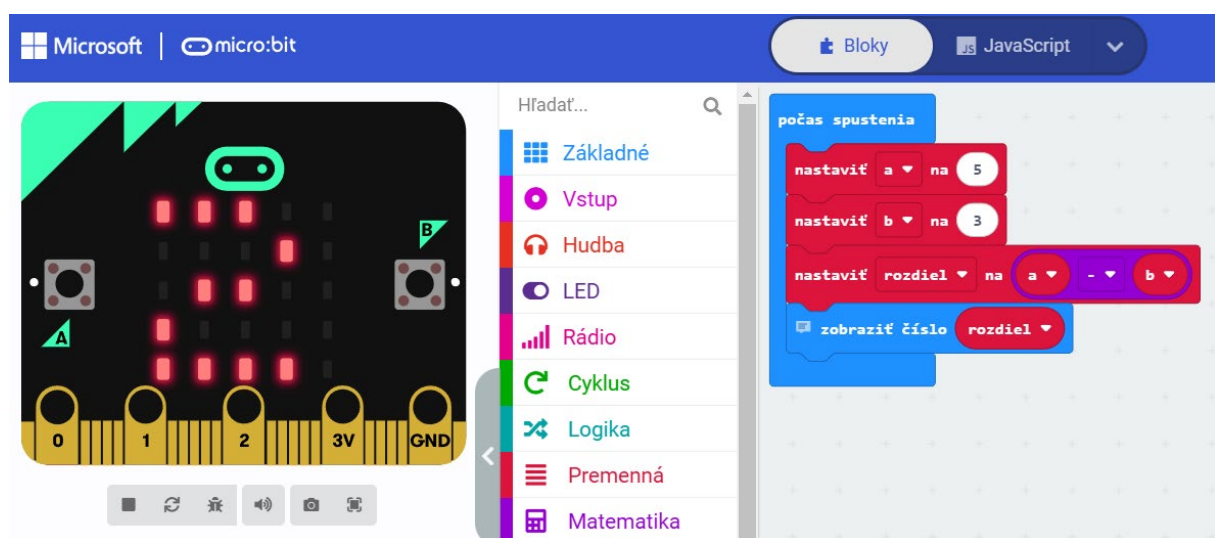
Funkcia odčítania v MakeCode slúži na vykonanie aritmetickej operácie odčítania medzi dvoma číslami. Táto operácia je veľmi užitočná na rôzne výpočty a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme odčítať.
2. Vykonanie odčítania: Potom použijeme príkaz na odčítanie týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť odčítanie v MakeCode:



Obrázok 2 Funkcia odčítania

V tomto príklade:

1. Definujeme premennú a s hodnotou 5.
2. Definujeme premennú b s hodnotou 3.
3. Odčítame premennú b od premennej a a výsledok uložíme do premennej rozdiel.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej rozdiel, čo je 2.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí výsledok odčítania.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia násobenia

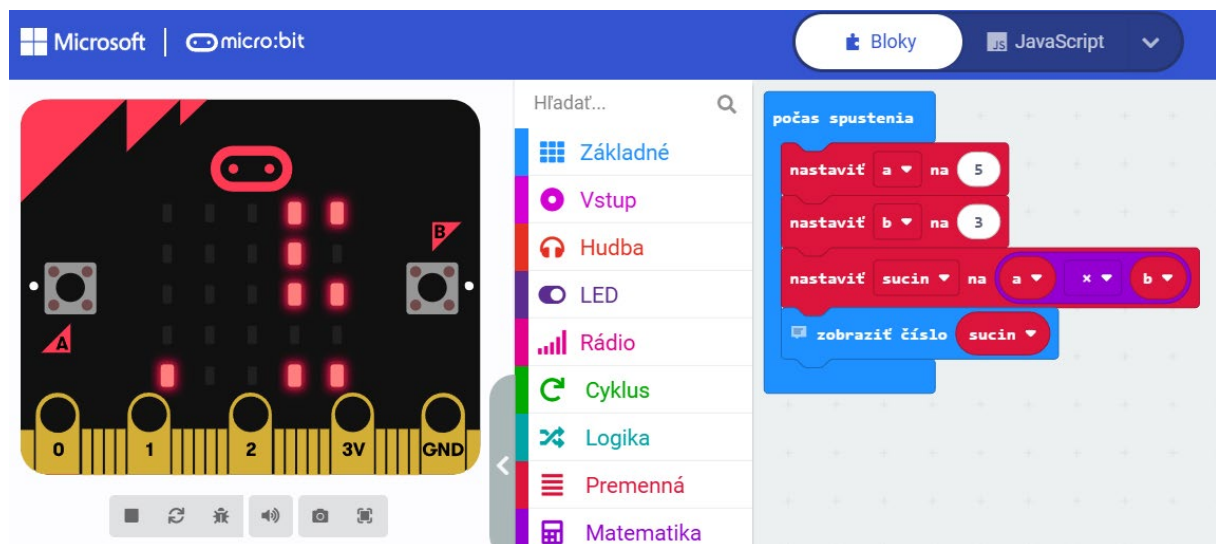
Funkcia násobenia v MakeCode slúži na vykonanie aritmetickej operácie násobenia medzi dvoma číslami. Táto operácia je veľmi užitočná na rôzne výpočty a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme násobiť.
2. Vykonanie násobenia: Potom použijeme príkaz na násobenie týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť násobenie v MakeCode:



Obrázok 3 Funkcia násobenia

V tomto príklade:

1. Definujeme premennú a s hodnotou 5.
2. Definujeme premennú b s hodnotou 3.
3. Násobíme tieto dve premenné a výsledok uložíme do premennej sucin.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej sucin, čo je 15.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí výsledok násobenia.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia delenia

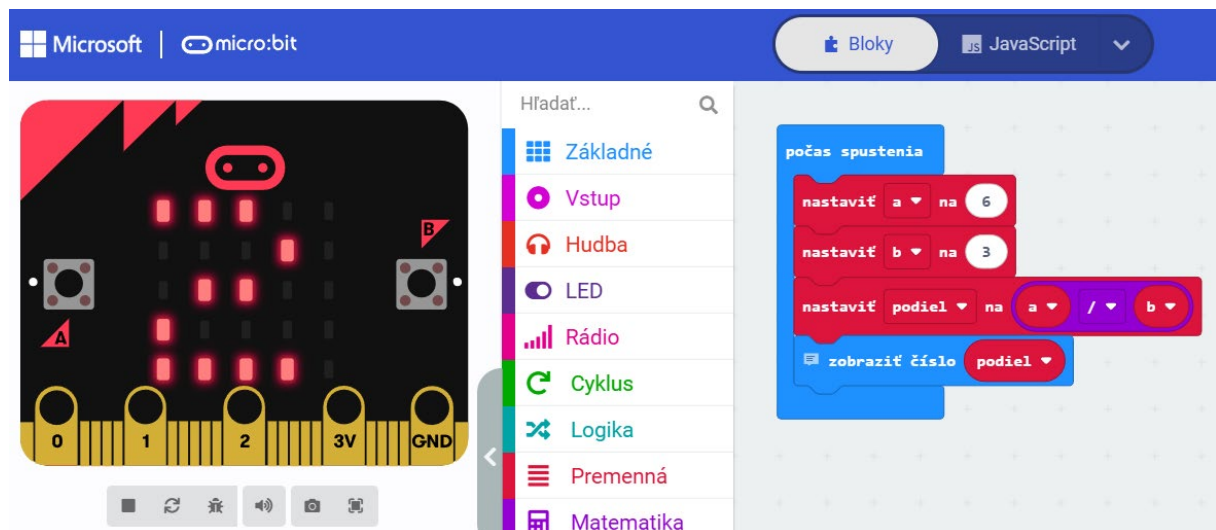
Funkcia delenia v MakeCode slúži na vykonanie aritmetickej operácie delenia medzi dvoma číslami. Táto operácia je veľmi užitočná na rôzne výpočty a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme deliť.
2. Vykonanie delenia: Potom použijeme príkaz na delenie týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť delenie v MakeCode:



Obrázok 4 Funkcia delenia

V tomto príklade:

1. Definujeme premennú a s hodnotou 6.
2. Definujeme premennú b s hodnotou 3.
3. Delíme premennú a premennou b a výsledok uložíme do premennej podiel.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej podiel, čo je 2.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí výsledok delenia.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia zvyšku po delení (remainder of)

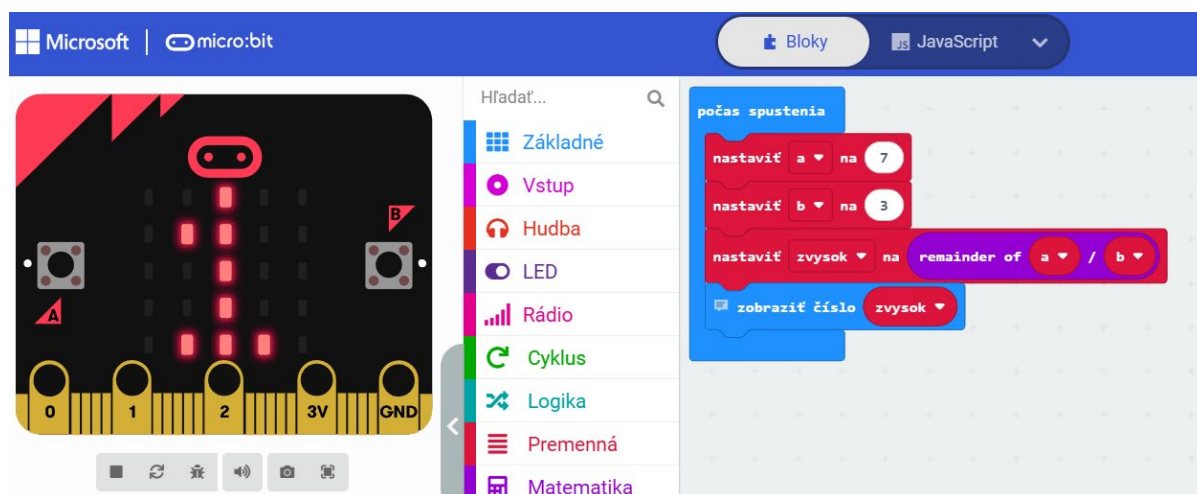
Funkcia zvyšku po delení v MakeCode slúži na vykonanie aritmetickej operácie zistenia zvyšku po delení medzi dvoma číslami. Táto operácia je veľmi užitočná na rôzne výpočty a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme deliť a zistiť zvyšok.
2. Vykonanie operácie: Potom použijeme príkaz na zistenie zvyšku po delení týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu zvyšku po delení v MakeCode:



Obrázok 5 Funkcia zvyšku po delení (remainder of)

V tomto príklade:

1. Definujeme premennú a s hodnotou 7.
2. Definujeme premennú b s hodnotou 3.
3. Zistíme zvyšok po delení premennej a premennou b a výsledok uložíme do premennej zvysok.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej zvysok, čo je 1.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí výsledok zvyšku po delení.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia minimálnej hodnoty (min)

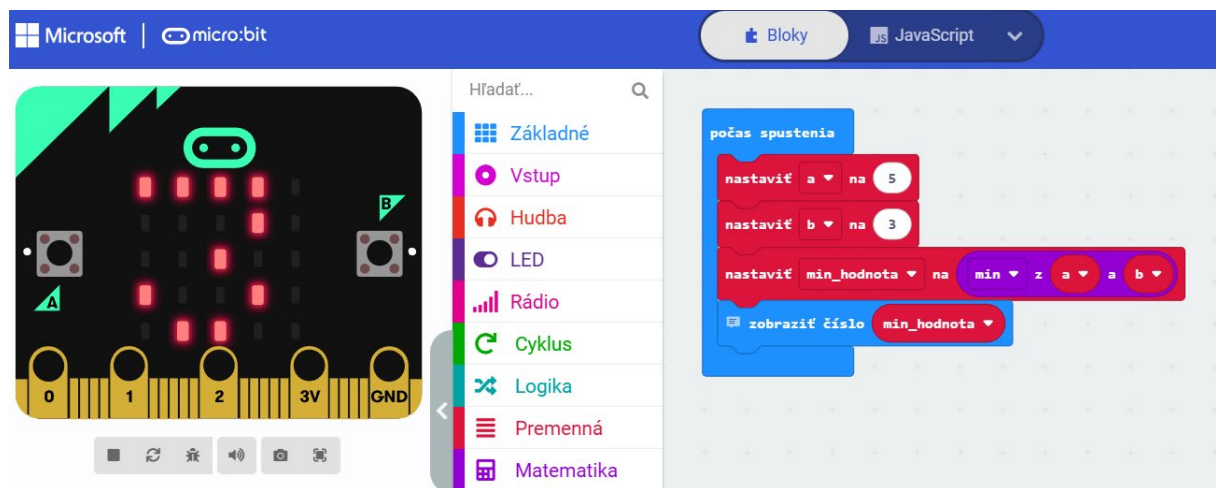
Funkcia minimálnej hodnoty (min) v MakeCode slúži na zistenie menšej z dvoch hodnôt. Táto operácia je veľmi užitočná pri rôznych výpočtoch a porovnávaniach a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme porovnať.
2. Zistenie minimálnej hodnoty: Potom použijeme príkaz na zistenie menšej z týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu minimálnej hodnoty v MakeCode:



Obrázok 6 Funkcia minimálnej hodnoty (min)

V tomto príklade:

1. Definujeme premennú a s hodnotou 5.
2. Definujeme premennú b s hodnotou 3.
3. Pomocou funkcie Math.min zistíme menšiu hodnotu medzi premennými a a b a výsledok uložíme do premennej min_hodnota.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej min_hodnota, čo je 3.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí minimálna hodnota z dvoch zadáných hodnôt.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a porovnávaniach a môže byť súčasťou komplexnejších programov.

Funkcia maximálnej hodnoty (max)

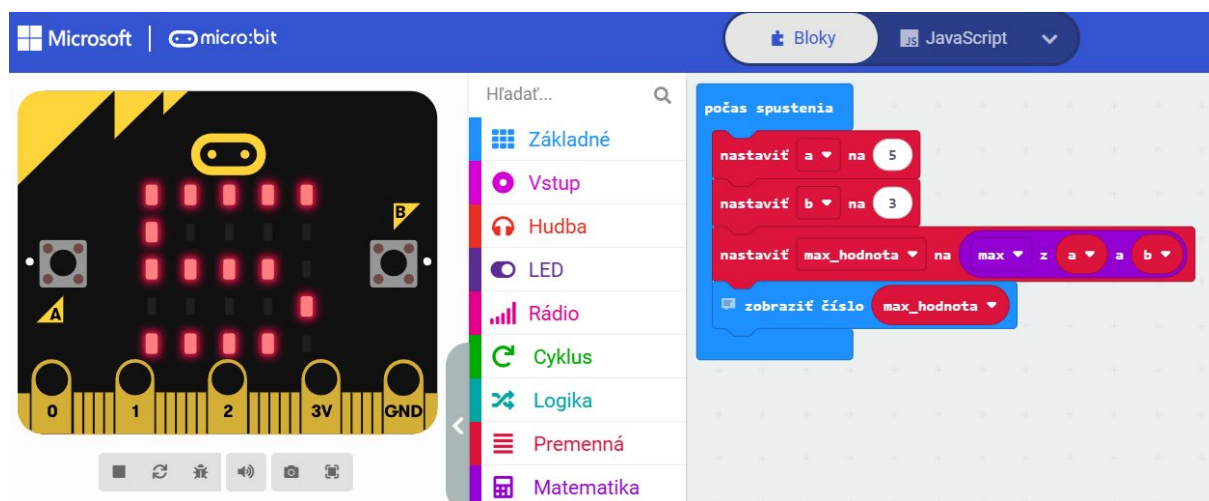
Funkcia maximálnej hodnoty (max) v MakeCode slúži na zistenie väčšej z dvoch hodnôt. Táto operácia je veľmi užitočná pri rôznych výpočtoch a porovnávaniach a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme porovnať.
2. Zistenie maximálnej hodnoty: Potom použijeme príkaz na zistenie väčšej z týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu maximálnej hodnoty v MakeCode:



Obrázok 7 Funkcia maximálnej hodnoty (max)

V tomto príklade:

1. Definujeme premennú a s hodnotou 5.
2. Definujeme premennú b s hodnotou 3.
3. Pomocou funkcie Math.max zistíme väčšiu hodnotu medzi premennými a a b a výsledok uložíme do premennej max_hodnota.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej max_hodnota, čo je 5.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí maximálna hodnota z dvoch zadáných hodnôt.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a porovnávaniach a môže byť súčasťou komplexnejších programov.

Funkcia absolútnej hodnoty

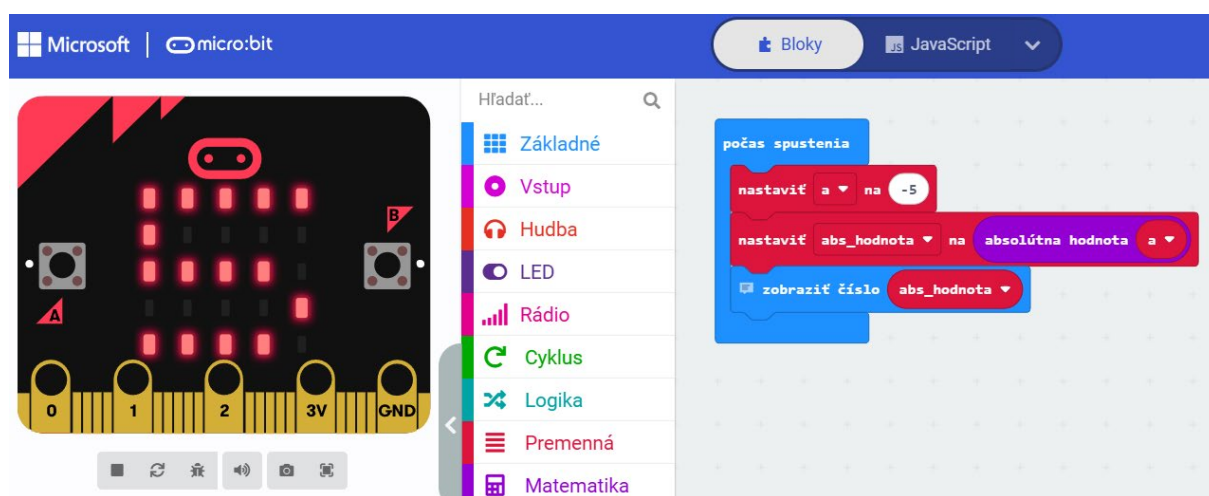
Funkcia absolútnej hodnoty v MakeCode slúži na zistenie absolútnej (kladnej) hodnoty zadaného čísla. Táto operácia je veľmi užitočná pri rôznych výpočtoch, kde je potrebné získať nezápornú hodnotu čísla, a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú, ktorej absolútnu hodnotu chceme zistiť.
2. Vykonanie operácie: Potom použijeme príkaz na zistenie absolútnej hodnoty tejto premennej a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu absolútnej hodnoty v MakeCode:



Obrázok 8 Funkcia absolútnej hodnoty

V tomto príklade:

1. Definujeme premennú a s hodnotou -5.
2. Pomocou funkcie Math.abs zistíme absolútnu hodnotu premennej a a výsledok uložíme do premennej abs_hodnota.
3. Na LED matici micro:bitu zobrazíme hodnotu premennej abs_hodnota, čo je 5.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí absolútna hodnota zadaného čísla.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia druhej odmocniny

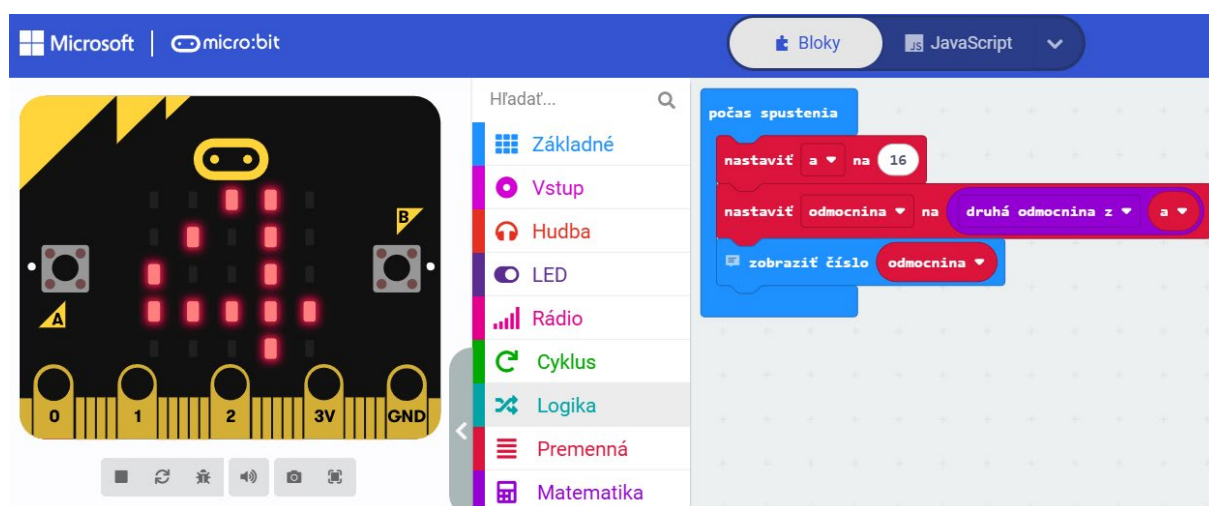
Funkcia druhej odmocniny v MakeCode slúži na zistenie druhej odmocniny zadaného čísla. Táto operácia je veľmi užitočná pri rôznych výpočtoch a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú, ktorej druhú odmocninu chceme zistiť.
2. Vykonanie operácie: Potom použijeme príkaz na zistenie druhej odmocniny tejto premennej a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu druhej odmocniny v MakeCode:



Obrázok 9 Funkcia druhej odmocniny

V tomto príklade:

1. Definujeme premennú `a` s hodnotou 16.
2. Pomocou funkcie `Math.sqrt` zistíme druhú odmocninu premennej `a` a výsledok uložíme do premennej `odmocnina`.
3. Na LED matici micro:bitu zobrazíme hodnotu premennej `odmocnina`, čo je 4.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí druhá odmocnina zadaného čísla.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia zaokrúhľovania

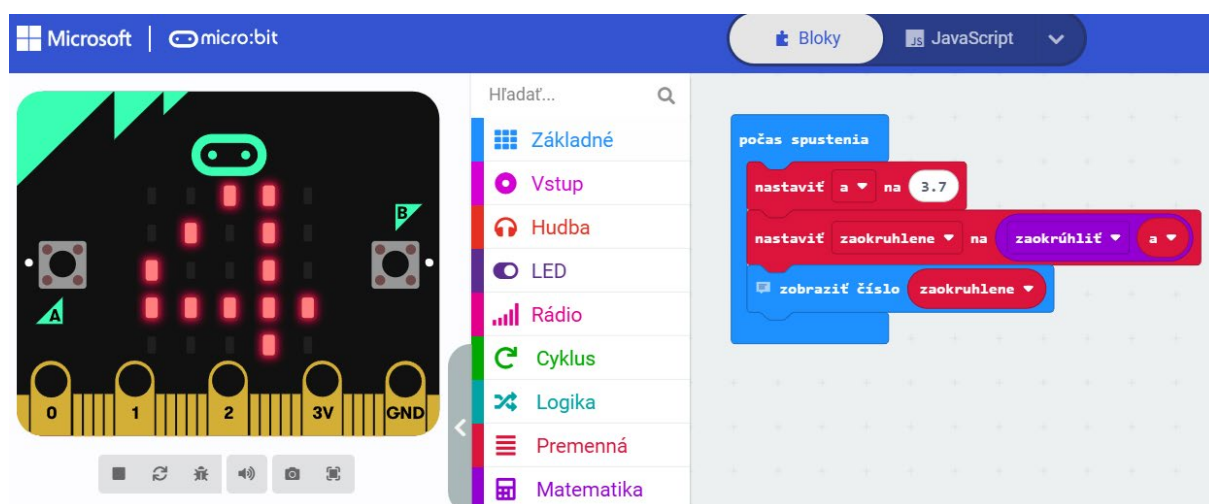
Funkcia zaokrúhľovania v MakeCode slúži na zaokrúhlenie zadaného čísla na najbližšie celé číslo. Táto operácia je veľmi užitočná pri rôznych výpočtoch a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú, ktorej hodnotu chceme zaokrúhliť.
2. Vykonanie operácie: Potom použijeme príkaz na zaokrúhlenie tejto premennej a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu zaokrúhľovania v MakeCode:



Obrázok 10 Funkcia zaokrúhľovania

V tomto príklade:

1. Definujeme premennú a s hodnotou 3.7.
2. Pomocou funkcie Math.round zaokrúhlime hodnotu premennej a a výsledok uložíme do premennej zaokruhlene.
3. Na LED matici micro:bitu zobrazíme hodnotu premennej zaokruhlene, čo je 4.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí zaokrúhlená hodnota zadaného čísla.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia výberu náhodného čísla

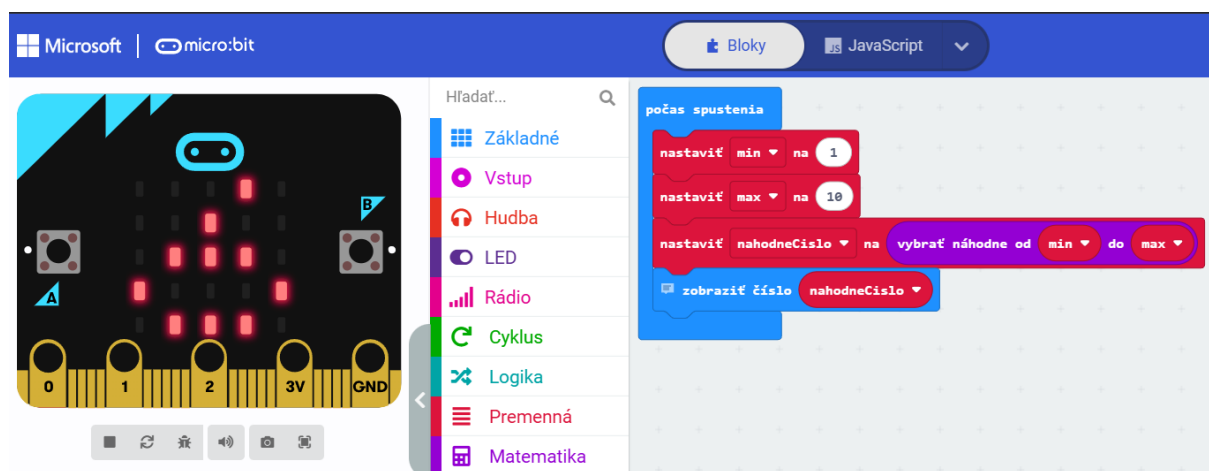
Funkcia výberu náhodného čísla v MakeCode slúži na generovanie náhodného čísla v zadanom rozsahu. Táto operácia je veľmi užitočná pri rôznych výpočtoch a hrách a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie rozsahu: Najprv definujeme rozsah, z ktorého chceme vybrať náhodné číslo.
2. Vykonanie operácie: Potom použijeme príkaz na výber náhodného čísla v tomto rozsahu a uloženie výsledku do premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu výberu náhodného čísla v MakeCode:



Obrázok 11 Funkcia výberu náhodného čísla

V tomto príklade:

1. Definujeme premennú min s hodnotou 1.
2. Definujeme premennú max s hodnotou 10.
3. Pomocou funkcie Math.randomRange generujeme náhodné číslo medzi hodnotami min a max a výsledok uložíme do premennej nahodneCislo.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej nahodneCislo, čo bude náhodné číslo v rozsahu od 1 do 10.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí náhodne vybrané číslo.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a hrách a môže byť súčasťou komplexnejších programov.

Funkcia obmedzenia hodnoty na interval

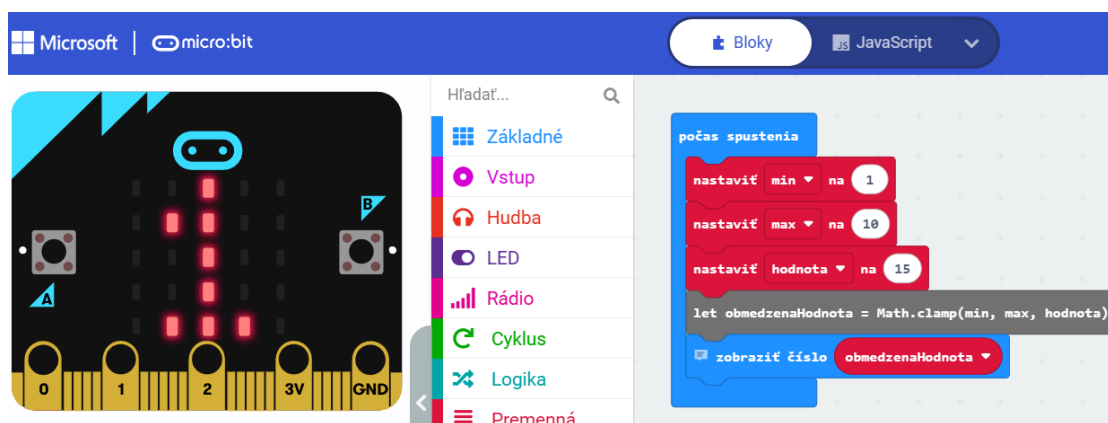
Funkcia obmedzenia hodnoty na interval v MakeCode slúži na obmedzenie zadaného čísla do zadaného rozsahu. Táto operácia je veľmi užitočná pri rôznych výpočtoch a kontrolách hodnôt a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie rozsahu: Najprv definujeme rozsah, v ktorom chceme obmedziť hodnotu.
2. Definovanie premennej: Definujeme premennú, ktorú chceme obmedziť do zadaného rozsahu.
3. Vykonanie operácie: Potom použijeme príkaz na obmedzenie tejto premennej do zadaného rozsahu a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu obmedzenia hodnoty na interval v MakeCode:



Obrázok 12 Funkcia obmedzenia hodnoty na interval

V tomto príklade:

1. Definujeme premennú min s hodnotou 1.
2. Definujeme premennú max s hodnotou 10.
3. Definujeme premennú hodnota s hodnotou 15.
4. Pomocou funkcie Math.clamp obmedzíme hodnotu premennej hodnota do rozsahu medzi min a max a výsledok uložíme do premennej obmedzenaHodnota.
5. Na LED matici micro:bitu zobrazíme hodnotu premennej obmedzenaHodnota, čo bude 10 (pretože 15 je mimo rozsahu 1 až 10 a je obmedzená na maximálnu hodnotu 10).

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí obmedzená hodnota zadaného čísla.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a kontrolách hodnôt a môže byť súčasťou komplexnejších programov.

Funkcia mapovania z intervalu od-do do iného intervalu

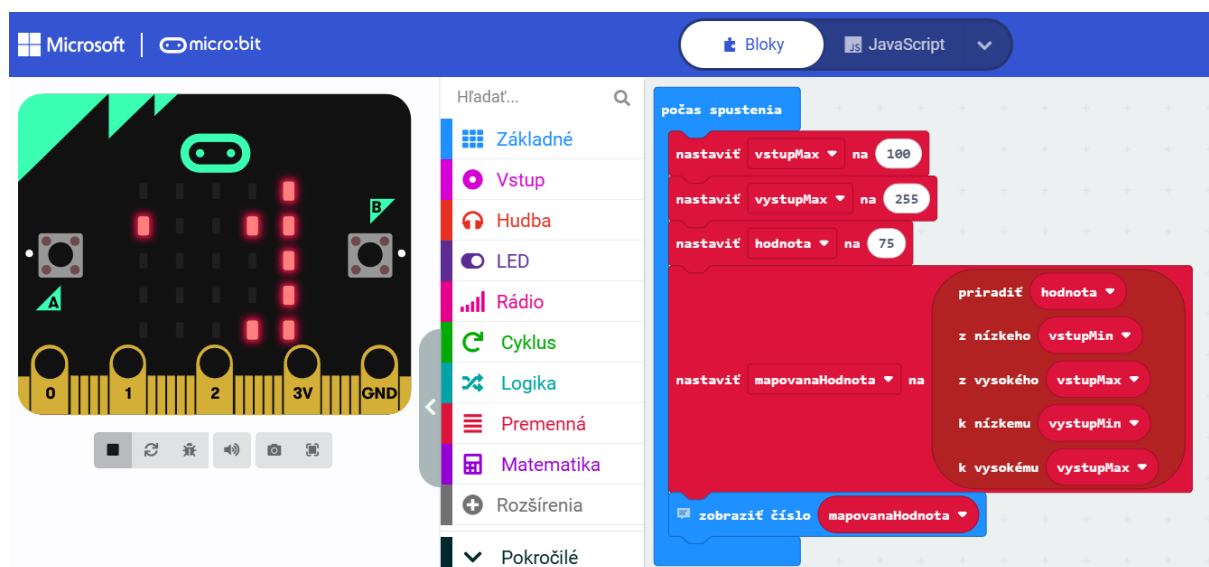
Funkcia mapovania v MakeCode slúži na prepočítanie hodnoty z jedného intervalu na zodpovedajúcu hodnotu v inom intervale. Táto operácia je veľmi užitočná pri rôznych výpočtoch a škálovaní hodnôt a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie vstupného a výstupného intervalu: Najprv definujeme vstupný interval, z ktorého chceme mapovať hodnotu, a výstupný interval, do ktorého chceme túto hodnotu mapovať.
2. Definovanie premennej: Definujeme premennú, ktorú chceme mapovať z jedného intervalu do druhého.
3. Vykonanie operácie: Potom použijeme príkaz na mapovanie tejto premennej a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu mapovania v MakeCode:



Obrázok 13 Funkcia mapovania z intervalu od-do do iného intervalu

V tomto príklade:

1. Definujeme premennú vstupMin s hodnotou 0.
2. Definujeme premennú vstupMax s hodnotou 100.
3. Definujeme premennú vystupMin s hodnotou 0.
4. Definujeme premennú vystupMax s hodnotou 255.
5. Definujeme premennú hodnota s hodnotou 75.
6. Pomocou funkcie pins.map mapujeme hodnotu premennej hodnota z intervalu 0-100 do intervalu 0-255 a výsledok uložíme do premennej mapovanaHodnota.
7. Na LED matici micro:bitu zobrazíme hodnotu premennej mapovanaHodnota, čo bude 191.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí mapovaná hodnota zadaného čísla.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a škálovaní hodnôt a môže byť súčasťou komplexnejších programov.

Funkcia náhodného výberu medzi pravdou a nepravdou

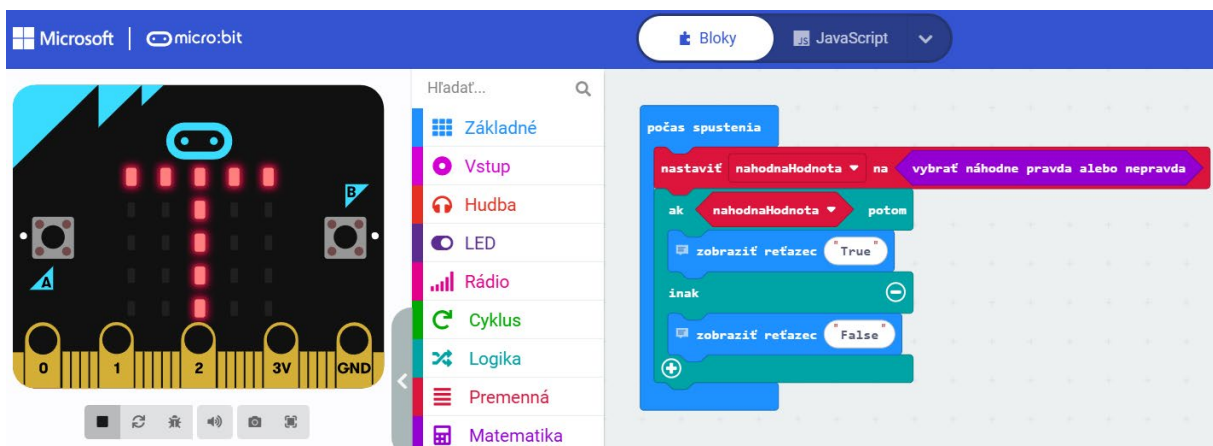
Funkcia náhodného výberu medzi pravdou a nepravdou v MakeCode slúži na generovanie náhodnej hodnoty true (pravda) alebo false (nepravda). Táto operácia je veľmi užitočná pri rôznych hrách, rozhodovaní a náhodných výberoch v projektoch s micro:bitom.

Ako to funguje?

- Vykonanie operácie: Použijeme príkaz na generovanie náhodnej hodnoty true alebo false a výsledok uložíme do premennej alebo priamo zobrazíme výsledok.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu náhodného výberu medzi pravdou a nepravdou v MakeCode:



Obrázok 14 Funkcia náhodného výberu medzi pravdou a nepravdou

V tomto príklade:

1. Pomocou funkcie `Math.randomBoolean` generujeme náhodnú hodnotu true alebo false a výsledok uložíme do premennej `nahodnaHodnota`.
2. Ak je náhodná hodnota true, na LED matici micro:bitu zobrazíme text "True".
3. Ak je náhodná hodnota false, na LED matici micro:bitu zobrazíme text "False".

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke [micro:bit](https://makecode.microbit.org).
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí náhodne vybraná hodnota pravda alebo nepravda.

Táto funkcia je veľmi užitočná pri rôznych hrách, rozhodovaní a náhodných výberoch a môže byť súčasťou komplexnejších programov.