

## 1. Zmienne

Zasady zmiennych:

- Nie zaczynamy od cyfr
- Nie używamy spacji tylko \_ lub kombinacja małych i dużych liter przykładowa\_zmienna, przykładowaZmienna
- Nie używamy znaków specjalnych i słów zakazanych
- Nie korzystamy z polskich ogonków
- Duża i mała litera ma znaczenie (Zmienna i zmienna)
- Self-descriptive – nazwa zmiennej powinna wskazywać co ona będzie robiła

TYP\_ZMIENNEJ NAZWA\_ZMIENNEJ (EW = WARTOŚĆ ZMIENNEJ)

sbyte	-128 do 127	8 bitów	1 bajt	liczby całkowite
short	-32.768 do 32.768	16 bitów	2 bajty	liczby całkowite
int	$-2^{31}$ do $2^{31} - 1$ (+/- 2 mld)	32 bity	4 bajty	liczby całkowite
long	$-2^{63}$ do $2^{63} - 1$	64 bity	8 bajtów	duże liczby całkowite
byte	0 do 255	8 bitów	1 bajt	liczby całkowite
ushort	0 do 65.535	16 bitów	2 bajty	liczby całkowite
uint	0 do $2^{32} - 1$	32 bity	4 bajty	liczby całkowite
ulong	0 do $2^{64} - 1$	64 bity	8 bajtów	duże liczby całkowite
float	$-1.5 \cdot 10^{(-45)}$ do $4.3 \cdot 10^{(-38)}$	32 bity	4 bajty	do 7 znaków po , F na końcu
double	$1 \cdot 10^{(-28)}$ do $7.9 \cdot 10^{28}$	64 bity	8 bajtów	do 28 znaków po ,
decimal	$10^{(-28)}$ do $7.9 \cdot 10^{28}$	128 bitów	16 bajtów	do wartości pieniężnych, 28-29 cyft, M na końcu
bool	prawda/fałsz	8 bitów	1 bajt	wartość logiczna
string	ciąg znaków	32 bity	4 bajty	tekst
char	pojedynczy znak	16 bitów	4 bajty	znak

```
double a = 2 / 3.0;  
Console.WriteLine(Math.Round(a, 2));
```

## 2. Operatory arytmetyczne

+ - \* / %

Inkrementacja

Post a++

Pre ++a

Dekrementacja

Post a--

Pre --a

a += 7 == a = a + 7

### 3. Operatory relacyjne

`==` true wtedy i tylko wtedy, gdy element są równe

`!=` true wtedy i tylko wtedy kiedy nie są równe

`>` większe niż

`<` mniejsze niż

`>=` większe lub równe

`<=` mniejsze lub równe

### 4. Operatory logiczne

Koniunkcja (i) `&&` - oba wyrażenia są prawdziwe

wyrażenie 1	wyrażenie 2	wynik
true	true	true
true	false	false
false	true	false
false	false	false

Alternatywa (lub) `||` - wystarczy, że jedno jest prawdziwe

wyrażenie 1	wyrażenie 2	wynik
true	true	true
true	false	true
false	true	true
false	false	false

Zaprzeczenie (negacja)

`!(true)` = false

`!(false)` = true